MASTER IN COMPUTER SCIENCE

The use of CCITT X.400 recommendations for EDI

Van Oost, Stanislas

Award date:
1991

Awarding institution:
University of Namur

Facultés Universitaires Notre-Dame de la Paix
Institut d'informatique

The use of CCITT X.400 recommendations for EDI

Thesis presented by Stanislas Van Oost in order to obtain the degree of Licencié et Maître en Informatique

Promoter - Professor Philippe van Bastelaer

Academic year 1990 - 1991

Rue Grandgagnage, 21, B - 5000 Namur (BELGIUM)

May we permitted to thank our promoter Professor van Bastelaer, for his good advice all along this work.

We would like also to thank Mister Ray Walker, Western European UN/EDIFACT Rapporteur and Vice-Chairman of the UN/EDIFACT Board, who had given to us the opportunity to do a probationary in SITPRO (UK).

Our wish is to thank also Mister John Berge and Mrs Man Sze Li who advised us during our stay in SITPRO between September 1990 and February 1991.

Finally, we thank each staff member of SITPRO and DG XXIII (CEC, Brussels) who helped us and each person who took part directly or indirectly in the achievement of this work.

Abstract :

This work explains how the CCITT X.400 recommendations message handling system can suit for Electronic Data Interchange (EDI). After an attempt of definition of EDI and a description of its technical needs, a large overview of the 1984 and 1988 X.400 recommendations is given. The EDI requirements for communication are matched with the capabilities of X.400 messaging handling system. Two possible solutions for the use of X.400 for EDI are explained. But these solutions which are based on the use of an interpersonal messaging system, do not suit perfectly for EDI. So a third solution describes a dedicated EDI message handling system based on X.400 recommendations. Finally, the possible use of a structured language, based on the notation ASN.1, for defining and coding EDI messages using X.400 is analysed.

Résumé :

Ce travail explique comment les recommandations X.400 du CCITT, pour un système de traitement de messages, peuvent convenir pour l'échange de données informatisées (EDI). Après une tentative de définition de l'EDI et une description de ses besoins techniques, un large aperçu des recommandations 1984 et 1988 de X.400 du CCITT est donné. Les exigences pour les moyens de communication pour l'EDI sont comparées aux capacités du système de traitement de messages X.400. Deux solutions possibles d'utilisation de X.400 pour l'EDI sont expliquées. Mais ces solutions qui sont basées sur l'utilisation du système de traitement de messages interpersonels, ne conviennent pas parfaitement à l'EDI. D'où une troisième solution décrit un système de traitement de messages, basé sur X.400, dédié à l'EDI. Finalement, l'utilisation possible d'un langage structuré, basé sur la notation ASN.1, pour définir et encoder des messages EDI utilisant X.400 est analysée.

Table of contents

Table of figures

Introduction

'Open' trading is the norm of the traditional business transactions. 'Open' trading means that any company can make business with any other with as less barriers as possible. The importance of Electronic Data Interchange (EDI) is growing in the economic world. EDI must also allow 'open' business but it needs syntax and communication standards to be implemented. We are living in a world of limited resources. Thus it is impossible, if many different standards are available, for each company to hold every different standards used by various trading partners. The use of internationally agreed standards is necessary for the further development of EDI. The goal of this work is to study a communication standard which is considered by the EDI communities as appropriate.

We will attempt to make a scientific study of the use of a standardized Message Handling System (CCITT X.400 recommendations) for carrying Electronic Data Interchange (EDI). We will first define and explain technical needs of Electronic Data Interchange (EDI). After that, we will explain the technical aspects and capabilities of CCITT X.400 recommendations. The EDI requirements for communication will be matched with the capabilities of X.400 Messaging Handling System. Three possible solutions for the use of X.400 for EDI will also be explained. At last, we will study the possible use of a structured language for defining and coding EDI messages using the X.400 Message Handling System as a communication medium.

The first chapter gives first an attempt to define Electronic Data Interchange (EDI). This is followed by a description of the implementation process of EDI and by four parts dedicated respectively to four basic technical needs for EDI. The first part concerns the EDI standards and describes the main points of UN/EDIFACT standards. The second part is dedicated to the translation softwares and their work through the EDI process. The third part describes the hardware implementation needed for EDI using personal computers, mini computers, mainframe computers and communication links. The last part of this chapter concerns the communication media. Because it is impossible to list all the usable protocols and communication means, we will only explain briefly two main solutions : the use of Value Added Network (VAN) services and the use of Open System standards.

The second chapter is dedicated to a rather technical approach of the X.400 recommendations. The X.400 recommendations have been created by the "Comité Consultatif International Téléphonique et Télégraphique" (CCITT). The CCITT X.400 recommendations describe a Message Handling System based on the Open System Interconnection (OSI) layer structure. The X.400 Message Handling System was the first application defined in the OSI layer seven. Even if use for other purposes is possible within the Message Handling System, only the exchange of person-to-person messages has been defined in X.400 recommendations. The X.400 recommendations have been published for the first time in 1984, these are the subjects of one of the two main parts of this chapter. An amended version of the X.400 recommendations have been published in 1988. The 1988 version contains additional facilities, is free of ambiguities and encompasses the OSI seven layer structure which was not completed in 1984. The 1988 recommendations are described in the second main part of this second chapter.

The third chapter gives first a list of EDI users communication requirements. This list was established according to a series of interviews of representative users, network providers and standard committees. A comparison of the capabilities of 1988 X.400 recommendations against these requirements are given. According to the results of the previous comparison, the implementation of an EDI Messaging System sending structured EDI messages and using the X.400 recommendations seems effective. But as described in the second chapter, only an InterPersonal Messaging System which allows the sending of unstructured messages has been standardized. Two solutions allowing users to exchange EDI messages using the existing EDI standards and X.400 recommendations are described in this chapter. The first one, called P0/1 solution, is standardized by the American Institute for Technical Standard and the American National Standard Institution. The second one, called P2 solution, is standardized by the Commission of the European Communities. Nevertheless, they are both seen by the EDI communities as interim solutions. The CCITT has worked on the creation of a messaging system dedicated to EDI using the X.400 recommendations. But due to its importance , it is the subject of the next chapter.

The forth chapter describes the main points of the CCITT EDI dedicated Messaging System. The work of CCITT in this field, called X.435 / F.435, allows the exchanging of EDI messages between trading partners' applications. A first part of this chapter describes the EDI Messaging System and introduces the main features. EDI messages replace numerous paper

documents. In doing so, they have legal consequences so the CCITT has defined the concept of EDI Responsibility. The second part of this chapter is dedicated to that concept . The last part is concerned with security in the EDI Messaging System, with naming, addressing and the use of Directory Service, with possible physical implementations and with the dedicated EDI Message Store capabilities.

The fifth and last chapter concernes the possible use of the abstract syntax notation ASN.1 for EDI using the X.400 recommendations as a communication medium. ASN.1 is standardized by the International Standard Organization (ISO) and by the CCITT as a X.400 recommendation. Even if the possible use of a structured language based on ASN.1 for EDI is at a very early stage, we think that from a scientific point of view its use must be studied. In the first part of this chapter the notation and the encoding rules of ASN.1 are explained. A usual method for defining EDI messages in ASN.1 can be found in the literature, we will explain in the second part why it is false and we will define what we consider a correct method. The third part makes a comparison of various EDI message encodings using UN/EDIFACT and ASN.1. The fourth part is concerned with the use of ASN.1 with X.400 recommendations. The last part gives the advantages and the disadvantage of this use.

Chapter 1 : EDI Explained

This chapter will deal with elements required for Electronic Data Interchange (EDI). First an introduction is given, including an attempt of definition and an overview of the implementing process of EDI. After the introduction the next parts will address the following issues : EDI standards, translation software, hardware and communication media.

1.1 Electronic Data Interchange

1.1.1. Definition

There is no official, nor legal, definition for Electronic Data Interchange (EDI). The following is an attempt at a definition by mixing all the major components that can be found in common other ones.

> Electronic Data Interchange is an electronic interchange of formatted data, following a standard, between trading partners' computer applications without any human intervention.

In that definition, several elements have to be defined or explained. This is done below.

- electronic interchange : Electronic interchange can be simply the exchange of magnetic tapes or better, using telecommunications.

- data : Data encloses commercial or administrative data contained in paper documents. Each paper document defines a message type. There are numerous message types : Purchase Order, Purchase Order Response, Purchase Order Change, Commercial Invoice, Transport message, Customs Declaration, Customs Response, Letter of Credit, Payment Order, Remittance Advice, etc, and also service messages : Control message, General message, Maintenance message.

- standard : A standard defines a standard way of representing the data in each message.

- computer applications : Computer applications are the systems generating, receiving and managing interchanges. Interchanges are made between computer applications rather than between computers.

- without any human intervention : The process is executed automatically without any human intervention. Several definitions only stipulate minimum human interventions, but in any case the concept of computer application to computer application interchange is essential.

1.1.2 Implementing EDI

This part deals with the implementing process of EDI in a company. The key ideas and the structure of the following elements are issued from [SITP 90].

The decision Process

The decision process leads to discover if the opportunity to 'go EDI' is good or not. The first task is the understanding by the management of the commercial aspects, of the standards arena of the EDI technology. The second point is to find out what process is subject to systemization. In the same time, an analysis of the costs (hardware, software, communication hardware and software, training, ...) has to be performed. Having done those points, benefits (reduce handling costs, reduce human resources, paper, ...) have to be quantified in terms of cash. Knowing all these informations, the management team is ready for a decision point.

Planning for EDI

If the decision to 'go EDI' is taken, then the company is ready to make a plan. An important point is that a company does not 'make EDI' alone. There are number of advantages to contact EDI professional associations. Decisions have to be taken considering the trading partners practice.

That plan can be seen as a set of subplans. These are : Technical plan, Commercial plan, Ongoing process, Interchange Agreement.

All these subplans are the subject of the following points :

- Technical plan

A system strategy (management and manufacture) has to be defined. Two different approaches are possible : additional sub-system or integration into the existing system. A communication medium has to be determined. Two major options are existing : the use of Value Added Network services or the use of Open System Interconnection (OSI) standards.

At that point, the company can procure, install and test the communication medium chosen.

A crucial decision is to find out which data standard best fits the company and trading partners needs. Some help can be found by consulting EDI users associations.

Now, a new structure of the system is drawn.

- Commercial plan

The introduction of EDI will change the business philosophy and organization. It is fundamentally important to co-ordinate the changes brought by EDI to the usual business. Several points of interest are listed below :

- Check business relationship
- Check legal requirements
- Specify the changes of business practice
- Publish the major decisions of the technical plan
- Set-up internal and external training
- Test acceptance of the procedures
- Check audit requirements
- ...

- Ongoing process

Considerations must be given to the future use of EDI. In order to maximise the benefits expected, a long term plan of development must be established.

- Interchange Agreement

It is necessary to give to all participants in EDI a clear understanding of the 'rules'. Interchange Agreement is a publication defining, as a check list, all the useful elements for an EDI Partner. These elements are :

- Standard supported
- Data element values for local and global use
- Processing cycles
- Legal requirements
- Terms and conditions of business
- ...

1.1.3 Basic technical needs

EDI has four basic technical needs :

- standard : A standard defines a structured manner for presenting data in transaction messages. It is necessary for EDI users to agree on standard defining formats and rules for encoding all elements from the message structure to the smallest item of information.

- translation software : A translation software has to transform information held in an in-house file to EDI standard, to transmit the EDI formatted information to the communication medium, to receive information from the communication medium and to translate it back into usable format. The translation software has to be integrated in the existing business applications.

- hardware : The hardware is the computer configuration used for holding the translation software. Computer used for EDI can be : personal computers, mini computers and mainframe computers.

- communication media : The communication media are the techniques used for transmitting the data from one computer to another. A medium is often telecommunication of files.

The following parts (from 1.2 to 1.5) of this chapter will deal with each of these needs.

1.2. EDI Standards

An EDI standard defines a structured manner and the rules for presenting data in transaction messages. All business sectors need to communicate some specialized information which are specific to that sector. But all sectors have to be able to communicate with each other. So an EDI standard must deal with the specific information needs of individual sectors and general trading.

In order to provide a high degree of compatibility between users, EDI standards have to be independent of :

- the type of software.
- the type of the EDI application (trade, transport, finance, ...).
- the hardware.
- the communication medium.

Firstly, the generic structure of an EDI message is explained. In a second section, the UN/EDIFACT standards are introduced.

1.2.1 Generic structure [VANG 89 I]

The various elements used for constructing and communicating business transactions are : messages, segments and data elements. Each of these elements has to be assembled according to standardized rules and formats.

A message is the smallest meaningful unit of information for an application, which can be processed. There are a lot of message types (order, invoice, ...). A message is organized into a group of segments.

A segment is a part of a message which relates to a single object (name and address, date and time, location, reference, ...). A segment is a logical group of data elements.

A data element is the smallest individual part item of information (road name, postal code, city name, quantity number, ...). It is held in a segment. Data elements are collected in various directories. Those directories provide a tag, a title, a description and a format for each data element. When it is needed a code list is given. The codes give defined values to data elements.

The figure 1.2.1 shows the generic structure of a message.

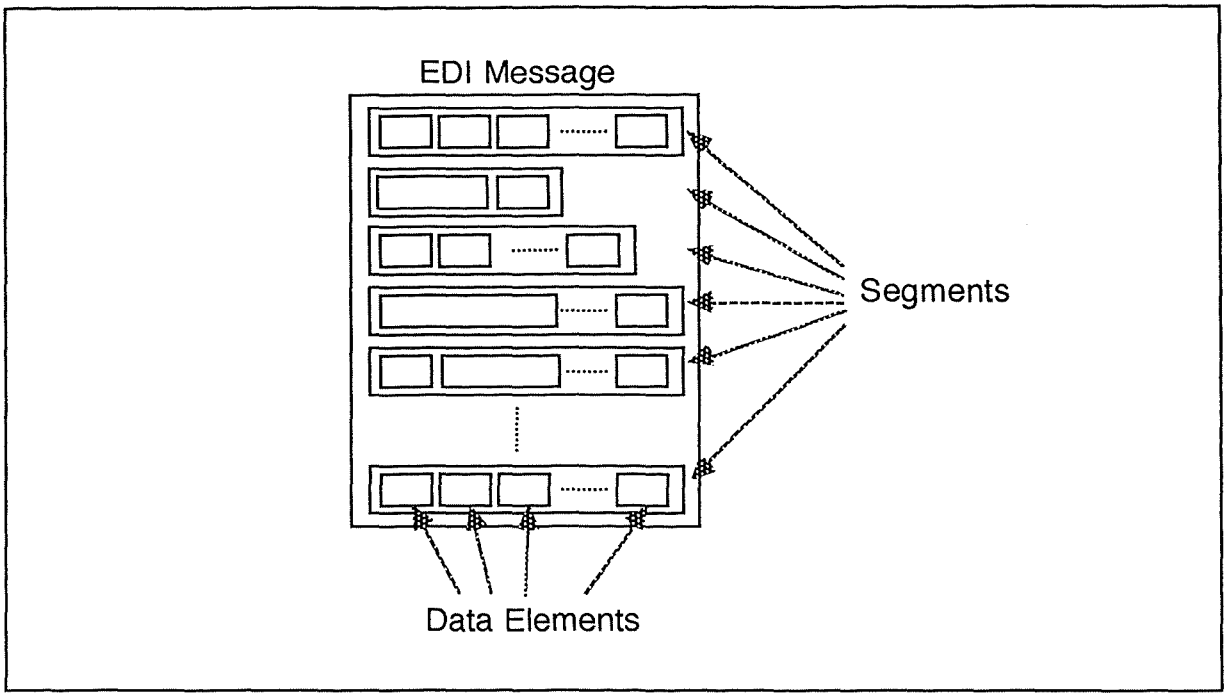

Figure 1.2.1 : EDI message generic structure

By analogy the message can be seen as a text, with the segments as the sentences and the data elements as the single words. But to link all these elements, "grammatical" rules are needed. The syntax defines those rules. It plays an important role in determining how data from in-house files is converted into standard messages and vice versa. An EDI syntax must cover various aspects. These aspects are the following :

- definitions of the various units of information (data element, segment, message, etc) ;
- rules and guidelines by which units of information are assembled into larger units ;
- character sets (upper/lower case letters, numbers and special symbols) ;
- characters which are used as delimiters;
- ways of enveloping messages with control information.

For creating EDI messages, message designers choose a syntax, a data element directory and a set of code lists and follow the agreed guidelines for structuring the segments and the messages.

1.2.2 UN/EDIFACT [SITP 90]

UN/EDIFACT (United Nations / Electronic Data Interchange for Administration, Commerce and Transport) is an international set of standards for EDI, supported by the United Nations.

The UN/EDIFACT standards are :

- UN/EDIFACT Data Elements Directory (EDED). It contains a subset of the United Nations Trade Data Elements Directory (UNTDED) (ISO7372).

- UN/EDIFACT Code List (EDCL). It contains a list of all code sets associated with coded data elements.

- UN/EDIFACT Composite Data Elements Directory (EDCD). It contains a list of the composite data elements with their component data elements.

- UN/EDIFACT Standard Segments Directory (EDSD). It contains a full description of all standard segments used in United Nations Standard Messages.

- UN/EDIFACT Messages Directory (EDMD). It contains a full description of all United Nations Standard Message types.

- UN/EDIFACT Syntax Rules (ISO 9735), They define in concise form the standard for formatting data elements and segments into messages.

- UN/EDIFACT Syntax Implementation Guidelines. They expand on some of the details of the syntax rules.

- UN/EDIFACT Message Design Guidelines. They are intended for message designers.

UN/EDIFACT standards were born out of UN/TDI and ANSI X.12. UN/TDI is the first European EDI standard supported by the United Nations Economic Commission for Europe. UN/TDI syntax is not greatly dissimilar from UN/EDIFACT. But there are some differences in the methodology of standard message design [FENT 89]. UN/TDI is adopted as a standard by a large EDI community in the UK. ANSI X.12 is the American standard defined by the American National Standard Institute. A detailed study of disparities between ANSI X.12 and UN/EDIFACT proves their close similarity [MOO 88]. UN/TDI and ANSI X.12 are national standards, but in a world of extended international trade, an internationally accepted standard is more appropriate. UN/EDIFACT has been created as an international standard. UN/EDIFACT, often called EDIFACT, retains the best characteristics of both previous standards in term of flexibility and efficiency.

It is difficult to expose all the concepts of UN/EDIFACT here. However, the major characteristics are summarized below. See figure 1.2.2.

In UN/EDIFACT terminology, a structured document is called a message. Messages are divided into a number of segments. Segments are made of data elements.

The UN/EDIFACT Data Elements Directory (EDED) defines whether a data element has to be alphabetic, numeric, alphanumeric or coded, and whether its length is fixed or variable. The occurrence of data elements may be mandatory or conditional within a particular segment. A data element may be simple, containing only one value, or composite, containing a number of values. The composite data elements are defined in UN/EDIFACT Composite Data Elements Directory ((EDCD). See the definition of the CITY NAME data element in appendix 1, the definition of the QUANTITY INFORMATION composite data element in appendix 2 and the definition of COUNTRY CODE data element with its codes in appendix 3.

The definition of a segment details which data elements are contained. A segment is identified in UN/EDIFACT Standard Segments Directory (EDSD) by a tag, made of three upper-case letters. The occurrence of a segment may be mandatory, conditional or multiple within a particular message. See definitions of the NAD and LIN segments in appendix 4.

The UN/EDIFACT Messages Directory (EDMD) specify a number of defined messages, called United Nations Standard Messages (UNSMs). See definition of the ORDER message in appendix 5. However, UN/EDIFACT allows the creation of non-standard messages from standard segments and data elements.

An interchange consists of one or more messages. If several messages of the same type are transmitted in the same interchange, they can be grouped in one Functional Group. An interchange can contain a number of Functional Groups, or alternatively a number of individual messages, but not a combination of both.

Service segments known as header and trailer segments are used for control purposes. They are :

- UNA : Service String Advice. Its use is conditional. The function of the UNA segment is to define the characters selected as delimiters and indicators in the rest of the interchange that follows. Usually, the apostrophe " ' " is used as segment terminator, the plus sign "+" as segment tag and data element separator, the colon ":" as component data element separator and the question mark "?" as release character.
- UNB : Interchange Header. The function of the UNB segment is to start, identify and specify an interchange.
- UNZ : Interchange Trailer. The function of the UNZ segment is to end and check the completeness of an interchange.
- UNG : Functional Group Header. The function of the UNG segment is to start, identify and specify a Functional Group.
- UNE : Functional Group Trailer. The function of the UNE segment is to end and check the completeness of a Functional Group.

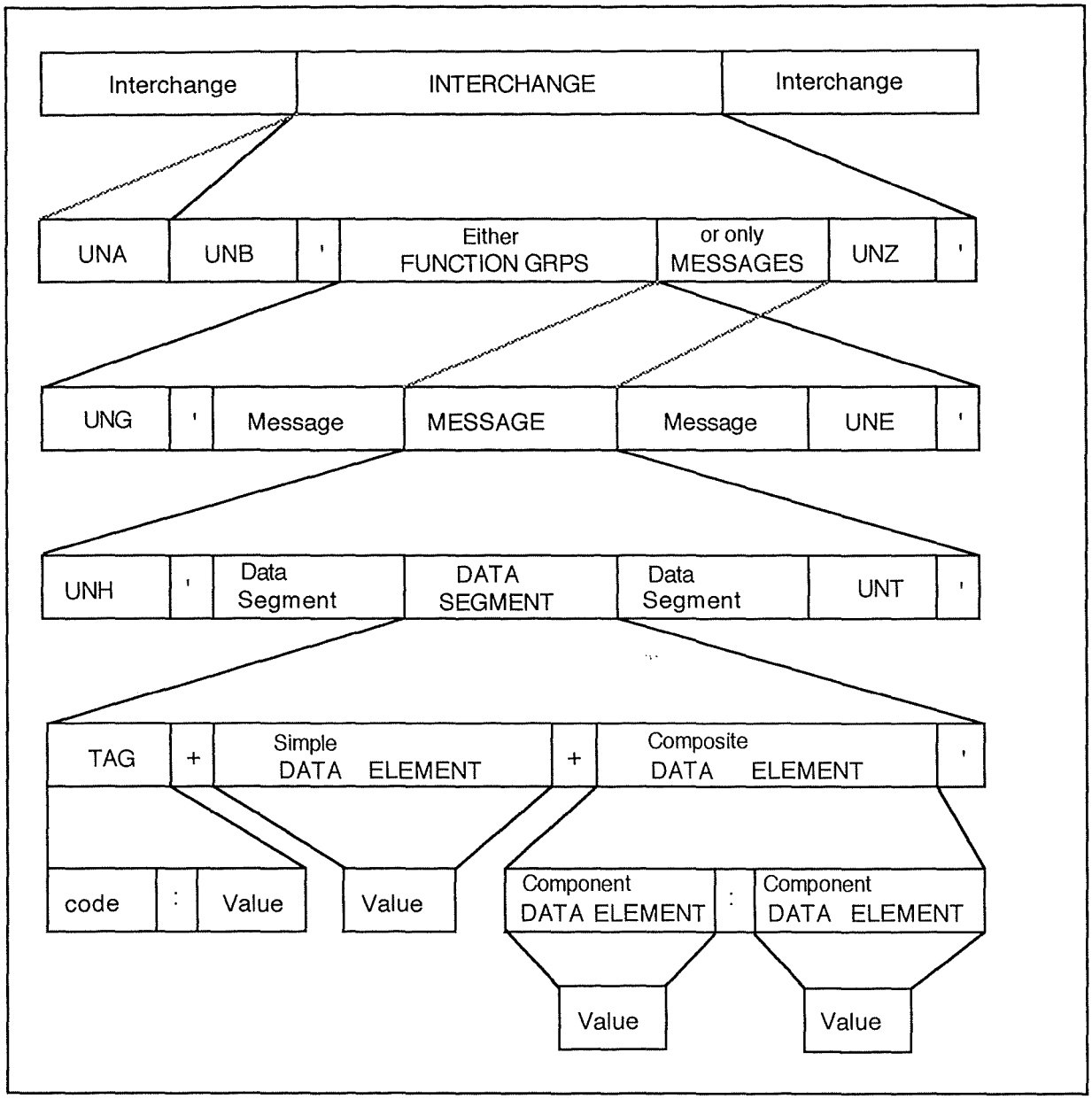

Figure 1.2.2 : UN/EDIFACT hierarchy from interchange level to data element level

- UNH : Message Header. The function of the UNH segment is to start, identify and specify a message.
- UNT : Message Trailer. The function of the UNT segment is to end and check the completeness of a message.

See definitions of the UNB and UNH in appendix 6.

UN/EDIFACT standards, amongst others (UN/TDI, ANSI X.12, ...), are based on character string delimiters between fields. Other standards could be based on the use of octet strings encoding rules, as ASN.1. Explanations about ASN.1 and its possible use for EDI are the subjects of chapter 5.

1.3 Translation software [WARD 89]

The translation software is concerned with the EDI process from the data extraction from the in-house application files to the sending of the EDI formatted messages and reverse. The term "translation software" could be misleading. Indeed translation does not suggest the range of capabilities which are provided by the translation software. A better term for translation software is management software. Nevertheless, in the following, the term translation will be maintained due to its large diffusion.

In order to define a translation software, this part will detail the critical functions of a translation software for EDI. These will be identified according to the three main steps of the EDI translation process. The steps are : data mapping, standard conversion and communication.

1.3.1 Data mapping

Data mapping is the first step in the translation process. It involves the data extraction from the in-house data files. Each company can have its own unique application system and its own unique data formats. The extraction consists in converting the data from the application system into a format that can be easily translated in the following step of the EDI translation process. The result of the conversion is stored in a file called a flat file. The terms "in-house file" and "flat file" are not standardized. Their meaning may vary in the literature. Nevertheless, in this work they are used as described in this part. See figure 1.3.1. The extraction of data in several in-house files and conversion of the data formats can be complex and may be too heavy to be executed by the next step.

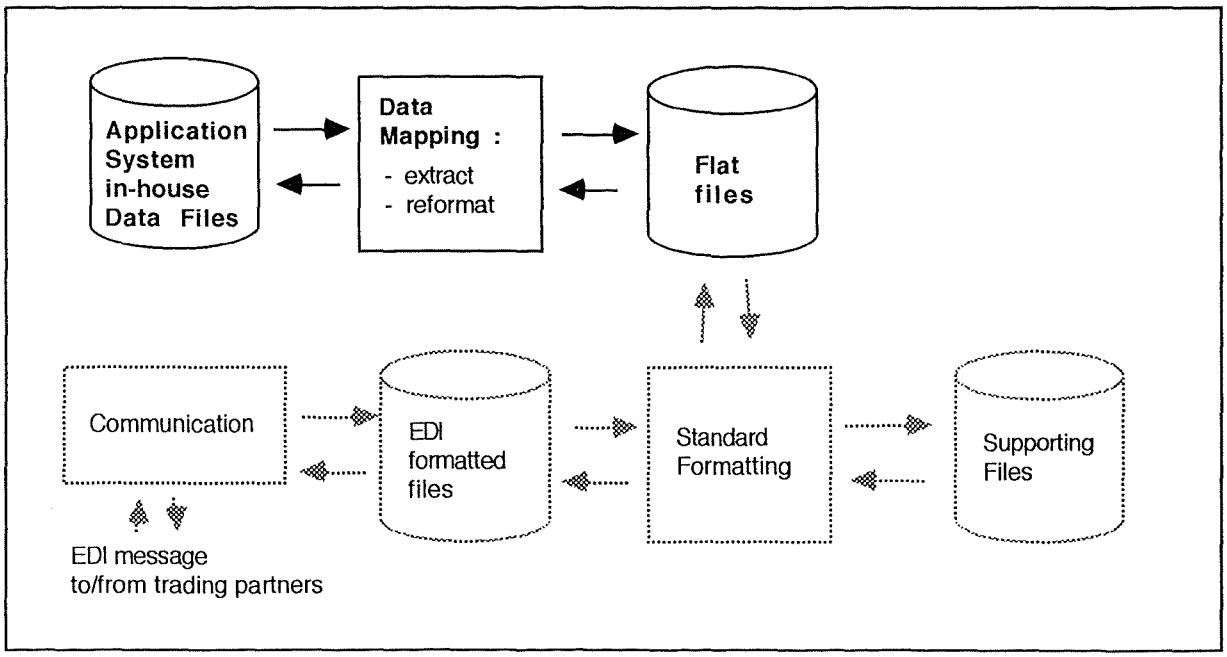

Figure 1.3.1 : Data mapping

Conversely, when an EDI message is received the data must be converted into a format that is compatible with the receiving company's application system. Thus mapping is the last step in the receiving process.

1.3.2 Standard formatting

Standard formatting is the second step. It is involved with the conversion of flat files into EDI formatted files. See figure 1.3.2. That process is executed using an EDI standard (for example : UN/EDIFACT, UN/TDI,...). Diverse companies and industries use a variety of standards, message types and versions. Each trading partner has particular needs. In order to be efficient a translation software has to be able to handle various types of messages (such as INVOICE, ORDER,...), to support various EDI standards and versions and to adapt itself to different trading partners' needs.

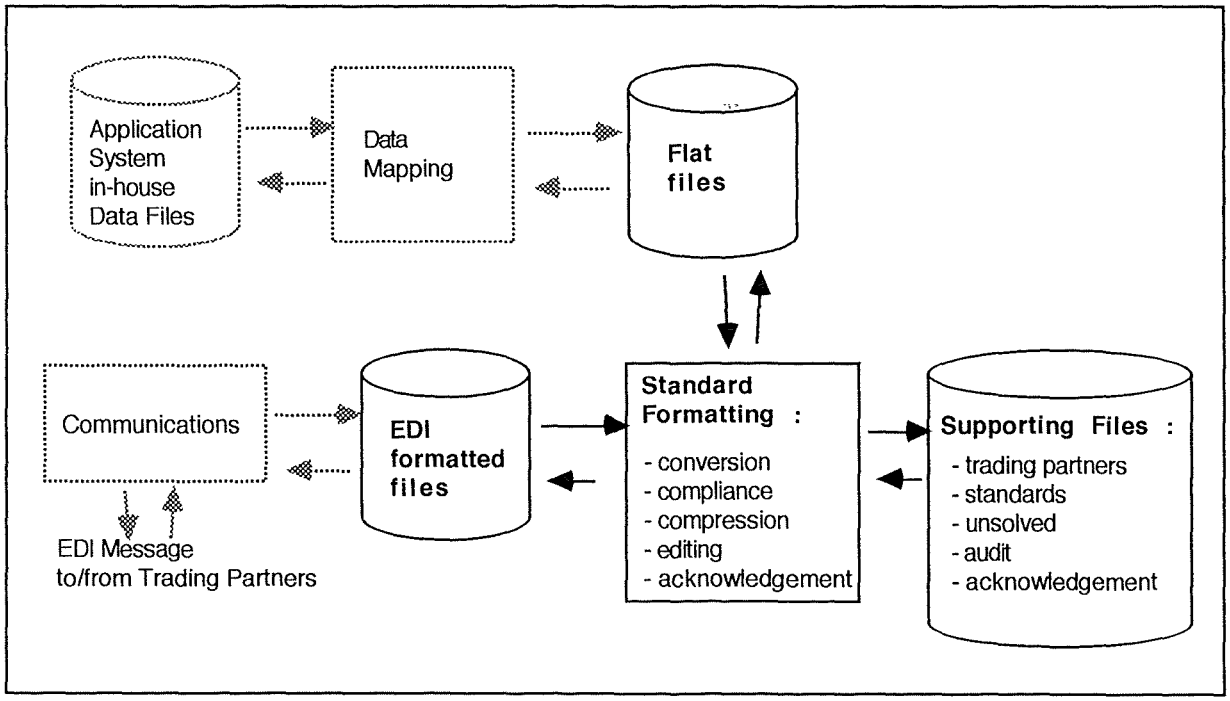

Figure 1.3.2 : Standard formatting

But the standard formatting step involves more than simply generating standard EDI messages.

It also includes :

- compliance procedures that verify the standard and the version being used, verify the syntax within the message, check the trading partner requirements and verify that the data is complete and correct. The compliance procedures use the standards file and the trading partners information file. They are both held in the supporting files.

- compression procedure that eliminates all the unnecessary blanks or zeros of outgoing data. The purpose of this compression is to reduce communicating price and time by reducing the quantity of non essential data that is transmitted.
- edition of data that can not be translated and need changes before being sent. This data is held in an unsolved file in the supporting files.
- holding of notifications certifying that incoming data was received and that outgoing data was transmitted. The audit file contains these notifications and is part of the supporting files.
- acknowledgement procedure that generates acknowledgements for incoming data. These are stored in an acknowledgement file held in the supporting files.

1.3.3 Communication

The communication is the last step in the translation software for an outgoing EDI message. Even if the communication aspects will be seen in more detail in part 1.5 of this chapter, some aspects will be explained here.

The communication software sends the outgoing messages coming from the EDI formatted files to the trading partners. Conversely it also delivers to the EDI formatted files the incoming EDI messages from trading partners. See figure 1.3.3. Usually the communication softwares provide several functions. These are : automate call to trading partners, protocols and control of the technical communication interface with trading partners.

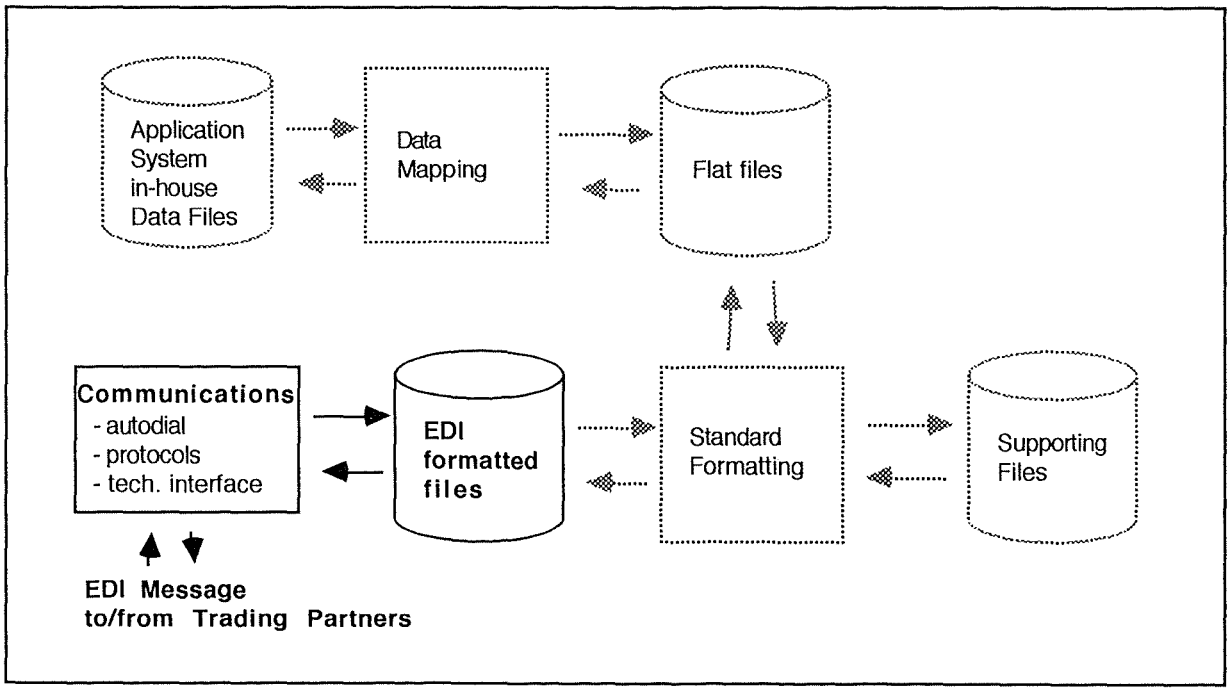

Figure 1.3.3 : Communication software

1.4. Hardware [CROU 90]

The purpose of this part is to describe the hardware platform. For simplification, we could say that there are three types of computer used : personal computers, mini computers and mainframe computers.

Implementing EDI requires at minimum, in addition to EDI translation software, a computer and a communication link.

Four different platforms are shown and explained in this part : stand-alone personal computer, front-end personal computer, front-end mini computer and integrated mini computer or mainframe computer.

1.4.1 Stand-alone personal computer

This configuration is made of a personal computer and a communication medium. The EDI translation software including communication software runs on the personal computer. The EDI data can be entered manually from the keyboard or using diskettes. See figure 1.4.1. EDI software packages are available and allow personal computers to act as stand-alone EDI solutions.

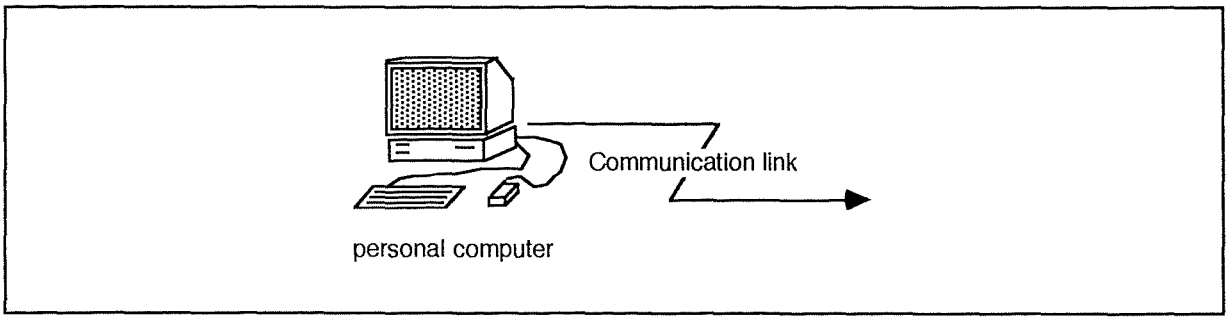

Figure 1.4.1 : Stand-alone personal computer configuration

The main advantages of this configuration are : low costs, ease of use, portability, user friendly, ease of implementation and additional functionalities.

The disadvantages are : slow system, impossibility of handling large volume of data, possibility of errors if the keyboard is used for entering EDI data, impossibility of handling multiple EDI standards or multiple communication protocols.

1.4.2 Front-end personal computer

In this configuration, a personal computer is used as a front-end to the organization's mini computer or mainframe computer. See figure 1.4.2. The standard formatting software and the

communication software run on the personal computer. The flat files are exchanged using a Local Area Network (LAN) between the personal computer software and the business application and data mapping softwares running on the mini computer or the mainframe computer.

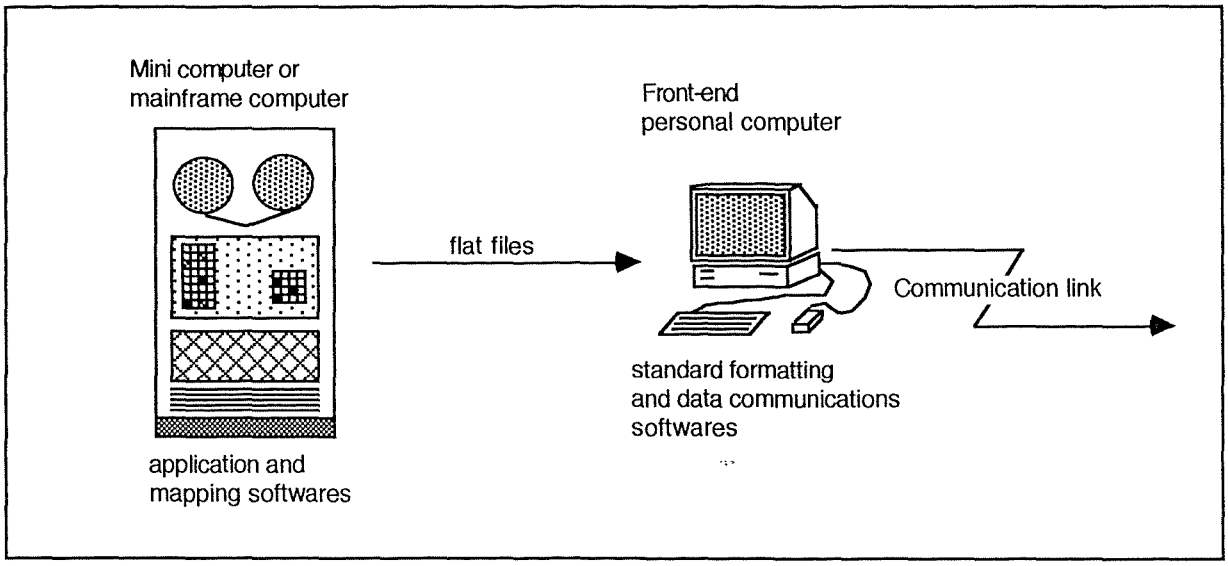

Figure 1.4.2: Front-end personal computer configuration

The advantages are : ease of use, cost advantages of the stand-alone configuration and possibility to handle higher data volumes.

The main disadvantage is : impossibility of handling multiple EDI standards or multiple communication protocols.

1.4.3 Front-end mini computer

This configuration is the same as the previous one but the personal computer is replaced by a mini computer. This offers the possibility to handle several EDI standards and communication protocols . See figure 1.4.3.

The two front-end configurations offer a low financial risk solution for pilot studies or interim step for an organization with limited resources. These configurations provide also a safe solution, because the front-end can protect from forbidden access to data on the computer.

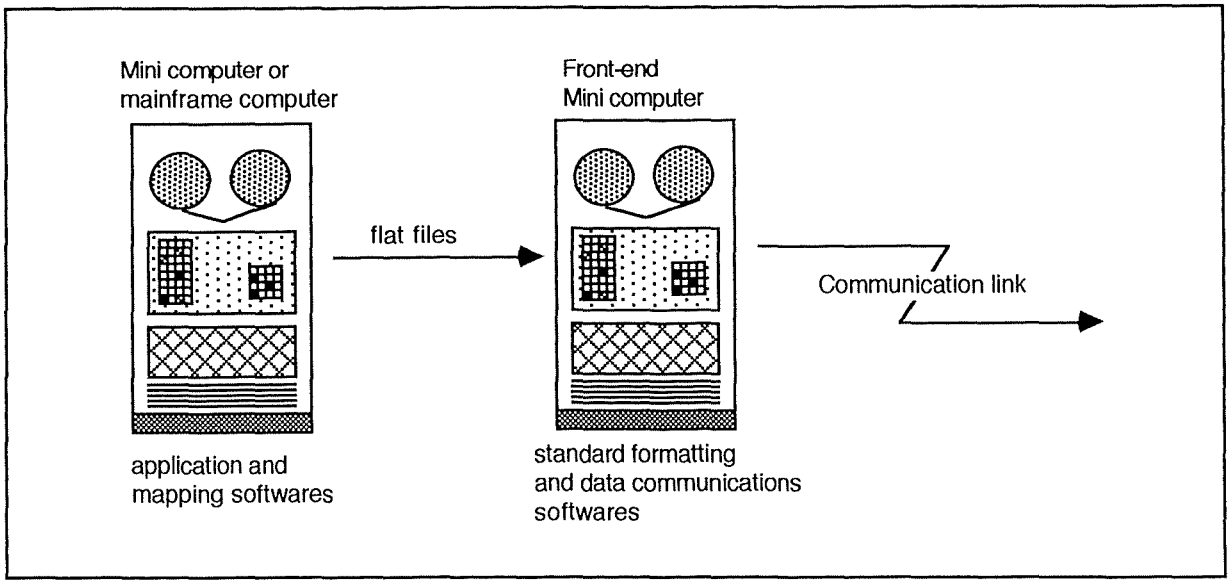

Figure 1.4.3 : Front-end mini computer configuration

1.4.4 Integrated mini computer or mainframe computer

This implementation is close to the first one, except that the personal computer is replaced by a mini computer or a mainframe computer. See figure 1.4.4. In this implementation all the softwares are running on the mini computer or the mainframe computer.

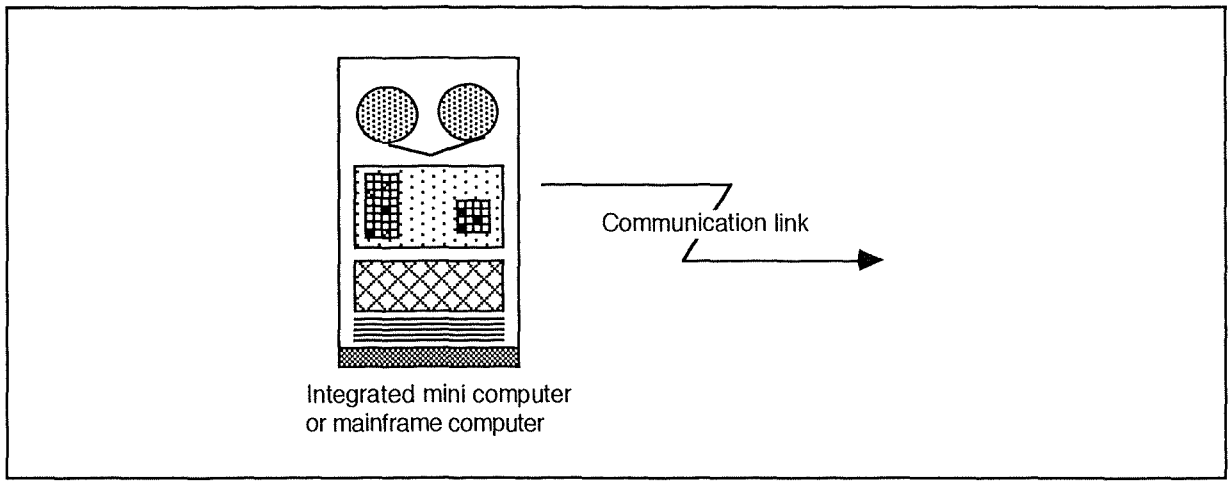

Figure 1.4.4 : Integrated mini computer or mainframe computer configuration

The main difficulty of using one of the two front-end solutions is the lack of integration of EDI and business applications. The integrated solution allows the mini computer or the mainframe to run the EDI and the business applications on the same hardware.

But integrated systems are more costly to buy and more complex to install, test and modify.

1.5 Communication Media

After the standard, the translation software and the hardware, the communication medium is the last basic element for implementing of EDI applications. The range of communication choices available are varied and wide. Even if some EDI applications still exchange magnetic tapes, telecommunications techniques are increasingly being used.

The number of networks and of proprietary or standard protocols is so enormous, that communication aspects are often seen as a huge field of confusion by companies which decide to "go EDI". As the number of trading partners increases, the number of different communication media increases. And it becomes impossible for each trading partner to support all the various protocols. In order to solve this problem, there are mainly two solutions : the use of services offered by Value Added Networks (VANs) or the use of Open System Interconnection (OSI) standards.

Firstly, the Value Added Network services will be exposed. This will be followed by explanations over various Open System solutions.

1.5.1 Value Added Networks services

"A Value Added Network provides value added services. A value added service is defined as one which allows the transmission and exchange of information or messages which are modified or acted on in some way. This action could take any of the following forms : the messages could be stored by the VAN operator for later retrieval, protocol conversion could be performed, speed conversion could be performed and the content of the messages could be changed to reflect some current information." [MACP 87]

Value Added Networks are also called Value Added Data Services (VADS). In this text, we will keep the term Value Added Network.

Certain Value Added Networks have been on the market for some time, but for general purposes. Their arrivals in the EDI world began in 1987. In the world, there are four major international VANs : GEISCO, IBM, EDS, Mc DONNELL DOUGLAS. In addition to these international ones, in Europe the number of VAN providers is growing, for example : AT&T ISTEL (UK), INS (UK), ALLEGRO (France), GSI (France), TELEVAS (Italy), OSIDES (Holland), TSI (Spain), ...

The various services offered by Value Added Networks may vary in details following the VAN provider. The following services are the basic ones:

- The VANs provide a message store in which the sender's messages are placed. This service allows the recipient to pick-up its messages according to its own convenience. Thus the recipient does not have to be on-line all the time.

- The VANs support several communication protocols (BSC, HDLC, X.25, ...), following the needs of users who are constrained by the hardware and the software system they use. A VAN can receive a message sent using a particular protocol from the sender, adapt that message to the recipient's protocol and forward the result to the recipient's message store.

- The VANs can check the message integrity and offer various other security services.

All those services are used by EDI users but are implemented for general communication purposes. Others are pure EDI dedicated. The EDI VANs usually offer the following services :

- The EDI VANs support several EDI standards and versions. A user can send a flat file, as described in the translation software part, to the VAN which translates it into a wished EDI standard.

- The EDI VANs can also perform the conversion of an EDI message from a particular EDI standard into another (for example : from UN/EDIFACT to UN/TDI).

- The EDI VANs provide access control to the message store and selective retrieval of EDI messages from the message store following EDI criteria.

- The EDI VANs can check that users do not receive EDI messages from unexpected trading partners nor wrongly formatted according to trading relationships and mutual exchange agreements.

- The EDI VANs hold audit files confirming the sending or the delivery of EDI messages.

And, even if the following services are not considered as value added data services, they are nevertheless provided by a number of EDI VANs.

- The EDI VANs provide their know-how in hardware, software, communication subjects or financial, organization, legal aspects.

- The VANs offer training courses for EDI users.

The VANs provide potential users or user groups a capacity for work. It is obvious that trading links can be built up more easily and quickly.

It is clear that all the services offered by the VANs are at a price. The price encloses the costs of using the facilities but also the costs of multiple connections to VANs. Indeed, the availability of interconnection of VANs is poor [VANG 89 II]. In order to be able to contact other trading partners, the lack of interconnection constrains the user to be linked to two or three, and perhaps more VANs.

The VANs have probably pushed the emergence of EDI, by giving a solution to the protocols incompatibilities [SITP 90]. Nevertheless the utility of VANs is based on those incompatibilities. Open System Interconnection can offer a simpler solution. That is the subject of the next point.

1.5.2 Open System Interconnection

Open System Interconnection (OSI) consists of a set of internationally agreed standards which define the way in which data is transferred between open systems. An open system is a system that can interwork with any other, without any reference to proprietary communication mechanisms. The OSI standards are developed under the supervision of the International Standards Organization (ISO). In addition the "Comité Consultatif International Télégraphique et Téléphonique" (CCITT) has also produced recommendations that are now adopted as part of OSI.

OSI is based on a reference model which is the framework on which the standards are themselves based. The reference model is made of layers, services and protocols. Those points are explained below. [TANE 89]

The communication systems are built on a number of separate layers. See figure 1.5.1. Each layer provides a defined function built on the use of the services offered by the lower layer. There are seven layers, shortly explained below :

Physical layer : The physical layer is concerned with the transmission of bits through a communication channel.

Data Link layer : The data link layer is concerned with the provision of a transmission free of errors.

Network layer : The network layer is concerned with the routing of information packets to their destinations.

Transport layer : The transport layer is concerned with the splitting of data received from the session layer into small units, passing these to the network layer and ensuring that the data arrives correctly at the other end.

Session layer : The session layer is concerned with the establishment of sessions between different machines, providing synchronization services for applications.

Presentation layer : The presentation layer is concerned with the syntax conversion of the information transmitted.

Application layer : The application layer is concerned with the applications themselves. Standard applications are : file transfer, electronic mail, remote job entry, virtual terminal, ...

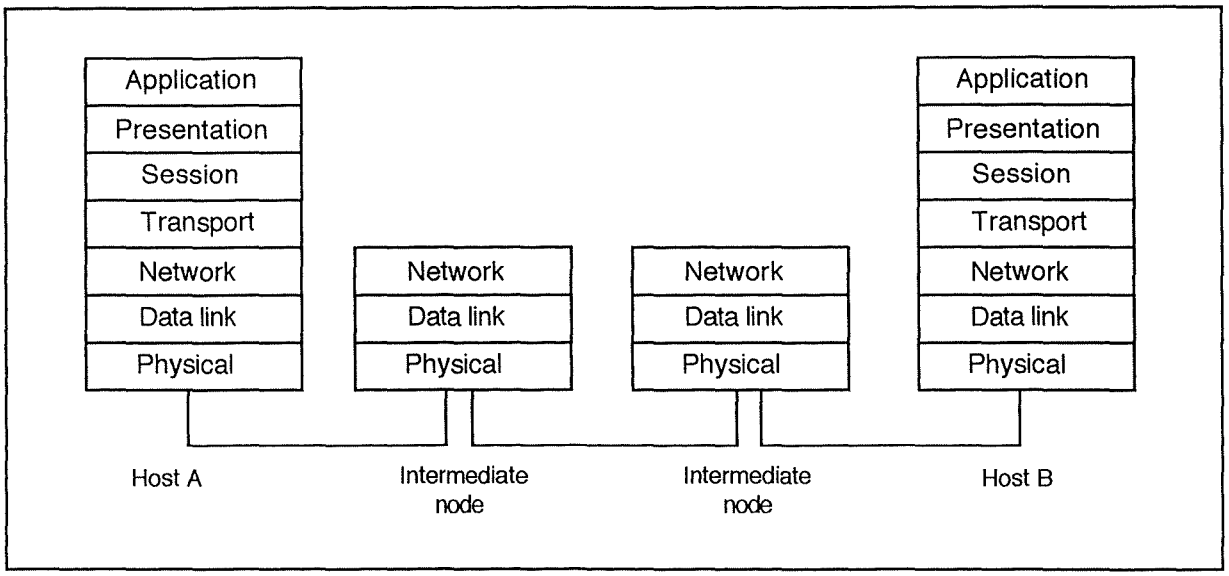

Figure 1.5.1 : OSI layer structure model

Each layer is described in terms of its services and protocols. See figure 1.5.2. A service is a set of primitives that the layer n-1 provides to the above layer n ; a protocol describes the dialogue between peer entities. Layers use protocols in order to implement their service definitions. The distinction between services and protocols will be explained and illustrated more in details in part 2.2.5.

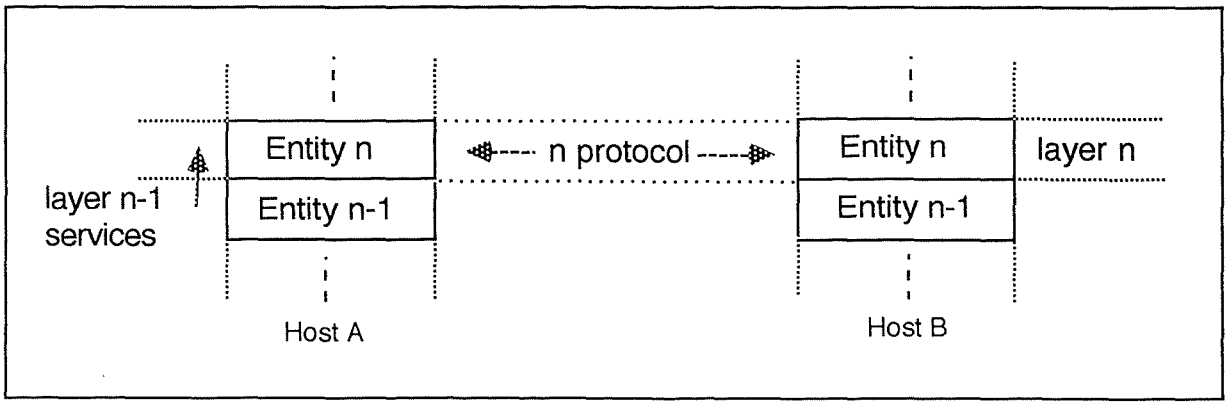

Figure 1.5.2 : OSI protocol and service model

The main advantages of OSI standards are that they are internationally agreed and published, allow inter operating between different systems and provide network and manufacturer independence. The disadvantages are that even if OSI lower layers are nearly available world-wide, PTTs do not always provide the most "up-to-date" standard [TOYE 89] (as example : 1988 X.400 is not provided by the Belgian PTT). Furthermore, scepticism on the success of OSI is widespread in professional and university communities. OSI outcome can be achieved only through implementation and experimentation and ... there were no "workable" implementations of OSI by mid-1989 [ROSE 90].

In the OSI layer structure model, two levels can be extracted, due to the characteristics of the services they offer. The first one is composed of the three lower layers, and can be called the network level. The second one, comprising the three upper layers, can be called the application level. The transport layer can be considered as a link between the two levels.

In the following, the possible solutions for EDI communication media will be explained considering the two levels architecture.

Network level

Currently, there are three solutions : the Public Switched Telephone Network (PSTN), the Leased Lines and the Public Switched Data Network (PSDN). Gradually the Integrated Services Digital Network (ISDN) becomes a fourth solution.

The Public Switched Telephone Network (PSTN) is used for ordinary domestic and business telephone calls. The connection of EDI applications to PSTN requires a modem. The connection is made by direct dialling. This solution is the simplest and the easiest. But its use must be fairly local, the number of transactions has to be small, otherwise PSTN use is costly.

If the number of transactions to the same trading partner is great, then a Leased Line can be a solution. A Leased Line is a permanent dedicated connection between two users. The line can be either analogue or digital.

The Public Switched Data Network (PSDN) has been developed for carrying packets of data. Many PTTs in the world provide this point-to-point analog communication network. The access to this network is done with a modem using the X.25 protocol and can use a Packet Assembler/Disassembler (PAD). See figure 1.5.3.

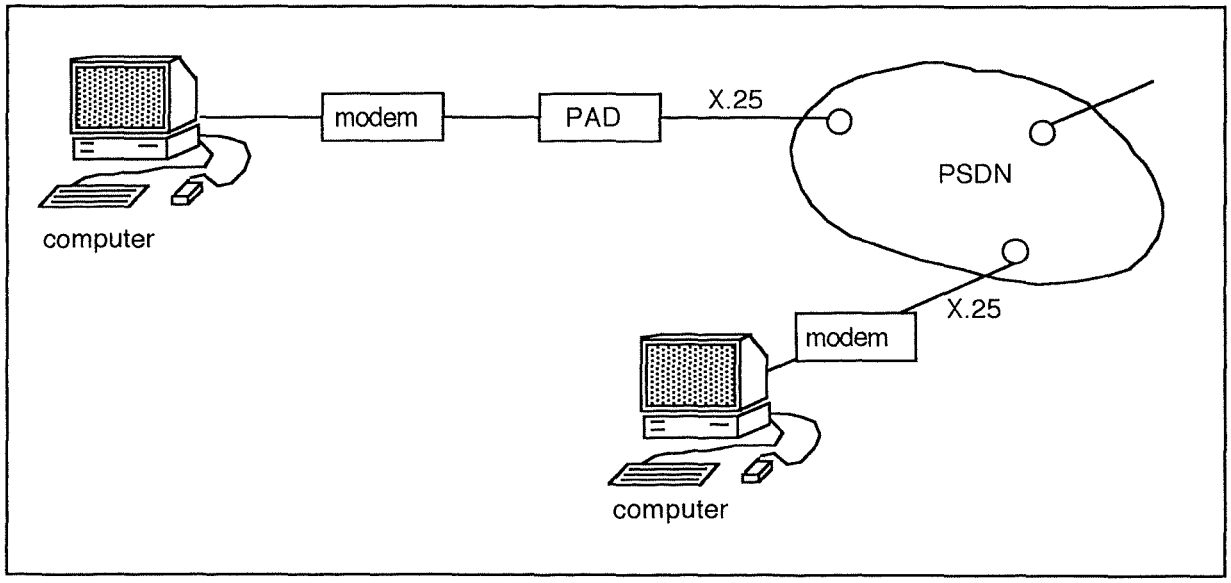

Figure 1.5.3 : PSDN accesses

The Integrated Services Digital Network (ISDN) is a network where voice and non-voice services are integrated. This system has been designed for voice transmission, data transmission, facsimile and video. Even if ISDN gradually becomes a used communication medium, we will not enter deeper in the subject.

Application level

Two major OSI applications seem to fit for EDI Interchange [HASE 89] : FTAM and X.400. Firstly, the File Transfer, Access and Management protocol (FTAM) and its use for EDI are shortly introduced in the following. In a second part, the X.400 Message Handling System is introduced. X.400 recommendations are the subject of the next chapter and its use for EDI is the subject of chapters 3 and 4.

File Transfer, Access and Management (FTAM)

The FTAM service and protocol contain a large number of functions. FTAM defines three different classes of functions : transfer, access and management.

The transfer class allows transfer of files between open systems. The access class allows inspection, modification, substitution and suppression of data in remote files. The management class allows creation, suppression and manipulation of remote files [ISO 8571]. For EDI, the transfer class is recommended, particularly by UK GOSIP (UK Government OSI Profile) [GOSIP 90]. An EDI Interchange can be carried as a file in a FTAM transfer mechanism. In the definition of FTAM [ISO 8571], the transfer class specifies no type for the file content. Each file is transferred using its own presentation syntax. This can be UN/EDIFACT, ANSI X.12, ...

The use of FTAM is considered in the EDI community as the solution when the EDI interchange is very large, when a point-to-point data transfer is required for security or when the interchange must take place in real time [SITP 90]. In appendix 7, a list shows how FTAM satisfies EDI user requirements for communication.

X.400 Message Handling System

X.400 is a set of CCITT recommendations. Correlations can be made between CCITT and ISO (MOTIS) recommendations, the ISO equivalent of X.400 to X.430 is ISO 10021-1 to 7. Chapter 2 gives explanations about the X.400 recommendations.

X.400 is seen in the EDI community as the OSI solution for transferring EDI Interchange in a store-and-forward manner. Several X.400 solutions carrying EDI character string syntaxes, as UN/EDIFACT, are currently defined (P0/1 US interim solution or P2 European interim solution). Those solutions are the subject of the chapter 3. A new additional standard defines a dedicated system for EDI in the X.400 world. This standard is called X.435 / F.435 or P_{edi}. This new standard is explained in chapter 4.

Chapter 2 : X.400 Explained

2.1 Introduction to Electronic Messaging System

A messaging system is a system which allows communication between users of the system by exchange of messages. Submission or delivery of messages to or from the system are done even if the recipient is not present. The message transfer is the responsibility of the messaging system.

The important messaging system features which influence its use are :

- The number of users who can be reached by the system.
- The efficiency of the delivery (successful and in a short period of time).
- The quality of the user interface.

A computer based messaging system is a messaging system implemented on one or a number of computer systems. The message transfer is performed by one or more computer systems. If the system is held on one computer system with one application (program), then the number of reachable users will be limited to those who have an access to the same computing facility. If the system allows interconnection of various applications on a number of computer systems then the number of users grows but with an increasing interconnection complexity.

Considering the need for interconnection, a standardized messaging system is the best issue. Based on the Open System Interconnection (OSI) layer architecture, the CCITT has created a Message Handling System called X.400. The CCITT X.400 recommendations define a medium-independent electronic messaging system on a store-and-forward basis between computer systems. X.400 was the first application defined at the OSI layer seven.

The X.400 recommendations provide a model of Message Handling System (MHS) with entities, services, facilities and interconnection protocols between the entities based on the OSI reference model. This allows interconnection of heterogeneous computer system.

Currently X.400 is internationally defined and agreed. Due to the urgency of creating an open Message Handling System, CCITT has published a first version of the X.400 recommendations in 1984. This first set of recommendations is detailed in the next part 2.2 "1984 X.400 recommendations". But at that time the structure of the application layer was not completed and for several other reasons, an amended publication has been done by the CCITT in 1988. This second set of recommendations is analyzed, with a presentation of the new layer seven structure in the part 2.3 "1988 X.400 recommendations". As far as known, a 1992 amended set of recommendations will not be published. However, a dedicated recommendation for an EDI Message Handling System has been standardized in 1990. That recommendation is the purpose of the chapter 4.

The part 2.2 is an abstract of [X.400 84] and the part 2.3 is an abstract of [X.400 88].

2.2 1984 X.400 recommendations

2.2.1 Message Handling Abstract Model

The Message Handling Environment is made of a Message Handling System (MHS) and numerous users. See figure 2.2.1.

Users

In the X.400 jargon, a user is a terminal which is using the MHS to send and receive messages. It is not a human user.

Message Handling System

The Message Handling System is a store-and-forward system. Thus it is not a real time system and therefore it is not suitable for real time applications. X.400 defines two types of elements required to provide the Message Handling System, namely the User Agents (UAs) and the Message Transfer System (MTS).

User Agents (UAs)

User Agents provide the user interface to the MTS. They handle all the protocols for interacting with the rest of the Message Handling System. The UAs accept messages from the users and then employ the MTS to submit them to the recipients. See figure 2.2.1.

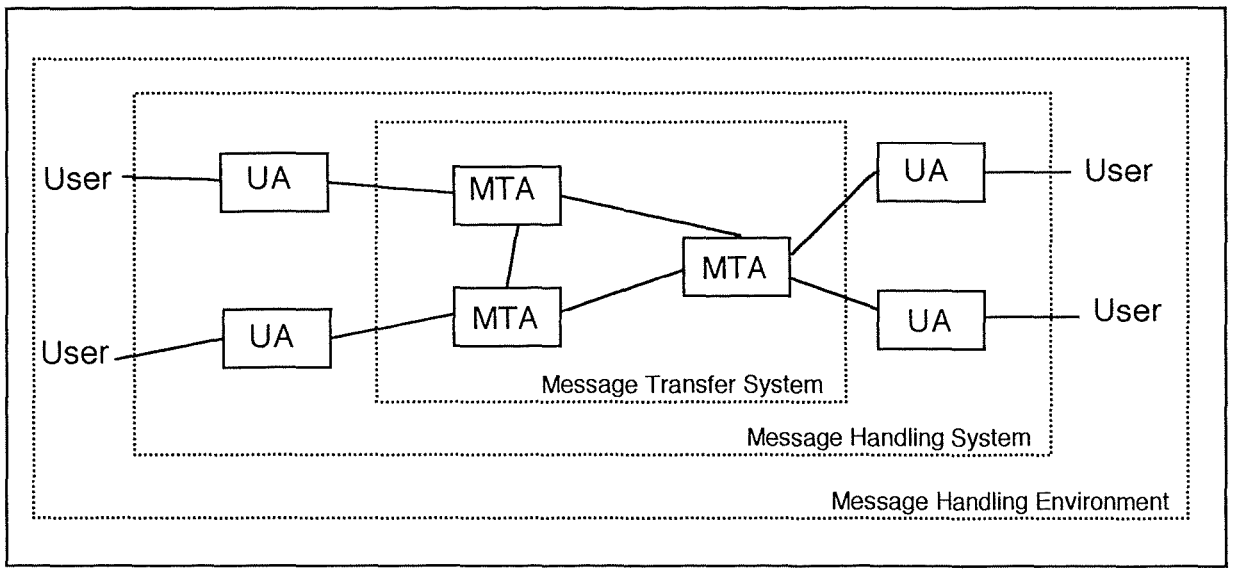

Fig 2.2.1 : Message Handling Environment

Message Transfer System (MTS)

The Message Transfer System is provided by the Message Transfer Agents (MTAs). The MTAs are interconnected and they route messages and deliver them, like the routing offices within the public postal offices, in a store-and-forward manner. Some MTAs only transfer messages to and from other MTAs, they are known as "relaying MTAs". MTAs are usually implemented on mini computers or mainframe computers.

2.2.2 Message Handling System Model [SIMO 89]

In the Message Handling System Model, there are five distinct phases. The figure 2.2.2 which is described below, shows the MHS Model.

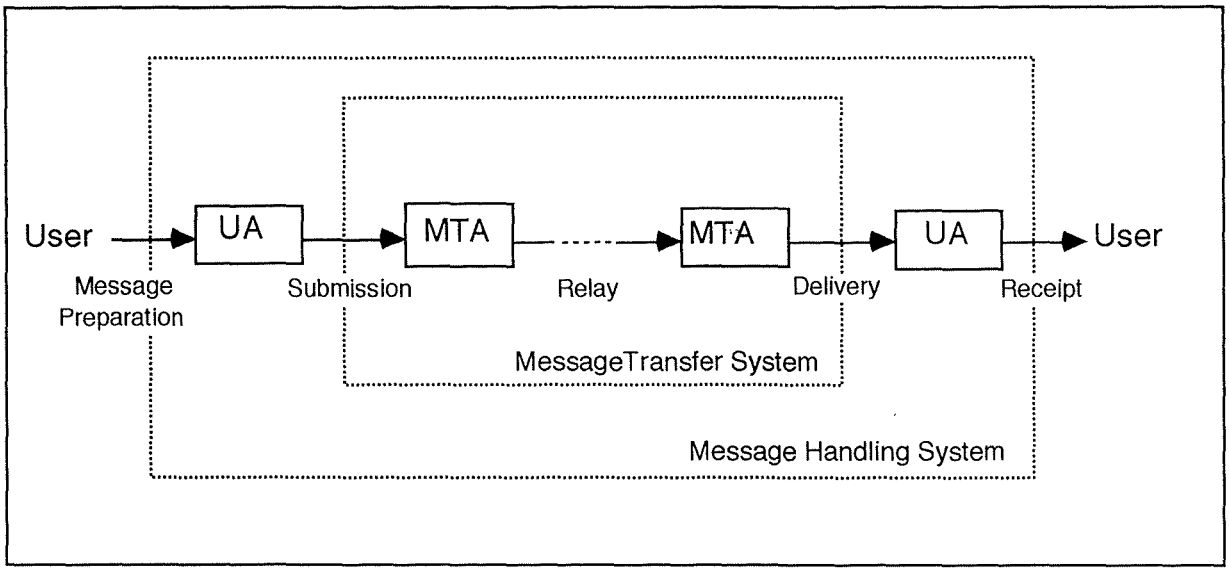

Fig 2.2.2 : Message Handling System model

Message preparation :

The user through interaction with the "originating UA" will produce a message for dispatch.

Message submission :

The UA will then submit the message to its MTA with the envelope information required for routing.

Message relay :

The "originator MTA" will create the envelope and will analyse the information on the envelope in order to determine the route across the MTS. The message receives a message identifier and a time stamp is affixed. If the recipient's User Agent is not local then the message is forwarded to another MTA.

Message delivery :

The message will progress through the system until it reaches the MTA associated with the recipient. The MTA will "deliver" the message to the recipient's UA.

Receipt :

The transfer is completed when the user accesses the message in order to read it.

2.2.3 Message types

For all message types a key idea is the distinction between the content and the envelope. The content conveys user information itself and it is transparent to the MTS. The envelope conveys information used by the MTS and other information about the message. The envelope contains parameters for transporting and interpreting the messages. The parameters can be classified in four groups : addressing, delivery, conversion and security.

There are three message types :

1- User messages :

The user message is composed of an envelope and a content. A user message content is the information sent from one user to another or to a group of users. See figure 2.2.3.

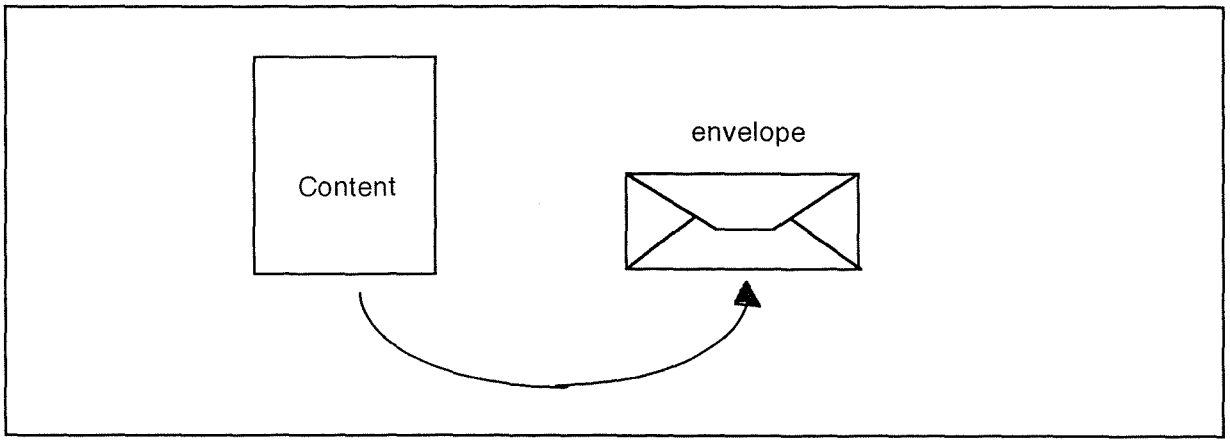

Fig 2.2.3 : User message

The content may be arbitrarily long and may contain anything the originator chooses to include.

The envelope parameters are quite numerous. It will be heavy to explain all of them in this text but we can say that these parameters allow the user to send its message with great freedom. The parameters are, for example : Message Identification, Delivery Notification, Content Return Request or Content Type. Content Return Request can be useful, in case of failure, for people

who send or receive a lot of messages. The parameter Content Type is used for defining the type of the message which could be : text, facsimile, digitised voice, etc, ...

2- Probes :

Probes are service test messages. They are meant to test deliverability of messages. The purpose of sending a probe is to find out if a destination is reachable. They are made of only an envelope without any content. See figure 2.2.4.

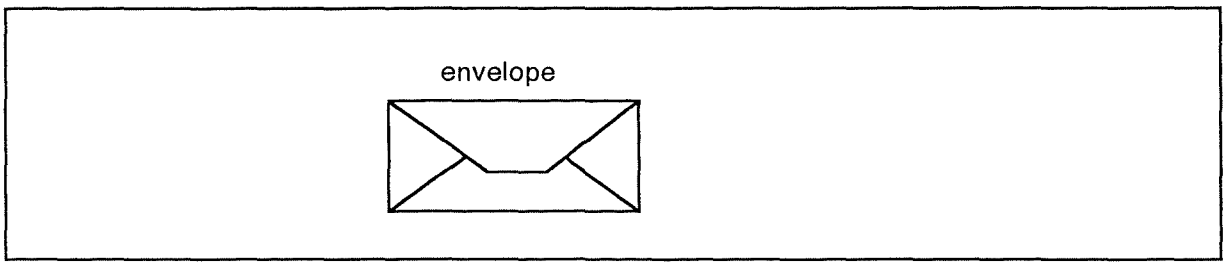

Fig 2.2.4 : Probe

The parameters of a probe envelope are close to those of the user message. The probes are always followed by a delivery report.

3- Delivery reports :

Delivery reports are generated by the system. They are sent back to the originator to report on whether its message was delivered or not. The delivery reports are composed of a delivery report envelope and of a delivery report content. See figure 2.2.5 .

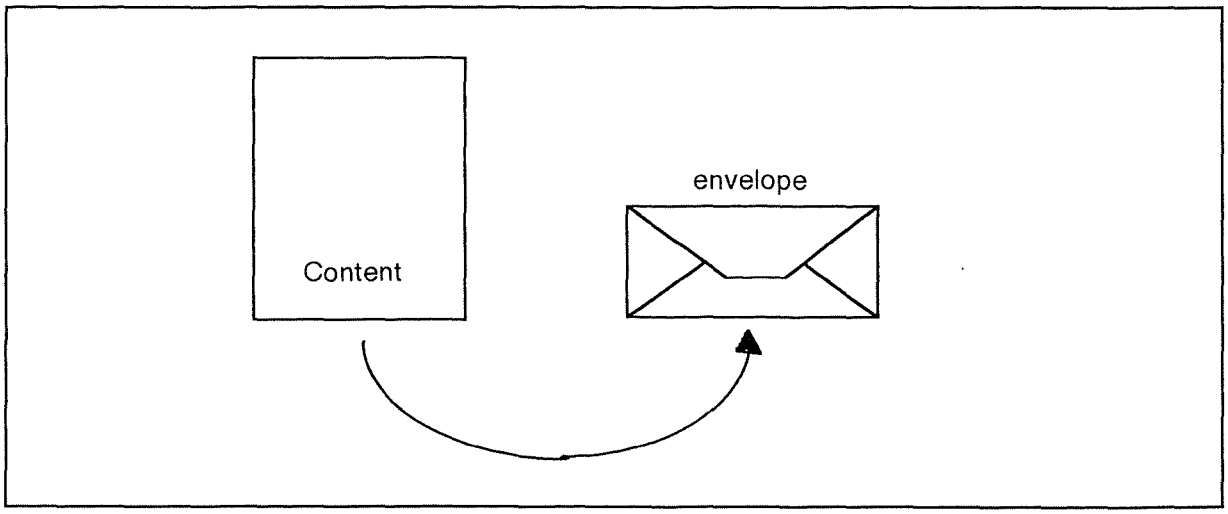

Fig 2.2.5 : Delivery report

The content holds the information of the delivery report itself. In order to provide a feel for the content, the following is a rough list of fields : Originator Message Identifier, Trace Info, Arrival Time, User Message Content, Billing Info and of course the report of Delivery Time or Non-Delivery Reason.

Delivery reports convey information on previously submitted user messages or probes. A probe is always followed by a delivery report or a non-delivery report. A user message, if asked by its originator, is followed by a delivery report or a non-delivery report. See figure 2.2.6 .

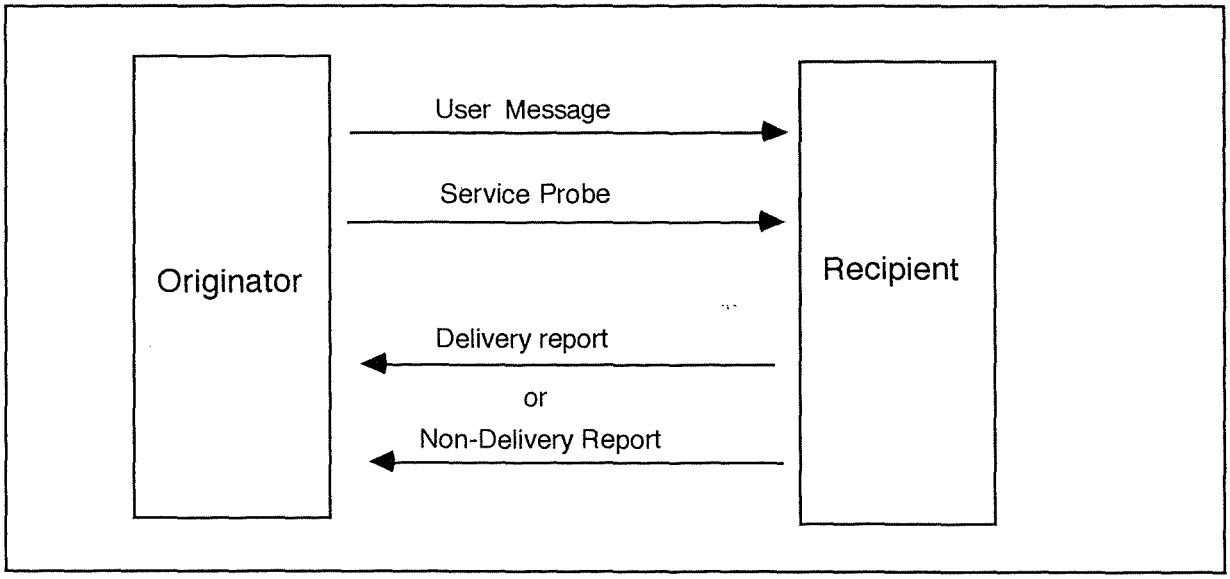

Fig 2.2.6 : Delivery report model

2.2.4 MHS configurations

In order to understand the realization of X.400 MHS, several different physical configurations are shown. There are a variety of ways to provide access to X.400. The various configurations illustrated in figure 2.2.7 are as follows :

- UA and MTA are co-located in the same processing system. The user (I/O device) interacts with the UA (See figure 2.2.7.a, 2.2.7.b syst A).

- UA and MTA are not co-located in the same processing system (See figure 2.2.7.b syst B and C). The user (Intelligent Terminal) interacts directly with the MTA through its built-in UA.

- A mixed configuration, with co-located and not co-located UA and MTA, is also possible (See figure 2.2.7.c).

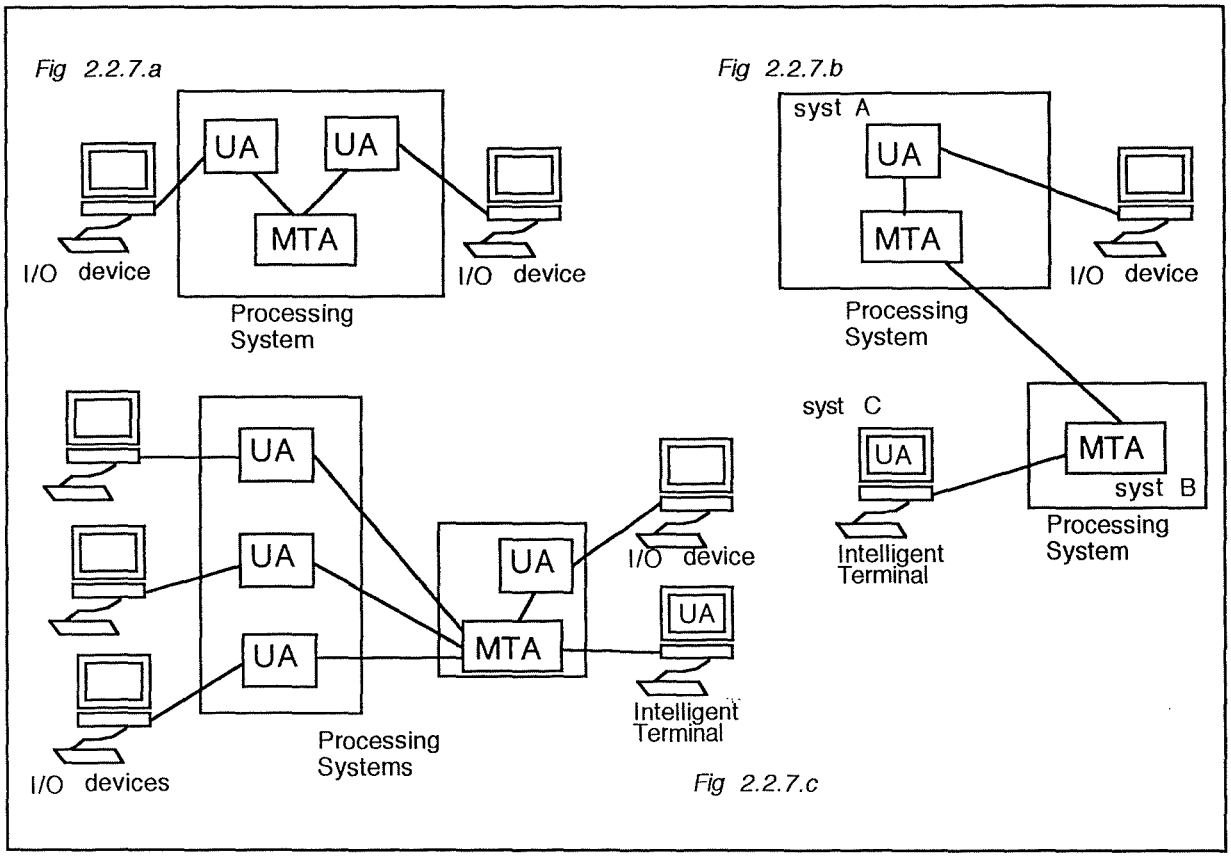

Figure 2.2.7: MHS physical configurations

2.2.5 MHS in the OSI environment

The 1984 X.400 MHS application layer seven is divided into two sublayers : Message Transfer Sublayer (MTL) and User Agent Sublayer (UAL). The figure 2.2.8 illustrates the configuration of the previous figure 2.2.7.b in the OSI environment.

The Message Transfer Sublayer offers the Message Transfer Service to the User Agent Sublayer. The MTL is made of cooperating Message Transfer Agent Entities (MTAEs) and Submission and Delivery Entities (SDEs). Two MTAs cooperate using two MTAEs and a MTA cooperates, using its MTAE, with a remote UA which uses a SDE. They are using cooperating rules, called protocols, to interact (P1 and P3).

The User Agent Sublayer uses the services provided by the MTL and offers the User Agent Service to the user application. The UAL is based on cooperating rules to interact (Pc : Content Protocol).

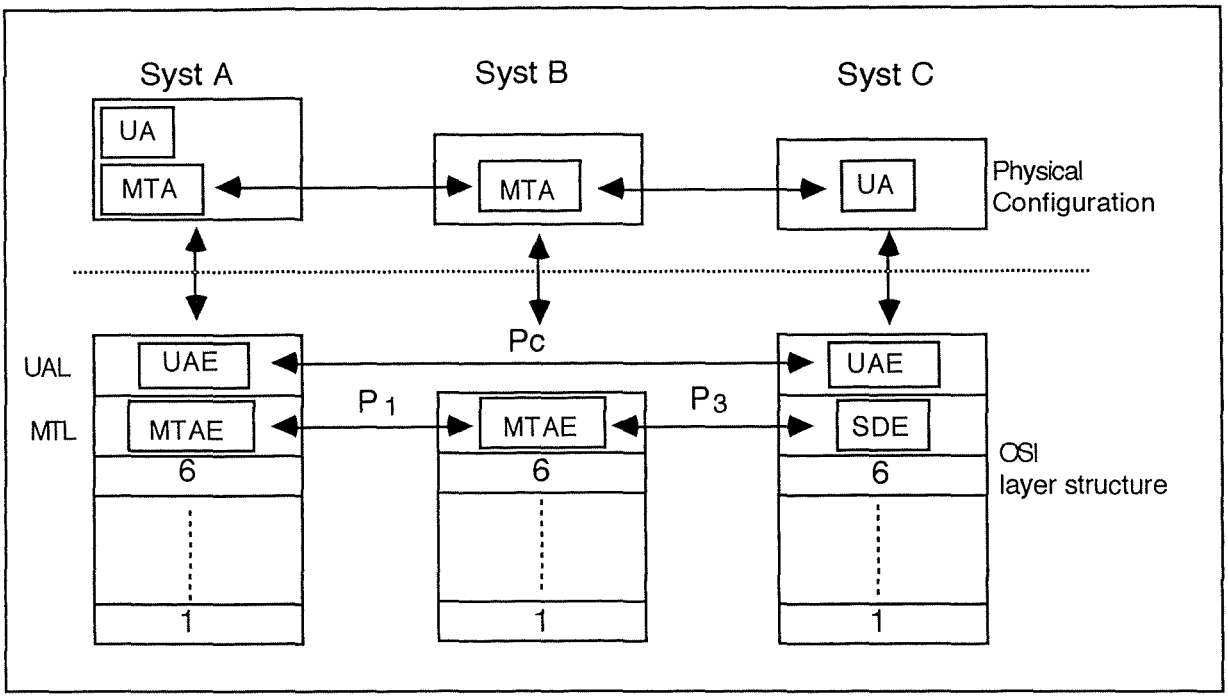

Fig 2.2.8 : Physical configuration on the OSI layer structure [SIMO 89]

The UAEs and MTAEs services and protocols will be detailed below. But before that part, let us make a little clarification about services and protocols . They are distinct concepts, although they are frequently confused. "A service is a set of primitives (operations) that a layer provides to the layer above it. The service defines what operations the layer is prepared to perform on behalf of its users, but it says nothing about how these operations are implemented. A service relates to an interface between two layers with the lower layer being the service provider and the upper layer the service user. A protocol, in contrast, is a set of rules governing the format and naming the frames, packets or messages that are exchanged by the peer entities within a layer. Entities use protocol in order to implement their service definitions. In this way, the service and the protocol are completely decoupled". [TANE 89]

2.2.6 MHS Services

The Message Transfer Service (MT Service) provides the Message Transfer Sublayer (MTL) services, called the service elements to the upper sublayer (User Agent Sublayer) and the User Agent Service (UA Service) provides the User Agent Sublayer (UAL) service elements to the user. Each of these service elements performs a specific function within the MT Service or within the UA Service.

Message Transfer Service (MT Service)

The Message Transfer Service is only concerned with the envelope information unless there is a request for conversion. It allows the UA to access and to be accessed by the MTS in order to exchange messages. The service elements offered to the UA Sublayer consist of five groups. The Basic Group is the only one which is not optional.

The service elements groups are :

- Basic service elements : Basic service elements are concerned with the access operations from the UA to the MTS (Submission -, Delivery of messages, Content type, Submission - and delivery time stamp, Coding information and Non-delivery notification).
- Submission/Delivery service elements : Submission/Delivery service elements are concerned with the Multi-destination delivery, Delivery information (deferred, alternate recipient, priority), Delivery notification and Return of content in case of failure. These service elements are optional.
- Conversion service elements : Conversion service elements are concerned with all conversion purposes. Conversion can be required for example if the recipient device is not using the same encoding type. These service elements are optional.
- Probe service elements : Probe service elements are concerned with the submission of a special test message. Capabilities which can be tested are : path through recipients, message size, content type and encoded information type. These service elements are optional.
- Information service elements : Information service elements are concerned with alternate recipient under specific circumstances and with the holding of delivery for a period of time because the UA is not ready to accept the message. These service elements are optional.

User Agent Service (UA Service)

The User Agent Service provides facilities to the user for accessing the MHS. There is one User Agent Service for each specific user task or application. They have to be able to interface with the service elements provided by the underlying sublayer. The User Agent provides all the service elements of the MTL and additional service elements for specific needs of the application or of the user.

2.2.7 MHS Protocols

In 1984 X.400 recommendations, three MHS protocols were defined at the application layer of the OSI model.

They are :

P1 : The P1 protocol is defined between MTAs. It provides the message envelope functions between MTAs.

P3 : The P3 protocol exists between MTA and SDE. It allows communications between a MTA and a remote UA.

Pc : The Pc protocol is defined between UAs. It is a generic protocol which defines the content of messages for each particular user application. In 1984 X.400, Pc for InterPersonal Messaging System is called P2. P2 will be detailed in part 2.2.8.

The first part will deal with the protocols used at the Message Transfer Sublayer level : P1 and P3, and the second one will explain the User Agent Sublayer protocol : Pc.

Message Transfer Sublayer : P1 and P3 protocols.

As described before, the Message Transfer Sublayer is made of cooperation of MTA Entities (MTAEs) and of Submission and Delivery Entities (SDEs). The purpose of the MTL is to provide UAE means to transfer and deliver messages within a time period and to process encoded type conversions on contents if necessary.

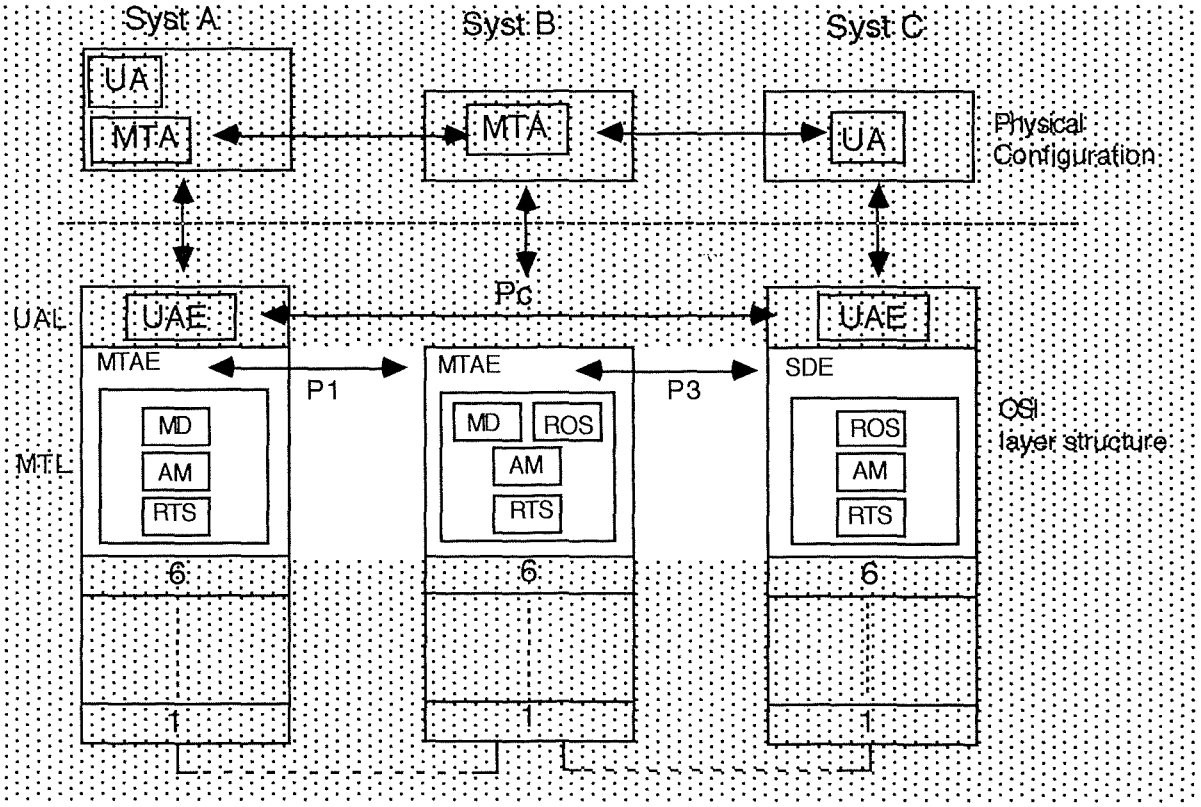

Fig 2.2.9 : Protocols P1 and P3, MTAE and SDE [SIMO 89]

The Message Transfer Agent Entity is made of several parts which are responsible for specific tasks. They are shown in the figure 2.2.9.

They are listed below :

Message Dispatcher (MD) : The MD uses the P1 protocol in order to allow cooperation between two MTAEs in two different open systems.

Remote Operation Server (ROS) : The ROS provides remote operations used between the MTAE and the SDE. The ROS uses the P3 protocol.

Association Manager (AM) : The AM maintains information on remote adjacent MTAs and SDEs, schedules the calls to adjacent MTAs and SDEs and controls the calls using the RTS.

Reliable Transfer Server (RTS) : The RTS provides a simplified interface with the layer 5 (session layer) which is the layer responsible for establishing and maintaining a relationship between user applications. From the user point of view the user application message is sent completely, exactly once or not transfered. There is no need of the presentation layer 6, because presentation is defined in X.400 using X.409 recommendation notation (ASN.1).

The Submission and Delivery Entity (SDE) is made of the same parts as the MTAE excluding the Message Dispatcher.

P1 Protocol

The P1 protocol defines the provision of service elements which requires cooperation between two MTAEs in different open systems. For that reason, P1 determines the content of the envelopes and how the Message Dispatcher uses them for providing the MTA service.

P1 fixes the envelope content. There are three different envelopes relating to the three different messages : user envelope, delivery report envelope and probe envelope. For each envelope type, the protocol defines all the fields contained.

P1 provides the basic message envelope functions between MTAs. P1 determines how the Message Dispatcher deals with incoming messages and probes from local UAs and messages, probes and reports coming from remote MTAs. P1 defines how the Message Dispatcher delivers messages and notifications to local UAs and how messages, probes and reports are routed to adjacent MTAs. P1 determines how a delivery report message is filled in with information about the delivery of a user message or of a probe.

P3 Protocol

The P3 protocol allows the UA which is remote from its MTA to reach the MT Service. P3 is made of a set of remote operations supported by remote operation protocol implemented on ROS. The remote operations are defined in a language which allows definition of operations, results and errors.

User Agent Sublayer : Pc protocol.

The UAE cooperation is defined by Pc. Pc is a class of protocols. Every instance of Pc defines a syntax and semantic of the message content for each special class of UA. See figure 2.2.10.

In the 1984 X.400 recommendations, only the InterPersonal Messaging System has been defined. The instance of Pc is called P2 . The message content is defined in an abstract syntax notation called ASN.1 (described in X.409).

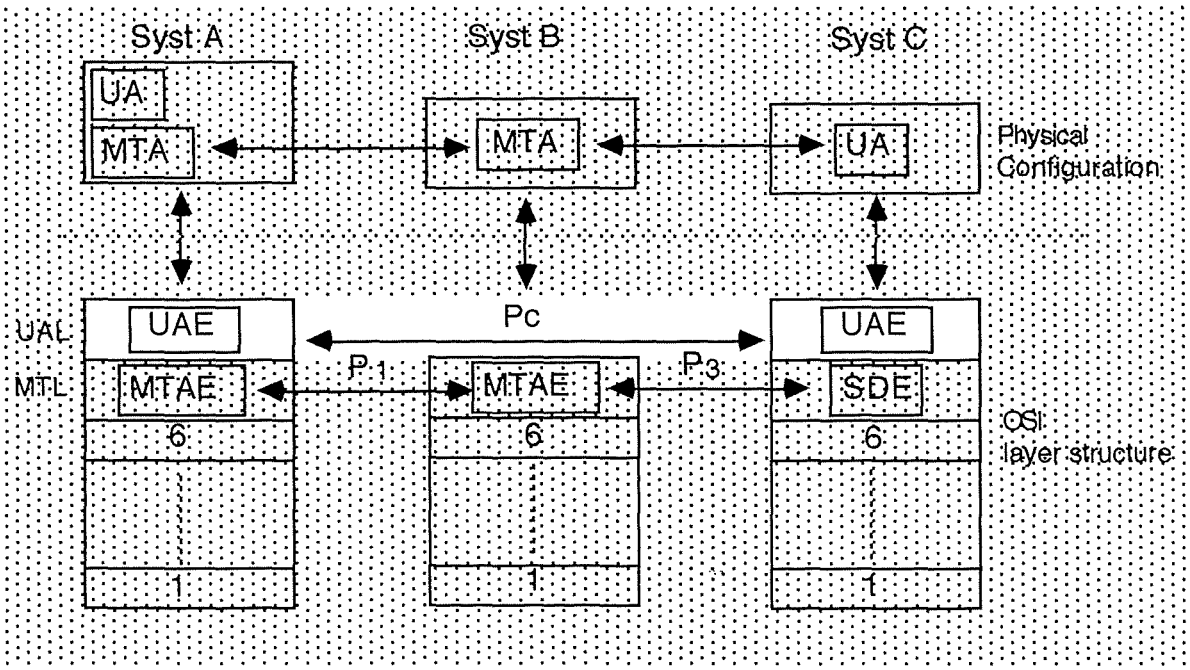

Fig 2.2.10 : Protocol Pc [SIMO 89]

2.2.8 InterPersonal Messaging System

InterPersonal Messaging (IPM) is the only user service and class of UA defined in the 1984 X.400 recommendations.

Components :

The InterPersonal Messaging System (IPMS) is made of the Message Transfer System and specially dedicated UAs. These UAs are called IPM-UAs, they allow person-to-person communication.

These elements interact to offer an InterPersonal Messaging Service, which allows users to exchange messages. Interworking with telex and telematic services (facsimile, teletext) is also provided.

The figure 2.2.11 shows all the components of the IPM System.

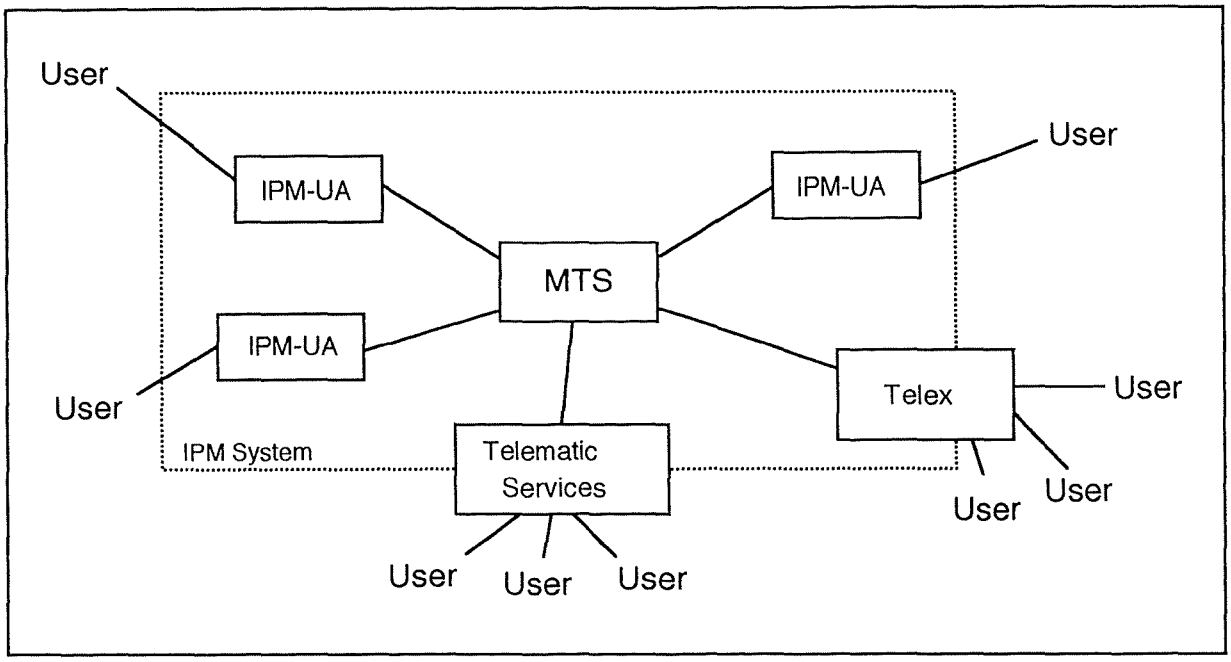

Fig 2.2.11 : IPM System

Services and Protocol

- IPM-UA Service

The IPM-UA Service provides service elements to the Users. The service elements are provided by the IPM-UAE. They are built on the MT Service. See figure 2.2.12. The MT Service is available only for the sublayer above the MTL. The MT Service is useful for Users. So the MT Service is provided by the UAL to the Users. The MT Service is described in part 2.2.6.

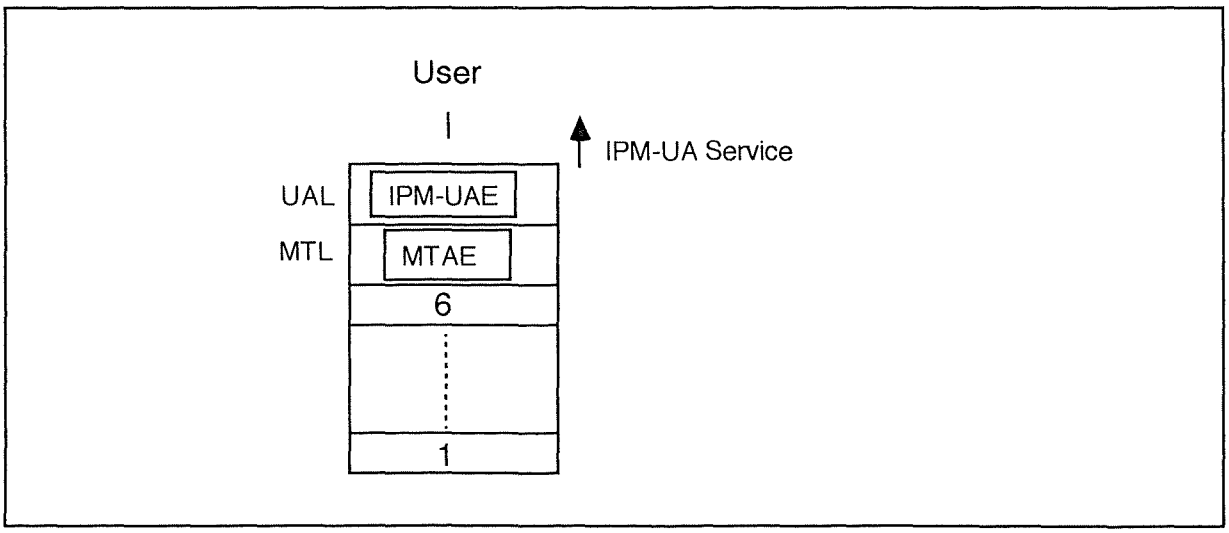

Fig 2.2.12 : IPM-UA Service

The IPM-UA service elements can be classified into groups, these are :

- Basic service elements : The IPM-UAE provides the Message Transfer Basic service elements, an IP-message Identification service element and a Type Body service element to the user.
- Submission / Delivery service elements : The IPM-UAE provides the Message Transfer Submission / Delivery service elements to the user.
- Conversion service elements : The IPM-UAE provides the Message Transfer Conversion service elements to the user.
- Query service elements : The IPM-UAE provides the Message Transfer Probe service elements to the user.
- Information service elements : The IPM-UAE provides the Message Transfer Information service elements to the user.
- Co-operating IPM-UA Action service elements : The IPM-UAE provides receipt/non-receipt, blind copy recipient indication, auto-forward indication service elements to the user.
- Co-operating IPM-UA Information Conveying service elements : The IPM-UAE provides service elements concerned with information about the message (subject indication, Expiry date indication, ...).

- IPM-Protocol :

The protocol P2 defines cooperation between IPM-UAEs. See figure 2.2.13.

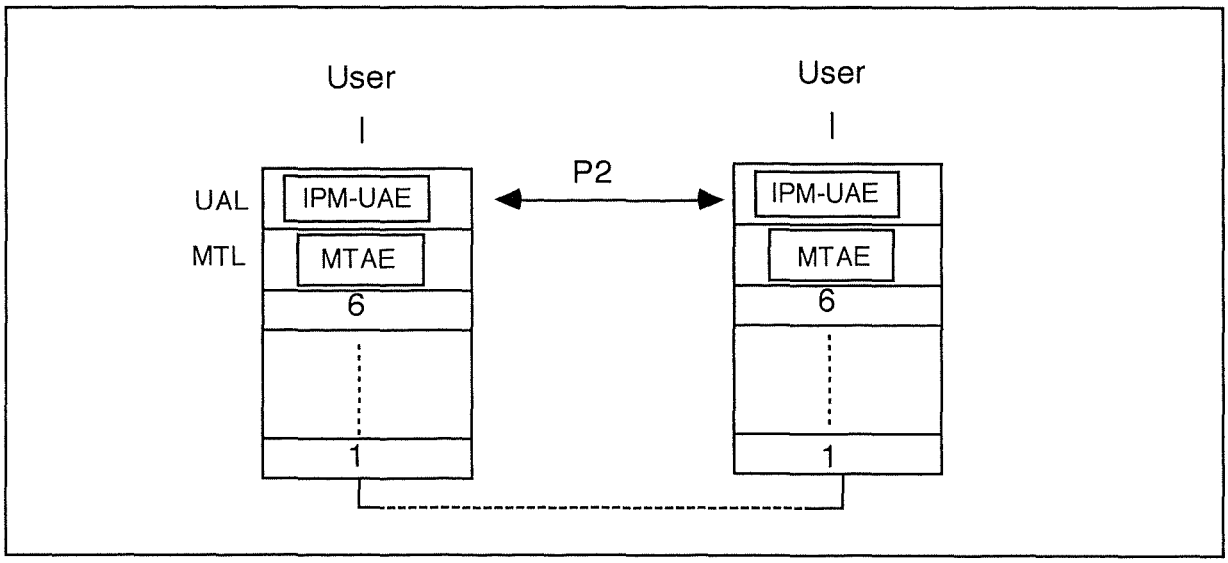

Fig 2.2.13 : P2 protocol

The messages exchanged between IPM-UAs are called IP-messages. P2 defines the semantics and the syntax of the IP-message content. There are two message types : user-ip-message and report-ip-message. They are described in CCITT X.420 recommendation using the X.409 syntax. See figure 2.2.14.

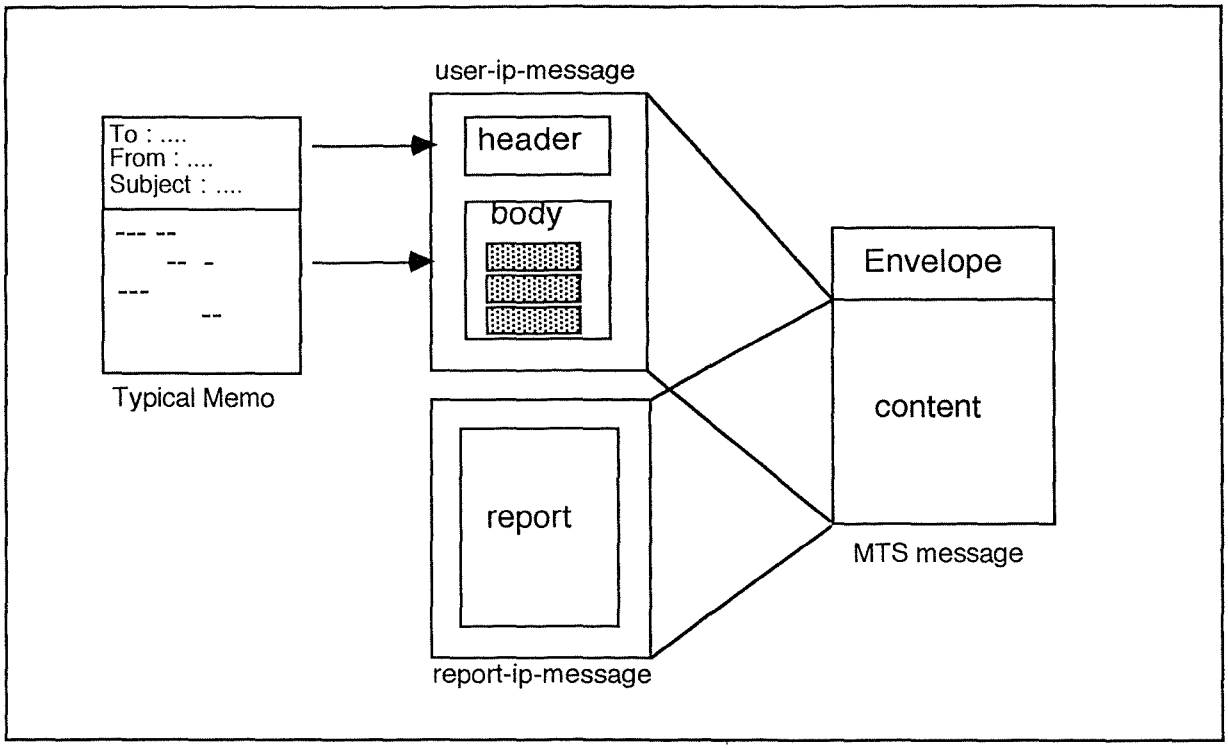

Fig 2.2.14 : InterPersonal messages [SIMO 89]

The user-ip-message is made of a header and a body. The header contains the general information about the message. The body is the information itself. The body is made of different body parts which can be of different types : IA5Text, Teletext, Telex, Videotext, digitised Voice, G3Fax, ...

The report-ip-message contains receipt/non-receipt information about a previously submitted IP-message. A distinction has to be made between delivery/non-delivery reports and receipt/non-receipt messages. The delivery/non-delivery reports are concerned with the delivery of messages from the MTS to the UAs. But the report-ip-messages are those, concerned with the receipt/non-receipt of messages from IPM-UAs to Users.

2.2.9 X.400 Naming and Addressing

In order to simplify naming and addressing, X.400 has created a user-friendly addressing system, the concepts of Originator/Recipient (O/R) Name and Address were introduced. It is important to specify the difference between an O/R name and address.

- an O/R Name is a descriptive name of an actual human user. Because an O/R Name identifies one and only one user in the MHS, a name can be used for specifying the recipient. But using an O/R Name for designing a recipient will require the MTS to perform a directory look-up function in order to determine the address.

- an O/R Address is a descriptive name for a UA which allows the MTS to locate the recipient in the messaging environment. Every O/R Address is an O/R Name, but not every O/R Name is an O/R Address. The use of an address implies that the MTS will not have to use the directory to find a recipient.

The standard attributes for designing an O/R Address (See example in figure 2.2.15) are:

country name.
administration management domain (ADMD) name.
private management domain (PRMD) name.
organizational name.
organizational unit name.
common name.

country name.	= BE
ADMD name	= RTT
PRMD name.	= FUNDP
organizational name	= INFO
organizational unit name.	= STUDENT
common name	= EDI

Fig 2.2.15 : Example of O/R Address attributes

In 1984 X.400 recommendations, there are two O/R Address forms defined. Form 1 for identifying MHS users only and Form 2 is for teletext, facsimile, telex or any other telematic services. Attributes between square brackets are optional.

These forms are shown below :

Form 1 (architectural base attribute set) consists of :

Var 1 : Country name
ADMD name
[PRMD name]
[Personal name]
[Organization name]
[Organization unit name]
Domain Defined Attributes.

Var 2 : Country name
ADMD name
[Domain Defined Attributes]

Var 3 : Country name
ADMD name
X.121 address.

Form 2 : (terminal oriented base attribute) consists of :
: X.121 address

The MHS supports Domain Defined Attributes (DDA) which are used privately within a domain. They allow additional information not defined in X.400.

It will be possible in the future to send a message to a recipient only by specifying the O/R Name. The future means the implementation of the CCITT X.500 Directory Service. These directories will provide a unique address of location and this will be used to route the message through the MTS. When all the X.500 Directory Service will be interconnected, as hoped, a world-wide Directory Service will be available.

2.2.10 Structure of the 1984 X.400 recommendations

The CCITT X.400 set of recommendations is comprised of eight documents briefly explained below. They define a generic architecture for designing a Message Transfer System.

X.400 - Specifies the model including the User Agents, Message Transfer Agents, Message Transfer and InterPersonal Messaging Services.
X.401 - Describes the basic service elements and optional user facilities of the MTS.
X.408 - Specifies the rules for converting one information type to another.
X.409 - Defines a structured syntax description language and an encoding scheme for transmitting messages. This syntax is also known as ASN.1 (Abstract Syntax Notation number One) in the OSI world.
X.410 - Specifies the facilities and protocols used by the Remote Operations Server (ROS) and Reliable Transfer Server (RTS).
X.411 - Specifies the services and protocol for the Message Transfer Sublayer (MTL).
X.420 - Specifies the content protocol for InterPersonal Messaging Service.
X.430 - Specifies the means of access for teletext terminal.

2.3 1988 X.400 recommendations

2.3.1 Introduction

The 1984 X.400 recommendations contain a number of errors and ambiguities (several areas like security, directory functions were left for further studies). Some extra facilities have been identified (Message Store, connection to physical delivery services, etc...). When 1984 X.400 was satisfied as a recommendation, the OSI architecture for the application layer was not completed. It has since been finished.

All these points have led to a new set of recommendations, called the 1988 X.400 recommendations. The 1988 version should be free of errors and ambiguities, encompass the new OSI application layer, be compatible with existing 1984 X.400 implementations and include new features and facilities. After a quick description of the OSI layer seven, the 1988 X.400 recommendations main features are explained.

2.3.2 Application layer revisited [SIMO 89]

This is a description of the main points of the new OSI application layer. In a model with distributed applications, tasks are called Application Processes (APs). The communication aspects of each AP are found in the Application Entities (AEs) which are located in the application layer of the OSI model. See figure 2.3.1.

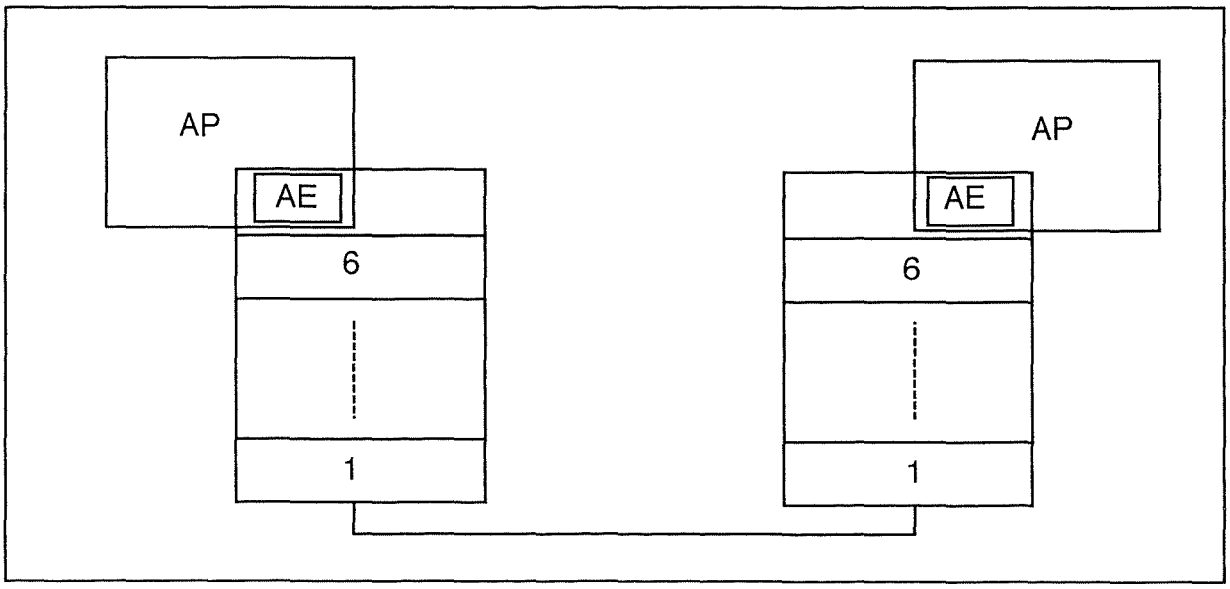

Fig 2.3.1 : Application Process on OSI stack

Each application entity is characterized by its type and is made of one or more Application Service Elements (ASEs). An AE has its own access point to the presentation layer called P-SAP (Presentation Service Access Point). See figure 2.3.2.

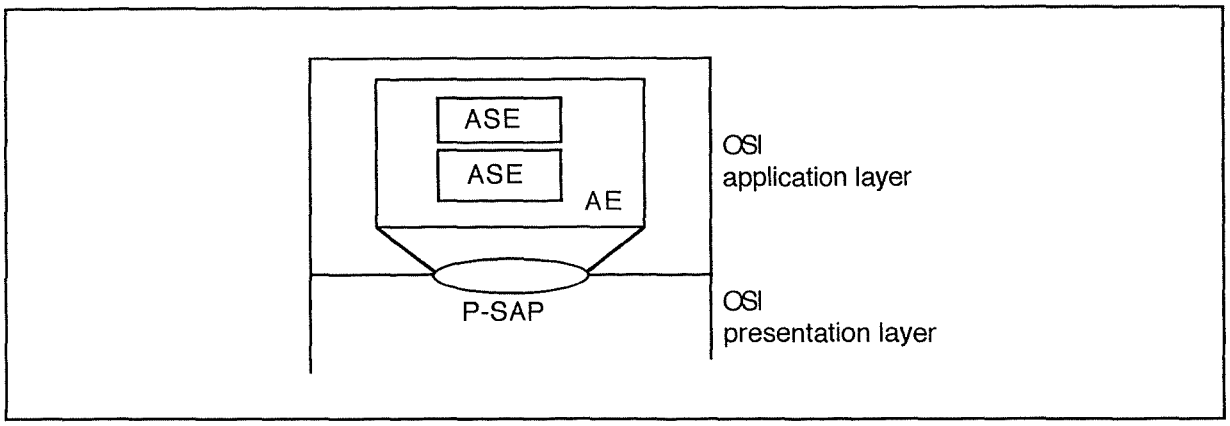

Fig 2.3.2 : ASE and P-SAP

The Application Service Elements (ASEs) cooperate in order to provide the functionality of an AE. Each ASE provides a specific OSI communication capability. The cooperation between two AEs of the same type defines an application protocol. An application protocol includes a list of the ASE types involved and the way these ASEs are used.

A typical application entity structure contains two main parts : specific ASEs (SASEs) and common ASEs (CASEs). The specific ASEs provide services for a particular application and the common purpose ASEs are used in various applications. See figure 2.3.3.

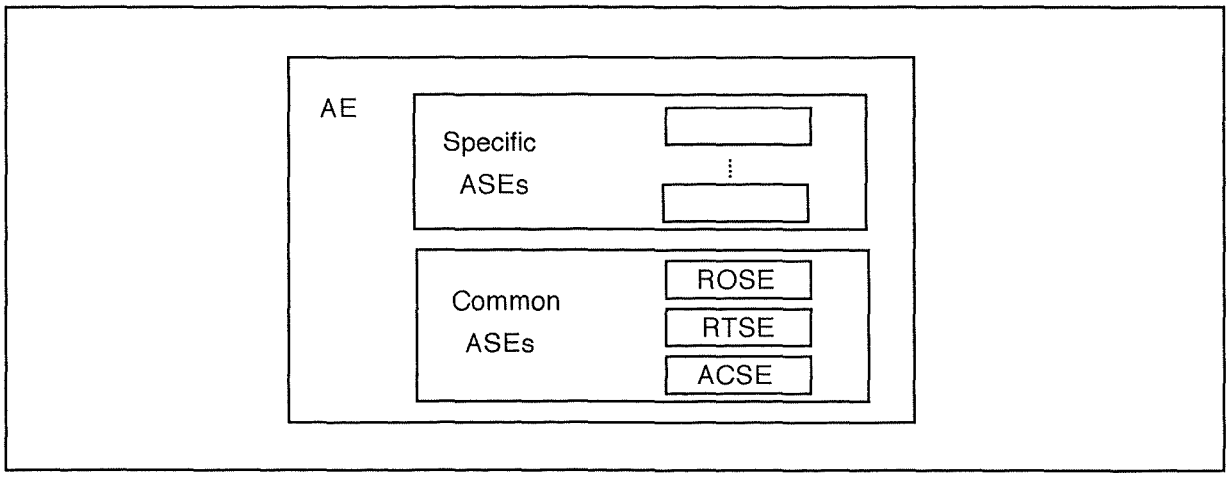

Fig 2.3.3 : AE content

The common ASEs are ROSE (Reliable Operation Service Element), RTSE (Reliable Transfer Service Element) and ACSE (Association Control Service Element). The ROSE and the RTSE are both evolutions of the ROS and RTS defined in 1984 X.410 recommendation. The ACSE establishes, manages and releases application associations between AEs of the same type. All the Application Entities contain the ACSE.

This new structure means for 1988 X.400 recommendations that the sublayers MTL and UAL will disappear. The introduction of AEs has to be accomplished in order to provide the same functions as 1984 X.400 plus additional ones.

2.3.3 New Description Techniques [NUSS 91 III]

OSI has adopted a standardized technique for the application definitions. A formalized syntax notation (ASN.1) and the object-oriented approach are now used.

The CCITT has defined an Abstract Service Model allowing the description and the specification of tasks in the OSI layer seven. We will only give a simplified description of this model. The reader can find more in [X.400 88].

The Abstract Service Model is based on the relation between Abstract Objects. Each Abstract Object uses or provides an Abstract Service. In order to provide an Abstract Service, Abstract Objects execute Abstract Operations. The Abstract Objects communicate with each other through Abstract Ports. An Abstract Bind establishes an association between two Abstract Objects and an abstract Unbind operation releases the association.

The Abstract Service Model makes a distinction between the Abstract Model and the Abstract Service. The Abstract Model describes the Abstract Objects and Abstract Ports. The Abstract Service contains the Abstract Bind, Abstract Unbind and Abstract Operations.

All these concepts will be used in this part.

2.3.4 1988 MHS Abstract Model

The 1988 X.400 functional model is illustrated in figure 2.3.4 . The Abstract Ports are not shown in figure 2.3.4, their use is explained in part 2.3.8.

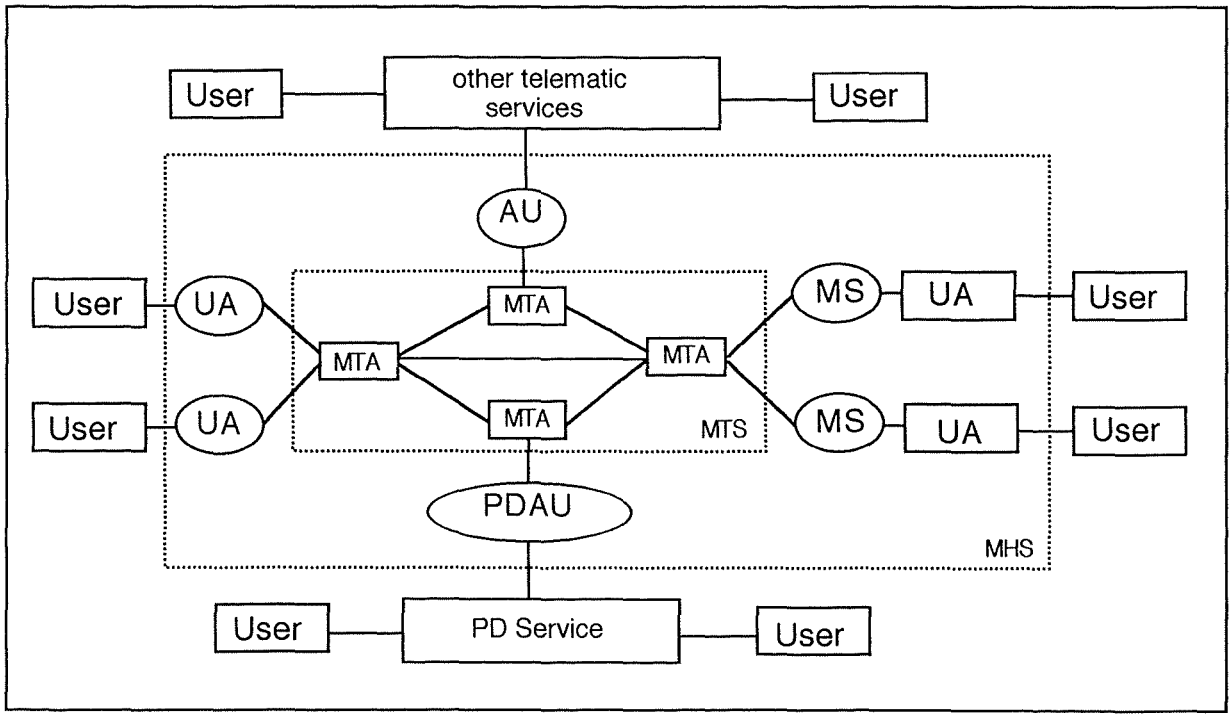

Fig 2.3.4 : 1988 MHS Abstract Model

The 1988 X.400 functional model has three additional features than the previous one :

- The Abstract Object Message Store (MS) is optional. It can be included between the MTS and a UA.
- The Abstract Object Access Unit (AU) is introduced to allow other telematic services. Conversion of the coded information is also defined.
- The Abstract Object Physical Delivery Access Unit (PDAU) provides a special access for Physical Delivery Service . It allows users to send a message to the Physical Delivery Service which produces a hard copy (paper, disk, tape, ..) and the envelope for sending the message physically.

2.3.5 Message Store

The message store is probably the most important improvement of the 1988 X.400 recommendations in comparison with the 1984 version. The message store is located between a MTA and its remote UA. See figure 2.3.5.

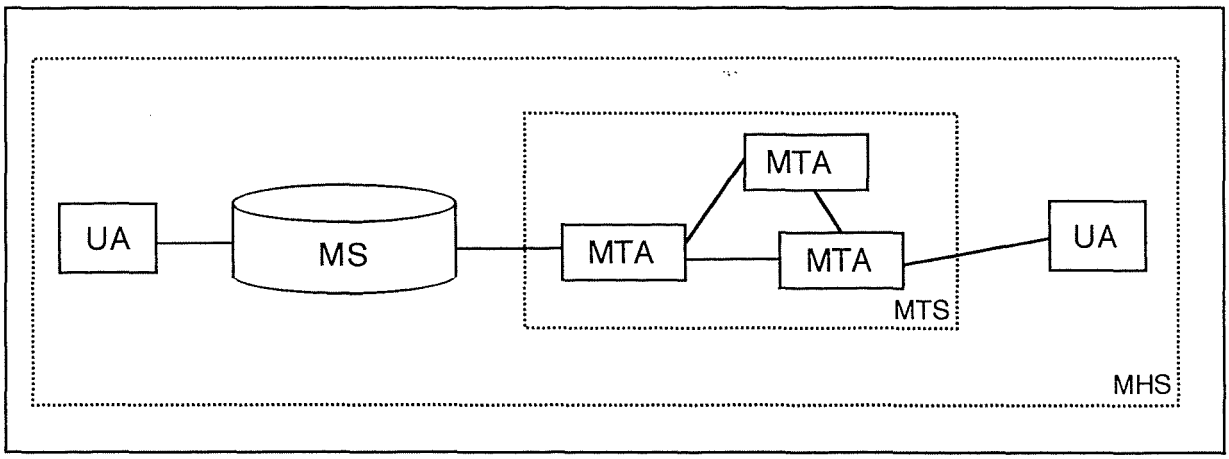

Fig 2.3.5 : MS in MHS environment

The inconvenience of the 1984 X.400 recommendations, is that when a remote UA is turned off for a period of time, the first time it is switched on again, a large number of messages can be sent from the MTA to the UA. But if the amount of messages is too large for the available information storage then message losses can occur.

The 1988 X.400 Message Store (MS) allows the remote UA to be turned off without the danger of message loss when it is switched on again.

The Message Store can be described as follows :

- a Message Store (MS) is for one user (for only one O/R Address).
- the message delivery is done through the Message Store only for UAs subscribing to a MS.
- when there is a MS between the remote UA and the MTA, all the messages for that UA are delivered to the MS. When the message is delivered to the MS, the MTS role for the transfer is completed.

- the message submission from UA to the MTA can be done directly or indirectly, via the Message Store.
- the users can list the delivered messages, they can also select the retrieval of messages or delete them.
- the user can request an alert when a particular message arrives.

See figure 2.3.6.

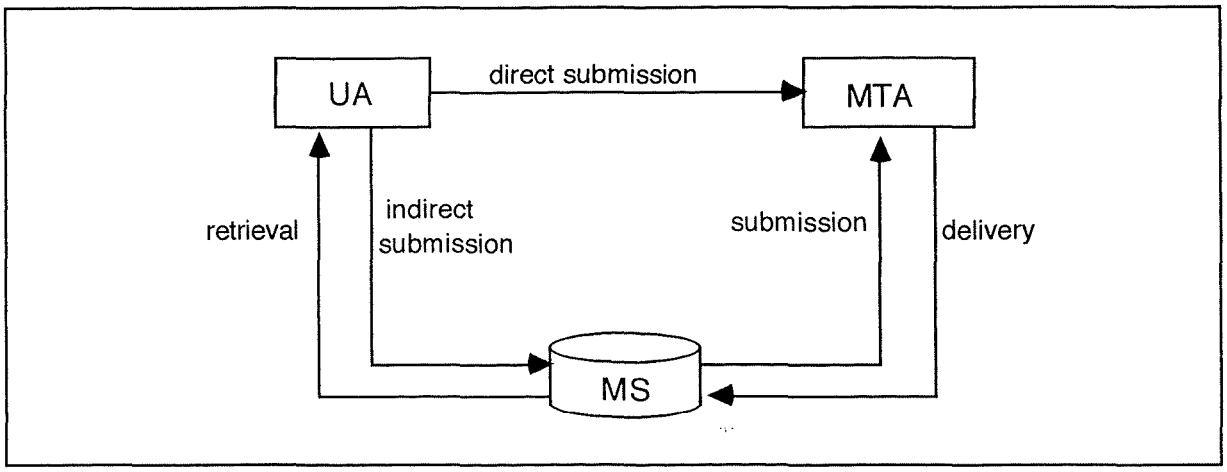

Fig 2.3.6 : MS Model

2.3.6 Security Enhancements [NUSS 91 IV]

The requirement for the security features is due to the distributed nature of the Message Handling System. This system requires protection against various threats. There are four categories of threats : access threats, inter-message threats, intra-message threats and data-store threats.

The first category of threats is concerned with the access to the system by invalid users.

The second category is concerned with inter-message threats executed by non authorized users. These can be masquerade, message modification, replay of the same message and traffic analysis. Masquerade allows the non authorized users to get secret information on a user.

The third category is concerned with intra-message threats executed by authorized users which act dishonestly. These can be repudiation or security level violation. The repudiation is the denial by the receiver or the originator of respectively the receipt or the sending of a message. The security level violation is the sending and the receipt of message by users that are not at the right confidential level in an organization hierarchy.

The last category is concerned with data store threats. These can be modification of routing information or preplay. The modification of routing information can allow an ill-intentioned user to get a copy of particular messages. Preplay allows a dishonest user to get a copy of a message which is held for deferred delivery.

An extract of 1988 X.400 recommendations provides a list of all the security capabilities used to counter the threats.

Message Origin Authentication. Enables the recipient, or any MTA through which the message passes, to authenticate the identity of the originator of a message.

Report Origin Authentication. Allows the originator to authenticate the origin of a delivery/non delivery report.

Probe of Delivery. Enables any MTA through which the probe passes to authenticate the origin of the probe.

Proof of Delivery. Enables the originator of a message to authenticate the delivered message and its content, and the identity of the recipient(s).

Proof of Submission. Enables the originator of a message to authenticate that the message was submitted to the MTS for delivery to the originally specified recipient(s).

Secure Access Management. Provides authentication between adjacent components, and setting up of the security context.

Content Integrity. Enables the recipient to verify that the original content of a message has not been modified.

Content Confidentiality. Prevents the unauthorized disclosure of the content of a message to a party other than the intented recipient.

Message Flow Confidentiality. Allows the originator of a message to cancel the message flow through MHS.

Message Sequence Integrity. Allows the originator to provide to a recipient proof that the sequence of message has been preserved.

Non Repudiation of Origin. Provides the recipient(s) of a message with proof of origin of the message and its content which will protect against any attempt by the originator to falsely deny sending the message or its content.

Non Repudiation of Delivery. Provides the originator of a message with proof of delivery of the message which will protect against any attempt by the recipient(s) to falsely deny receiving the message or its content.

Non Repudiation of Submission. Provides the originator of a message with proof of submission of the message, which will protect against any attempt by the MTS to falsely deny that the message was submitted for delivery to the originally specified recipient(s).

Message Security Labelling. Provides a capability to categorise a message, indicating its sensitivity, which determines the handling of a message in line with the security policy in force.

2.3.7 Directory Access

It is very important that the access to a Directory Service is provided to the MHS environment. A Directory Service is the means by which users can find addressing information and also by which the Message Transfer Service can establish a route for a message in the MHS.

The CCITT Directory Service is called X.500 recommendation. The service is so crucial for the Open System Interconnection world, that CCITT and ISO have co-ordinated their works (ISO

9594/1-8). After a description, explanations about how the MHS can use the Directory Service are given. The X.500 recommendation has been adopted and is fully integrated in the 1988 X.400 recommendations.

The X.500 recommendation for Directory Service describes both the information content and the access means to it. The Directory Environment is made of Directory User Agents (DUAs) and of one Directory System containing Directory Service Agents (DSAs). The Directory User Agents are the users of the Directory System. See figure 2.3.7.

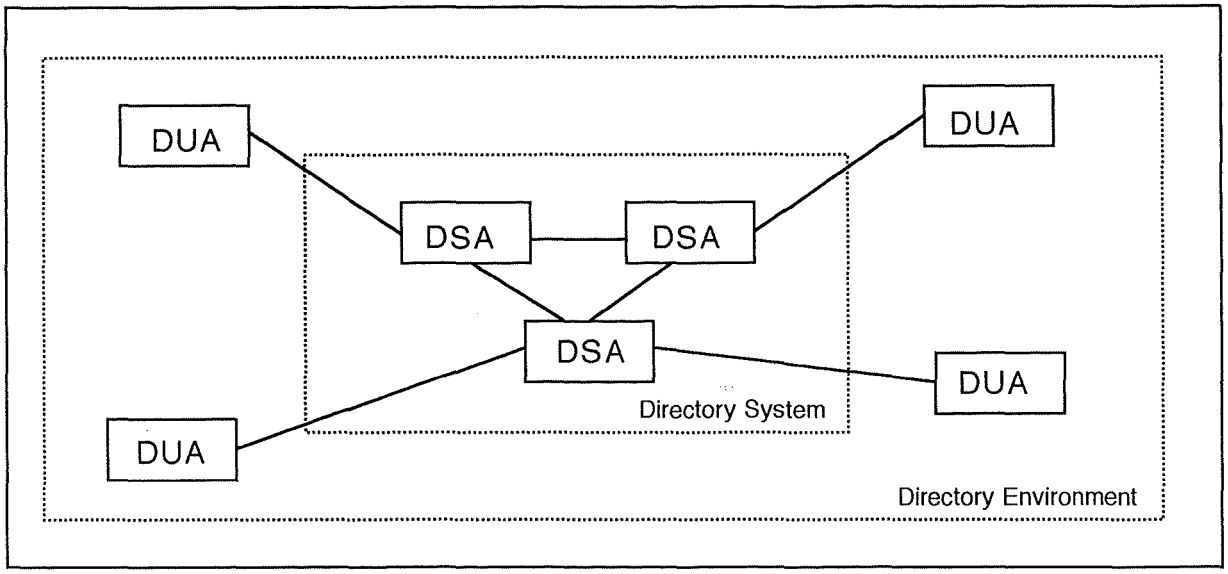

Fig 2.3.7 : Directory Environment

The Directory System is physically and administratively distributed between the Directory Service Agents. Each DSA both public and private possesses a fragment of information concerning one or more naming contexts and is responsible for maintaining its own information. The DSA provides to the DUA an access to the information.

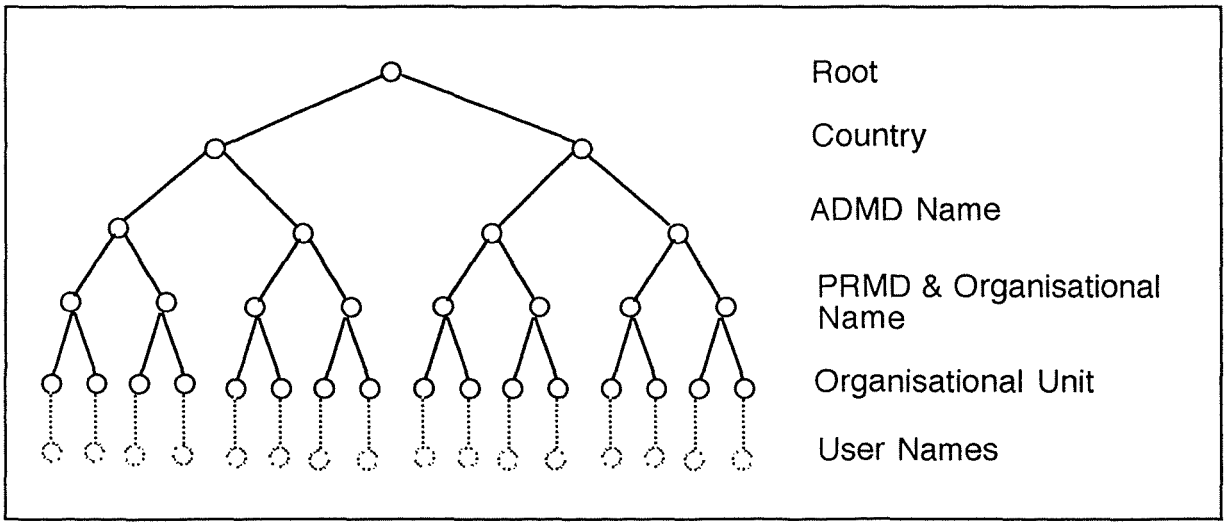

Fig 2.3.8 : Directory Information Tree

The Directory Information Base (DIB) is the reunion of all the information held in each DSA. The information in the DIB is organized according to the concept of hierarchical tree structure which is called the Directory Information Tree. See figure 2.3.8. An example of its use is given in part 4.4.2.

The DSAs communicate with each other via a protocol called Directory System Protocol (DSP). The DUAs and the DSA communicate via the Directory Access Protocol (DAP). Both protocols are based on the 1988 X.400 Remote Operation Service Element (ROSE). These protocols are illustrated in figure 2.3.9.

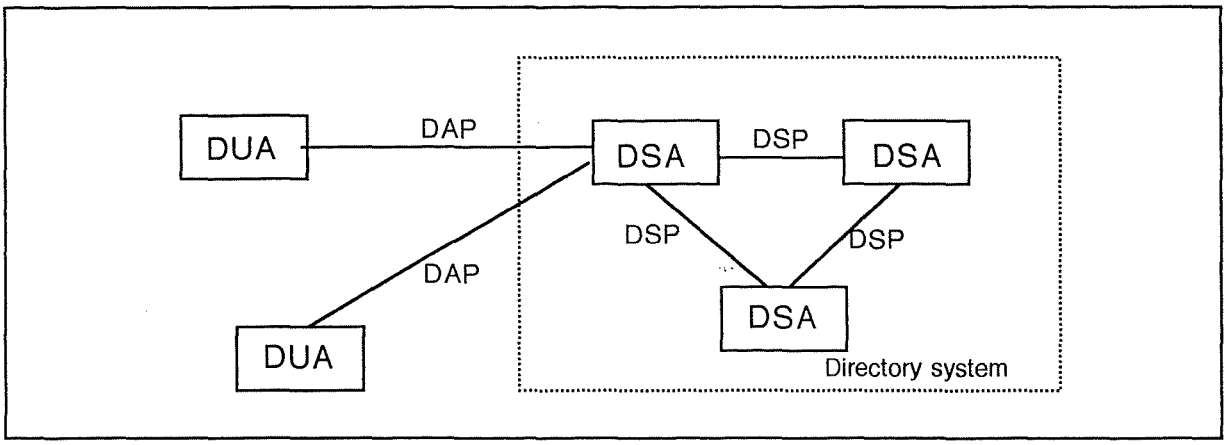

Fig 2.3.9: Directory protocols

The access to the directory is possible for Users, UAs and MTAs. See figure 2.3.10.

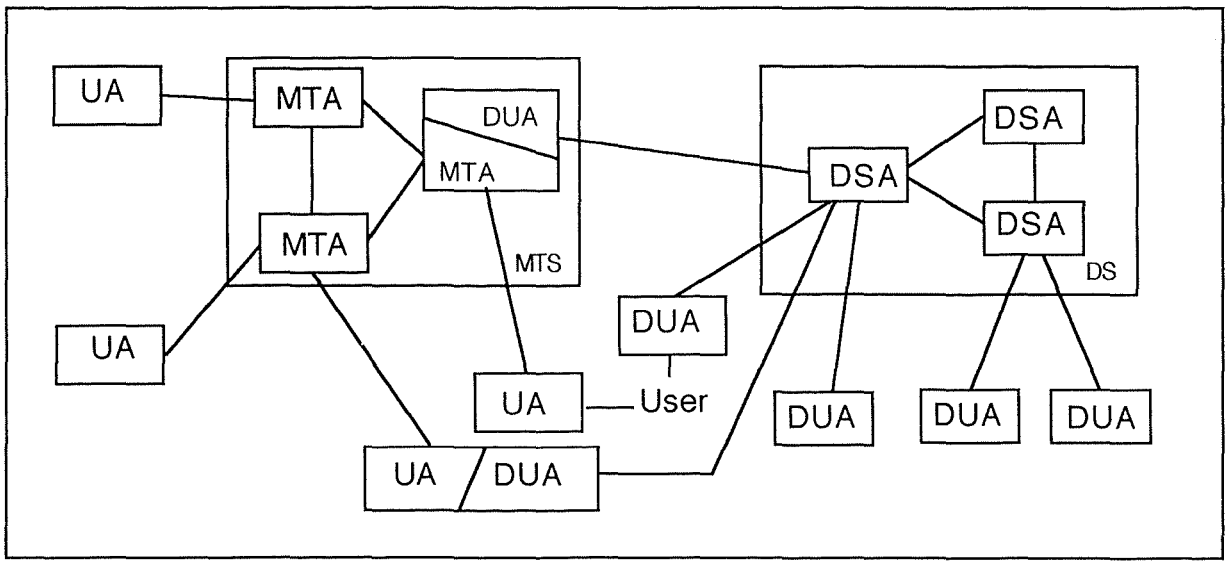

Fig 2.3.10 : MHS and Directory Services link

The UA and the MTA can have a DUA built into them. A User can have access to the Directory Service through a stand alone DUA, through a UA connected to a MTA using a built-in DUA or through a UA using a built-in DUA. For the user point of view, this solution provides a user-friendly naming system and a means for the MTS to establish its route across the MHS when a message is submitted.

2.3.8 MHS Abstract Service

The Abstract Operations are executed in order to provide the Abstract Service. The 1984 X.400 service elements are now called elements of service in the 1988 X.400 recommendations. The elements of services are more detailed definitions of the Abstract Service. The reader can find them in [X.400 88].

MT Abstract Service

The MTS provides MT Abstract Service to the MTS-Users (UA and MS) through three Abstract Ports (Submission, Delivery and Administration Port) and Abstract Service between MTAs through one Abstract Port (Transfer Port). See figure 2.3.11.

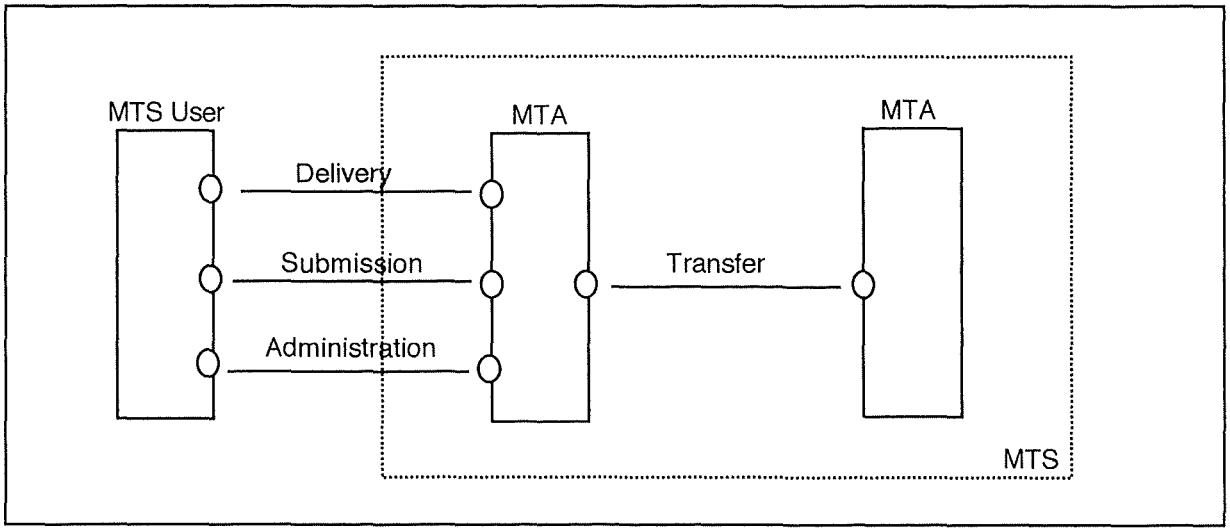

Fig 2.3.11 : MTS Abstract Model

The MT Abstract Service to MTS-Users is :

Bind/Unbind :
- MTS-bind
- MTS-unbind

Submission Port Abstract Operations :
- Message Submission
- Probe Submission
- Cancel-deferred-delivery
- Submission control

Delivery Port Abstract Operations :
- Message delivery
- Report delivery
- Delivery control

Administration Port Abstract Operations :
- Register
- Change credentials

The MT 'internal' Abstract Service is :

Bind/Unbind
- MTA-bind
- MTA-unbind

Transfer Port Abstract Operations
- Message Transfer
- Probe Transfer
- Report Transfer

MS Abstract Service

A MS provides MS Abstract Service to a UA through three Abstract Ports (Retrieval, Indirect Submission, Administration Port) and uses the Abstract Service provided by the MTS through the Abstract Ports defined in the previous part. See figure 2.3.12

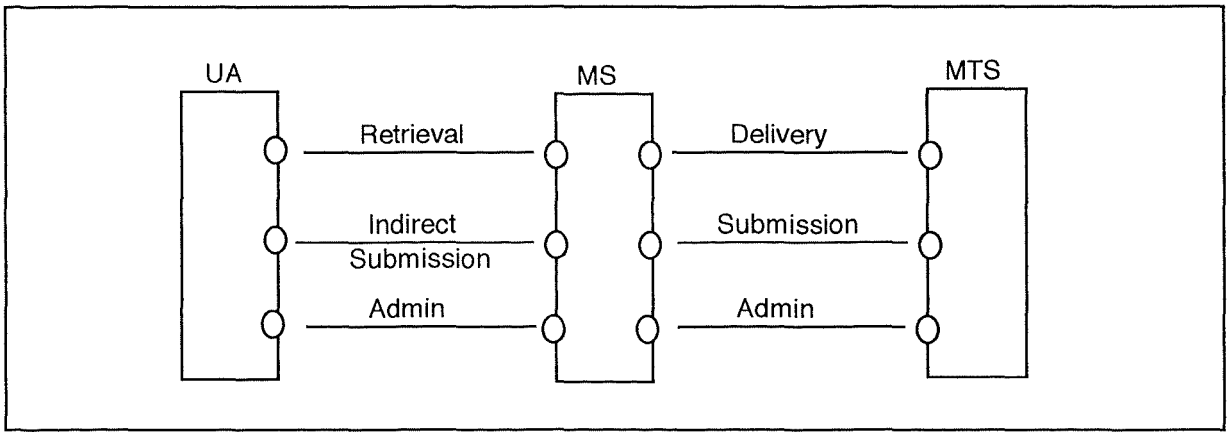

Fig 2.3.12 : MS Abstract Model

The MS Abstract Service to UA is :

Bind/Unbind :
- MS-bind
- MS-unbind

Retrieval Port Abstract Operations :
- Summarize
- List
- Fetch
- Delete
- Register-MS
- Alert

Indirect submission Port Abstract Operation :
- Indirect Submission

Administration Port Abstract Operation :
- Administration

IPM-UA Abstract Service

An IPM-UA provides IPM-UA Abstract Service to an IPM-User through three Abstract Ports (Origination, Reception, Management Port) and uses Abstract Service provided by the MS and MTS. See figure 2.3.13.

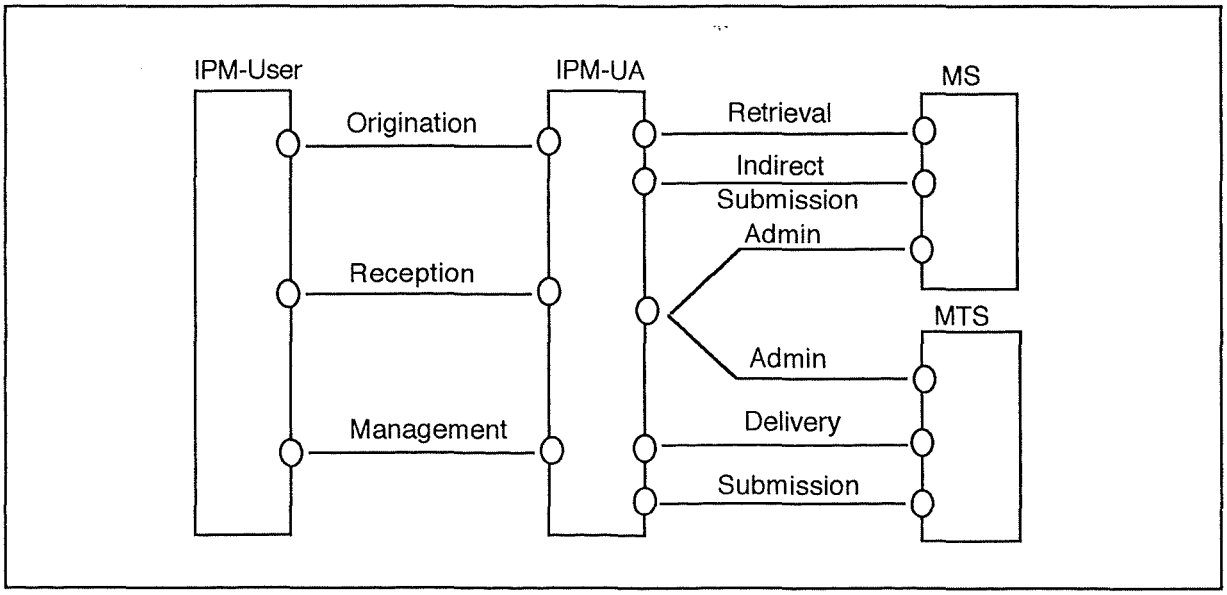

Fig 2.3.13 : IPM-UA Abstract Model

The IPMS Abstract Service to IPM-User is :

Origination Port Abstract Operations :
- Originate Probe
- Originate IPM
- Originate Receipt Notification

Reception Port Abstract Operations :
- Receive Report
- Receive IPM
- Receive Receipt Notification

- Receive Non.Receipt Notification

Management Port Abstract Operations :

- Change auto-discard
- Change auto-acknowledgement
- Change auto-forwarding

2.3.9 MHS Protocols

See protocols in figure 2.3.14.

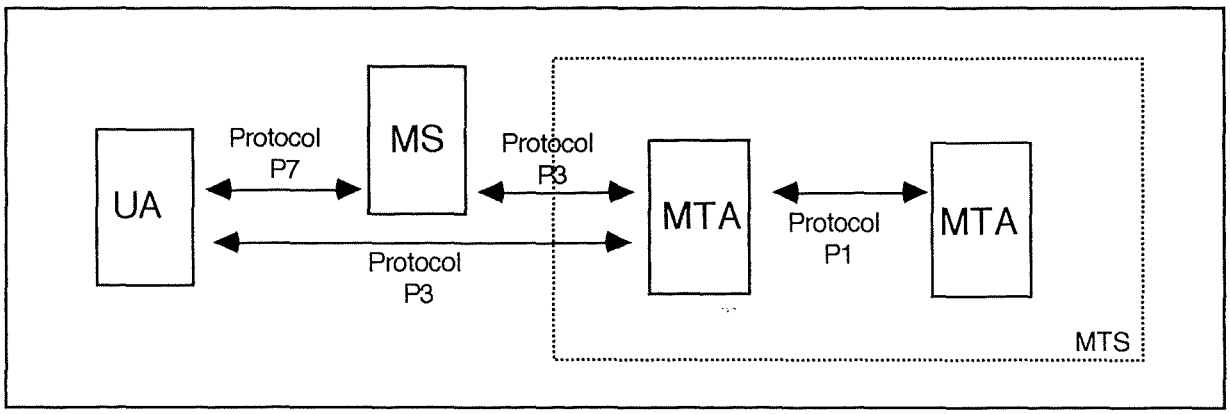

Figure 2.3.14 : 1988 X.400 protocols

MTS Transfer protocol P1

The MTS Transfer protocol P1 is supported by a specific ASE, called Message Transfer Service Element (MTSE), supporting the Transfer Abstract Port between MTAs.

MTS Access protocol P3

The MTS Access protocol P3 is supported by three ASEs, each supporting a type of Abstract Port between the MTS and the MTS Users. They are :

- Message Submission Service Element (MSSE) supports the Submission Port.
- Message Delivery Service Element (MDSE) supports the Delivery Port.
- Message Administration Service Element (MASE) supports the Administration Port.

MS Access protocol P7

The MS Access protocol P7 is supported by three ASEs, each supporting a type of Abstract Port between the MS and the UA. They are :

- Message Retrieval Service Element (MRSE) supports the Retrieval Port.
- Message Submission Service Element (MSSE) supports the Indirect Submission Port.
- Message Administration Service Element (MASE) supports the Administration Port.

IPMS content protocol P2 [MANR 89]

P2 was, in 1984 X.400 recommendations, the content protocol of the UA Sublayer between the UA Entities defining the user-ip-message and the report-ip-message. The term 'P2 protocol' in the 1988 X.400 recommendations is kept for sentimental reasons. Indeed with the abolition of sublayers, as described in part 2.3.2, P2 is a format definition of IP-messages rather than a protocol. The IP-message header now contains additional information relating to the introduction of the Message Store.

2.3.10 Structure of the 1988 X.400 recommendations

The CCITT X.400 set of recommendations is comprised of nine documents briefly explained below. They define a generic architecture for designing a Message Transfer System.

X.400 - Introduces the general concepts of MHS including new features.
X.402 - Specifies in detail the new concepts and how the MHS Model relates with the new application layer architecture.
X.403 - Describes the test methods, criteria used for conformance testing with the 1984 MHS.
X.407 - Specifies the services to be provided in a MHS environment in an abstract form.
X.408 - Specifies as in the 1984 version, the rules for conversion type to another but all the points let for further studies have been completed.
X.411 - Defines the Abstract Service provided by the MTS and specifies procedures to be performed by the MTAs.
X.413 - Defines the procedures to be performed for using the Message Store.
X.419 - Specifies the protocols (P1, P3 and P7).
X.420 - Specifies how users can exchange messages, through the InterPersonal Messaging System, between humans.

Chapter 3 : The use of X.400 for EDI

3.1 Introduction

EDI means replacing the traditional exchange of paper-business documents like orders, invoices,.... with computerized messages reaching the same goal. EDI is concerned with the exchange of structured data between computer applications.

The choice of a communication medium is primordial. One way to find out which communication system best fits EDI is to compare the requirements and capabilities.

The EDI communities require several services in order to use EDI for business effectively. The first part of this chapter contains a full list of user's requirements. In order to get a good feeling for the capabilities of X.400, for each requirement, the capability which could be provided by the 1988 X.400 Messaging System is explained.

The second part deals with possible implementations of EDI application using X.400. Two scenarios are explained in that part. They are both using the X.400 existing implementations. A dedicated recommendation especially designed for EDI is now existing. It is often called P_{edi}, even though it is now named X.435 / F.435. Due to its importance this scenario is explained in the next chapter "X.435 / F.435 recommendation"

3.2 Comparison of X.400 Capabilities against EDI Requirements

A list of EDI requirements for the EDI Messaging System is proposed below. This list of requirements is extracted from the Vanguard publication on EDI and X.400 [VANG 88]. The requirements are based on a series of interviews of representative users, network providers and standards committees. Vanguard has classified the requirements into seven classes : Message Transfer, Message Store and Selective Retrieval, Interworking, Security and Audit, Acknowledgement, Administration and Others.

The list contains requirements wanted by all the EDI community but some of them are not considered with the same importance by various people.

For each requirement, the capabilities of the 1988 X.400 are described. The sentences enclosed between quotation marks are extracted from [VANG 88].

3.2.1 Message Transfer

(1) **"The reliable transfer of a set of EDI messages (an EDI Interchange) between two computer systems" .**

"X.400 operates over a reliable transfer service".

(2) **"The ability to transfer an EDI Interchange as a whole, encoded in a syntax (UN/TDI, ANSI X.12, UN/EDIFACT) not recognised by X.400 systems (not coded in ASN.1)" .**

Encoding syntaxes for EDI like UN/TDI, ANSI X.12, UN/EDIFACT are based on character strings with delimiters between fields. The ability to carry existing character-based syntaxes transparently is provided.

(3) **"The ability to hold EDI Interchanges in transfer for forwarding to the designated recipient when it is ready" .**

The 1988 X.400 recommendations define a store-and-forward service.

3.2.2 Message Store and Selective Retrieval

(4) **"The ability to select specific EDI Interchanges for receipt based on attributes of the EDI Interchange" .**

The 1988 X.400 Message Store provides the ability to select specific messages. This selection is provided through the use of information contained in the heading of X.400

messages. This information is encoded in ASN.1. ASN.1 Basic Encoding Rules (BER) are based on the use of strings of octets comprising type, length and value fields. EDI encoding syntaxes are based on character strings with delimiters between fields. So it is impossible to use easily the information contained in EDI Interchanges. "Two approaches could be taken to resolve this incompatibility, either :

(a) EDI header information is repeated in an X.400 end to end protocol (Pc protocol) header.

(b) The EDI Interchange header, and possibly the body, is recoded using ASN.1"

Both approaches should be "supported within X.400, allowing the user to select the approach which best suits his requirements". The X.400 header can be mapped or copied from any of the three main EDI syntaxes (UN/TDI, ANSI X.12, UN/EDIFACT).

(5) "The ability to select specific EDI Functional Groups (a group of invoices) within specific EDI Interchanges for receipt based on attributes of the EDI Interchange and the EDI Functional Group" .

"The X.400 Message Store already supports retrieval of InterPersonal Messaging body parts from X.400 Messages. Similar functions can be used to retrieve EDI Functional Groups". Providing the selection of Functional Groups within an EDI Interchange requires that the EDI structure be coded in ASN.1. This recoding of useful attributes in ASN.1 is possible for the three main EDI syntaxes.

(6) "The ability to list general information about EDI Interchanges awaiting retrieval from storage within the communication system (list size and sender of all interchanges containing invoices)" .

"The X.400 Message Store provides this facility. Again this requires the attributes to be listed to be recoded or repeated in ASN.1".

3.2.3 Interworking

(7) "The ability to interwork with existing Value Added Network for EDI" .

"X.400 supports interworking between Management Domains (separately administered networks)". A VAN can be connected to X.400 Message Transfer Service, thus it is reasonable to think that a relay between a non-X.400 network with a X.400 network is possible.

(8) "The ability to interwork internationally between different network providers" .

X.400 services provide international connectivity between a large amount of Public Administration and Private Management Domains.

(9) "The ability to interwork with a private message handling network, run by an organization for its own use" .

"X.400 supports interworking between Private Management Domains and Public Administration Management Domains".

(10) "The ability to interwork, possibly via an application gateway, with an InterPersonal Messaging System (IPMS) based on X.400 so that :

(a) EDI messages might be submitted into an InterPersonal Messaging System for onward routing to an EDI Application" .

"This could be achieved through a gateway system which on one side acts as an 'EDI Messaging User Agent' (dedicated User Agent for EDI) and on the other side acts as InterPersonal Messaging User Agent". The gateway plays the role of converting the message in a way that it could be readable by both sides.

(b) "InterPersonal Messages, containing for example an EDI message with additional text for human action, may be submitted by an EDI Application" .

This "could be achieved by support of an InterPersonal Messaging System User Agent in parallel with an EDI Messaging User Agent in the application system".

3.2.4 Security and Audit

(11) "The ability to log all the parts or selected attributes (sender, recipient, control reference and digital signature) of outgoing and incoming EDI Interchanges for audit purposes" .

"This is not currently supported by X.400". This could be achieved by the user application itself using log files.

(12) "The ability to correlate incoming acknowledgements with the outgoing message log" .

"This is not currently supported by X.400". This could be achieved by the user application itself using log files.

(13) "The ability to authenticate the origin and integrity of EDI Interchange to the recipient and to third parties (authenticity that can not easily be repudiated later by originator)" .

The 1988 X.400 Recommendations include a 'Message Origin Authentication' Security Service. This element of service enables the recipient, or any MTA through which the message passes, to authenticate the identity of the originator of the message.

The 1988 recommendations also contain a 'Content Integrity' Security Service which allows the recipient to verify that the original content of the message has not been changed.

These two elements of service could be used to provide the ability to authenticate the origin and integrity of an EDI Interchange.

(14) "The ability to restrict delivery of EDI Interchanges to those containing EDI messages of a specific type from specific originators with whom there is an agreement to interchange data" .

"X.400 supports a 'Restricted Delivery' Service which allows the Message Transfer Service to return messages undelivered because delivery from the sender is not authorised for the given recipient. This element of service, however, does not specify how much such a restriction could be based on attributes (e.g. EDI Message Type) held within a X.400 Message. In addition, there is no provision for remote management, by standard protocols, of an access control list for Restricted Delivery.

Security between X.400 Message Transfer Service and its users is based to a large degree on the use of 'Security Labels'. The form of this label is fairly free, allowing a message to be tagged with any information which categorizes it for security purposes. X.400 currently uses 'Security Labels' to control the flow of messages to and from users to those it is authorised to handle. However, currently the 'Restricted Delivery' Service does not make use of 'Security Labels'. No standard element of service is defined whereby a remote Originator can be assigned a Security Context (a set of 'Security Labels') by Recipients to restrict the class of messages to be received on a per Originator basis. For further explanation of Security Context the reader must refer to the 1988 X.411 recommendation.

For EDI a field of the 'Security Label' could be used to classify the type of EDI messages being carried in a X.400 Message. In addition, The 'Restricted Delivery' Service could be extended to allow a Security Context (a set of Security Labels) to be assigned to the list of remote originators from whom a user is prepared to receive messages. Thus, if 'Restricted Delivery' is supported, before delivering a message, the Message Transfer Service not only checks that the Originator is authorised to send a message to the Recipient, but it also checks that the Security Label (which contains information on the type of EDI message being carried) is part of the authorised Security Context assigned to the Originator for the Recipient.

Finally, it is proposed that X.400 is extended to enable the list of authorised Originators and Security Context to be up-dated by the user or his authorised administrator".

(15) "The ability to restrict flow of EDI Interchanges between interconnected sub-networks to those of a given type between specific Sender and Recipient pairs" .

"In X.400, control of flow between X.400 Management Domains, which are considered to be equivalent to sub-networks, is based on the use of a 'Security Label'. As described above, a 'Security Label' could be used to classify the type of EDI messages held within a X.400 Message.

It is considered neither feasible, nor necessary, to control flow between sub-networks on an individual Sender/Recipient basis. Users who could be classified in user groups may be added to the Security Label of a X.400 Message. Before passing a message between domains the sending domain checks whether the recipient domain has been cleared to handle messages for the user group held in the Message Security Label".

(16) "The ability to separate flow of EDI messages and InterPersonal Messages" .

"If X.400 Messages relating to EDI and InterPersonal Messages are separate content types then separation of flow can be provided by X.400".

(17) "The ability to handle EDI messages without the need to interpret the message content" .

There is no requirement of interpretation of the message content for any elements of service of the Messaging system except for conversion of encoding syntax. This conversion can be prohibited in the X.400 P1 envelope.

(18) "The ability to cipher EDI Interchanges" .

"The X.400 'Content Confidentiality' Security Service could support encryption of the whole of an interchange, including any header information. To allow selective retrieval of the message from a Message Store some of the Interchange header needs to be in plain text. Thus an additional facility is required to support encryption of the Interchange body and leaving the header in clear".

3.2.5 Acknowledgement

(19) "The ability to notify the EDI Sender of successful transfer of an EDI Interchange to the recipient's store awaiting retrieval".

"The X.400 Message Transfer supports notification back to the originator of delivery of a message to a Message Store using the Delivery Report Service".

(20) "The ability to notify the EDI Sender of failure to transfer an EDI Interchange" .

"The X.400 Message Transfer Service supports notification of non-delivery through the Non-Delivery Report Service".

(21) "The ability to notify the EDI Sender of retrieval of an EDI Interchange/Functional Group by the recipient" .

As for InterPersonal Messaging X.400, a Receipt Notification facility can be included in the EDI Messaging System. But in case of separate retrieval of Functional Groups, the notification should be sent after all of them have been retrieved.

(22) "The ability to notify the EDI Sender of failure of the Recipient to retrieve an EDI Interchange or Functional Group".

"Non-receipt Notification is supported by X.400 InterPersonal Messaging. A similar facility should be supported by an EDI Messaging System".

(23) "The ability to notify the EDI Sender of acceptance for processing of all/some messages in an EDI Interchange" .

"Acceptance of EDI messages is considered to be a function of the application processing the messages, outside the concern of EDI Messaging Systems".

(24) "The ability to notify the EDI Sender of non-acceptance of specific messages within an EDI Interchange" .

"Non-acceptance of EDI messages is considered to be a function of the application processing the messages, outside the concern of EDI Messaging Systems".

3.2.6 Administration

(25) "The ability for the user or a third party cleared to administer the user (a user group administrator) to up-date information used to manage the provision of messaging to the user (access control list, addressing information) independent of the provider of message store/transfer facilities" .

"X.400 provides facilities for a user to administer his own X.400 facilities. However, no provision is made for an authorised third party to use the management services on the user's behalf. Extensions to both the Message Store and the Message Transfer Service are required to support administration through an authorised third party other than the network provider and originator".

3.2.7 Others

(26) "The ability to specify the urgency of delivery of a specific EDI Interchange" .

There are three levels of priority supported by X.400 : non-urgent, normal and urgent.

"The X.400 Recommendations do not define performance figures for these three levels. CCITT recommends 24 hours, 4 hours and 45 minutes for the three levels of priority. Investigations indicate that many EDI users would expect a faster response. A more acceptable speed of delivery for EDI Messaging might be 1 hour and 5 minutes for normal and urgent messages through a single Management Domain".

(27) "The ability to deliver the same interchange to several recipients" .

The X.400 recommendations support the ability to send copies of the same message to several recipients. In the case of EDI, interchanges are exchanged between two cooperating systems based on bilateral agreements. Thus the problem of this requirement is not part of X.400 capabilities but is in the EDI field.

(28) "The ability to convert between different syntax encodings (as from UN/TDI to UN/EDIFACT)" .

"The X.400 Message Transfer Service provides an Implicit Conversion element of service which allows the encoding of information held in a message to be converted before delivery".

But this requirement counters others. For security reasons, users may use the 'Conversion Prohibition' element of service.

3.3 Possible scenarios

The 1984 and 1988 X.400 recommendations only define one type of UA. This UA is dedicated to InterPersonal Messaging. The services offered by the IPM-UA do not fit perfectly for EDI purpose. CCITT has worked on the design of a new class of UA dedicated to EDI. This work is named X.435 / F.435. The recommendation X.435 / F.435 is explained in the next chapter. But X.435 / F.435 implementations are not yet existing. In order to allow the use of X.400 recommendations for EDI, two interim solutions have been defined (P0/1 solution and P2 solution).

The two next paragraphs explain the two interim solutions. The Appendix 7 contains a comparison between the interim solutions capabilities and the EDI requirements for communication studied in the previous part.

3.3.1 P0/1 Solution [GUIL 87]

Introduction

The P1/0 approach is an interim solution for carrying EDI Interchanges through the Message Handling System. This approach is standardized by the National Institute for Technical Standards and by ANSI X.12. This approach is often called the P1/0 US Solution. This solution can be used either on 1984 or 1988 X.400 recommendations implementations. It is worth noting that no changes are made on the existing EDI standards and to X.400 protocols and services. After a description of the solution, the use of Message Transfer Service is discussed.

Description

The figure 3.3.1 shows how the EDI Interchange fits into the content of a X.400 message.

In order to illustrate the meaning of various data in the figure 3.3.1, two EDI Applications and two OSI stacks with the layer seven protocols and services are shown. In the P1/0 solution, the EDI Interchange is carried as the content, directly in the P1 envelope. Even if the 1988 Message Store is available, selective retrieval of EDI Interchange is impossible. This is due to the fact that the EDI Interchange is encoded in a character based syntax, which the MS cannot interpret.

An EDI Application creates an EDI Interchange and gives it to its UA. The User Agent can be of any class. Thus a UA can be of any type of user or application as desired, as long as it conforms to the requirements for communicating with the Message Transfer Agent for submission and delivery of messages. The EDI Interchange is encapsulated in the content part of the X.400 message. There will be only one EDI Interchange carried in the content. The UA provides the envelope attributes to the MTA which creates the P1 envelope, and submits the X.400 message to its MTA. The reversal operations are made by the recipient.

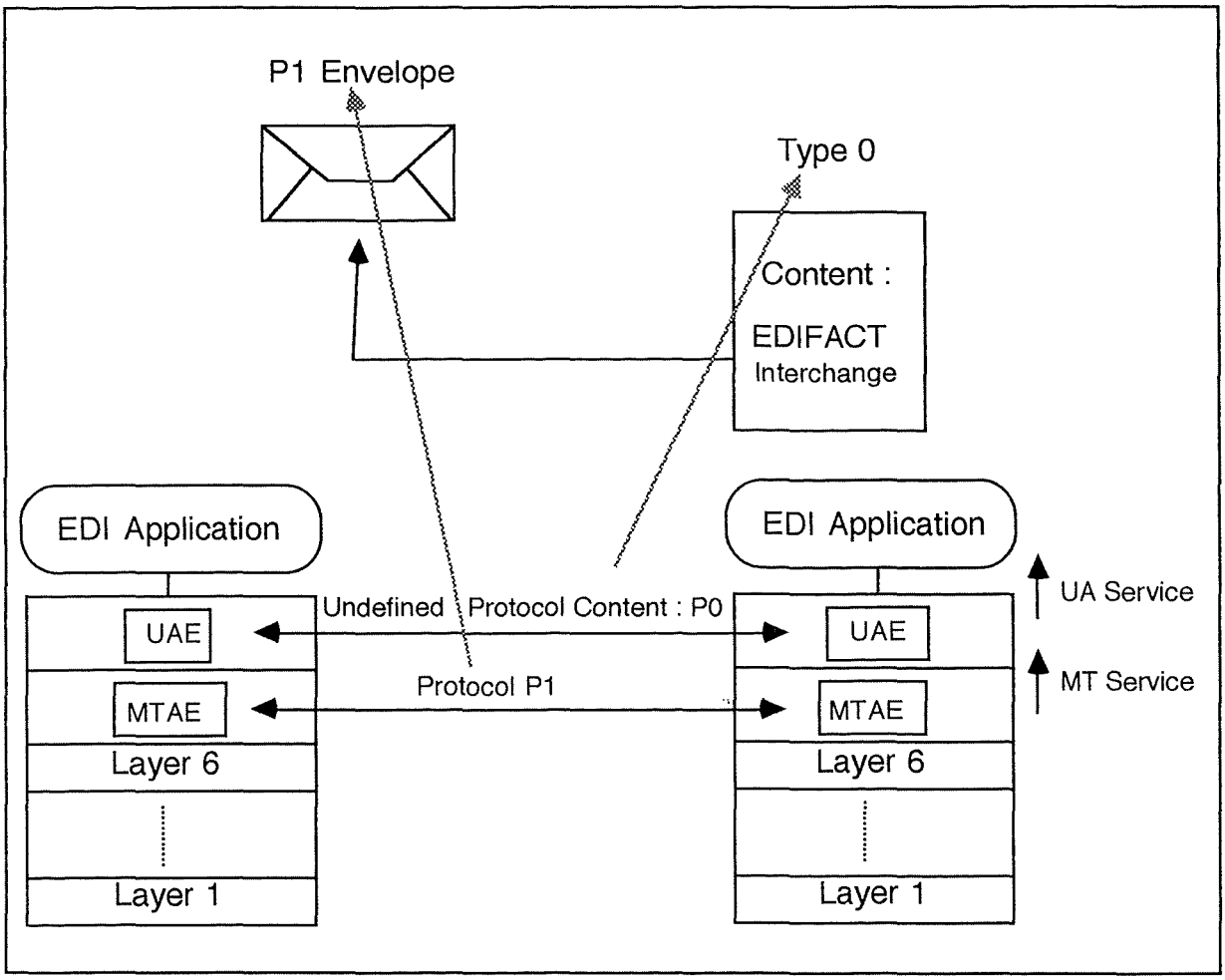

Fig 3.3.1 : P0/1 approach and correspondence with OSI Stack elements

The information about addressing, contained in the EDI Interchange header have to be mapped into P1 service elements.

Since no Directory Service is available for addressing, correspondence between EDI Interchange header fields and P1 fields has to be done in a local directory. The UN/EDIFACT UNB 'Interchange sender' and 'Interchange recipient' segments have to be transformed into X.400 O/R Addresses. The figure 3.3.2 shows an example of addressing correspondence between UN/EDIFACT UNB header fields and P1 addressing fields.

In the figure 3.3.2, the 'Sender Identification' and the 'Identification code Qualifier' are used in order to find, in the local directory, the O/R Address of the sender with all the attributes (Country Name, ADMD, PRMD, Organization Name, Organization Unit Name). The two UN/EDIFACT UNB header fields are fitted into the Domain Defined Attributes. The same operations are executed with the 'Recipient Information Code' and 'Identification Code Qualifier' UNB fields for the recipient.

EDI Interchange Addressing in UNB segments	P1 Addressing
Sender Identification = 56879 Identification code Qualifier = 25	sender O/R Address : COUNTRY = BE ADMD = RTT PRMD = FUNDP ORGANIZATION = INFO ORGANIZATION UNIT = STUDENT DDA = 56879 25
Recipient Information Code = 56780 Identification Code Qualifier = 26	receiver O/R Address : COUNTRY NAME = BE ADMD = RTT PRMD = FUNDP ORGANIZATION = DROIT ORGANIZATION UNIT = CRID DDA = 56780 26

Fig 3.3.2 : EDI addressing mapping into X.400

Use of service elements

Among all the service elements of the Message Transfer Service, the use of several are mandatory and are explained below. The other service elements are not used, unless through bilateral agreement, between EDI Applications.

Original Encoded Information Type Indication :

This service element enables an originating UA to specify to the MTS the encoded type of a message being submitted. When a message is delivered, it also indicates to the recipient UA the encoded type of the message. For P0/1 solution, the value must be IA5Text.

Conversion Prohibition :

This service enables an originating UA to instruct the MTS that the encoded information type conversion(s) should not be performed for a particular submitted message. For P0/1 solution, all users must support and use IA5Text, so this field will not be consulted.

Content Type Indication :

This service element enables an originating UA to indicate the Pc protocol for each submitted message. For P0/1 solution, the type is not defined. Thus the value must be 0.

3.3.2 P2 solution [CEC 88] and [GENI 88]

Introduction

The P2 approach is an interim solution for carrying EDI Interchanges through the Message Handling System. This approach is standardized by the Commission of the European Communities. It is supported by European PTTs. The P2 approach is often called the P2 European Solution. This solution can be used either on 1984 or 1988 X.400 recommendations implementations. It is worth noting that no change is made on existing EDI standards and X.400 protocols and services. After a description of the solution, the use of Message Transfer Service and InterPersonal Service is discussed.

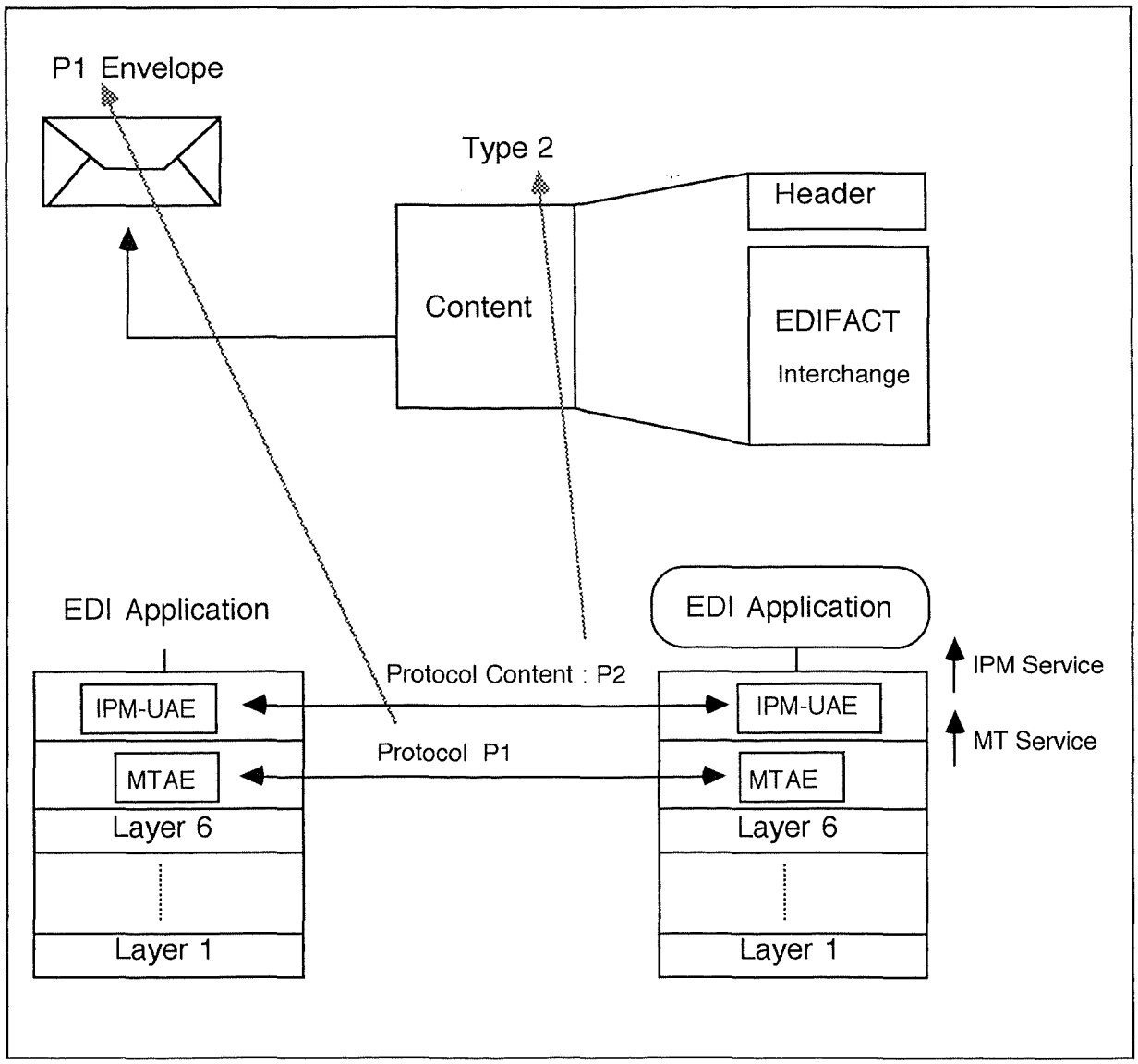

Figure 3.3.3 : P2 approach and correspondence with OSI stack elements

Description

In the P2 solution, the EDI Interchange is carried within a body Part of an InterPersonal Interchange according to the rules of the protocol P2. If the 1988 X.400 Message Store is available, selective retrieval of EDI Interchanges is made possible within this approach. The figure 3.3.3 shows how the EDI Interchange fits into an InterPersonal Message. In order to illustrate the meaning of various data in the figure, two EDI Applications and two OSI stacks with the layer seven protocols and services are shown.

An EDI Application creates an EDI Interchange and gives it to its IPM-UA. The IPM-UA fills the header fields of an IP-message. Specific fields filling are described below. The EDI Interchange is encapsulated in the IP-message body part. The body part type is unstructured IA5Text. Only one body part is allowed and within this body part only one EDI Interchange is carried. The IPM-UA provides the envelope attributes to the MTA which creates the P1 envelope, and submits the IP-message to the MTS. The reversal operations are made by the recipient.

The following information contained in the EDI Interchange header have to be mapped to P2 and P1 service elements.

Addressing

Since no Directory Service is available, correspondence for addressing between EDI Interchange header fields and P1/P2 fields has to be done in a local directory. The UN/EDIFACT UNB 'Interchange sender' and 'Interchange recipient' segments have to be transformed into X.400 O/R Addresses. The example showed in figure 3.3.2 is also valid in this case.

Subject

The UN/EDIFACT UNB 'Application reference' is the header field that specifies the type of message (INVOICE,ORDER,...). It is present if only one type is held within the EDI Interchange. This field has to be mapped into the 'subject' P2 header field. One use of the 'subject' field is for selected retrieval of IP-messages from the Message Store.

Interchange identification

The UN/EDIFACT UNB 'Interchange Control Reference' is the header field that references uniquely the EDI Interchange. The content of this field is assigned by the sender. This field has to be mapped into the 'IP-messageId' P2 header field. This mapping between the EDI Interchange and the IP-message is used for tracking purposes like, for example, correspondence between notifications received and IP-messages sent.

Use of service elements

Among all the service elements of the Message Transfer Service and InterPersonal Service, the use of several are mandatory and are explained below. The other service elements are not used unless through bilateral agreements between EDI Applications.

Message Transfer Service

Registered Encoded Information type :

This service element enables a UA to inform the MTS of the encoded information type(s) that can be delivered to it. For P2 solution, all the users must support IA5Text.

Original Encoded Information Type Indication :

This service element enables an originating UA to specify to the MTS the encoded information type of a message being submitted. When a message is delivered, it also indicates to the recipient UA the encoded information types of the message. For P2 solution, the value must be IA5Text.

Conversion Prohibition :

This service enables an originating UA to instruct the MTS that the encoded information type conversion(s) should not be performed for a particular submitted message. For P2 solution, all users must support and use IA5Text unless they agree on the usage of another encoded information type in bilateral communication agreements.

Content Type Indication :

This service element enables an originating UA to indicate the Pc protocol content for each submitted message. For P2 solution, the value must be P2.

InterPersonal Messaging Service

Typed body :

This service element specifies the nature and attributes of the body of the IP-message to be conveyed within the body. For P2 solution, the value must be unstructured IA5Text.

Chapter 4 : X.435 / F.435 recommendation

4.1 Introduction

The comparison of X.400 capabilities against EDI requirements for communication (chapter 3) indicates that X.400 is an appropriate means for Electronic Data Interchange. Nevertheless 1984 and 1988 X.400 Message Handling System recommendations only contain specifications for an InterPersonal Messaging System (IPMS). IPMS allows its users the exchanging of unstructured messages that are created and read by humans. We know that EDI messages are not created and read by humans but rather by computer applications.

In the previous chapter, we have seen how an EDI message can be sent using the existing X.400 recommendations. But this is not enough to allow the emergence of X.400 use for EDI. X.400 recommendations present lapses. Indeed there are no provisions for selective retrieval from the Message Store on pure EDI attributes, not enough security capabilities and no concept of responsibility.

In order to satisfy these lacks, the CCITT has created a Rapporteur Group for EDI and X.400 in August 1988. This group had to define extensions to X.400 to accommodate EDI user needs. It completed its work in June 1990. The final work consists of the recommendation X.435 / F.435, which is often called Protocol for EDI or P_{edi}.

X.435 / F.435 contains several features that are unique to it, and that are not defined in the other X.400 recommendations. X.435 "defines the message handling application called EDI messaging system, specifying in the process EDI messaging, a form of message handling tailored for exchange of electronic interchange (EDI) information, a new message content type and associated procedures" [X.435 90]. F.435 "defines the overall system and service description of the message handling application called EDI messaging" [F.435 90]. In the following, we will use the term X.435 instead of X.435 / F.435.

The main features of X.435 are described in this chapter. In part 4.2, a general introduction to EDI messaging is given including all its components : EDI messaging system, EDI Messaging User, EDI User Agent, EDI Message Store, EDI Access Unit and Message Transfer System. This is followed by the description of the EDI Message, the EDI Notification and the EDI Forwarding. The part 4.3 is dedicated to the EDI responsibility. In a fourth and last part, called other topics, we will first describe the EDI Security capabilities. This will be followed by the naming, the addressing and the use of directory for EDI and by several explanations about possible physical implementations of EDI applications. This part is ended by a description of the EDI Message Store additional features.

The appendix 7 gives a list describing how X.435 recommendation suits EDI requirements for communication.

This chapter is an abstract of [X.435 90] and [F.435 90].

4.2 EDI Messaging

4.2.1. Introduction to EDI Messaging

The EDI Messaging (EDIMG) provides the EDI Messaging Service. The EDIMG consists of an exchange of two types of objects : EDI Messages (EDIMs) and EDI Notifications (EDINs). An EDI Message is a X.400 Message carrying one EDI Interchange. An EDI Notification is an acknowledgment that an EDI Message recipient sends to the originator of the EDIM.

The environment in which the EDI Messaging works is called the EDI Messaging Environment (EDIME). The EDIME consists of a single EDI Messaging System (EDIMS) and of a group of EDI Messaging Users (EDIMG Users). See figure 4.2.1.

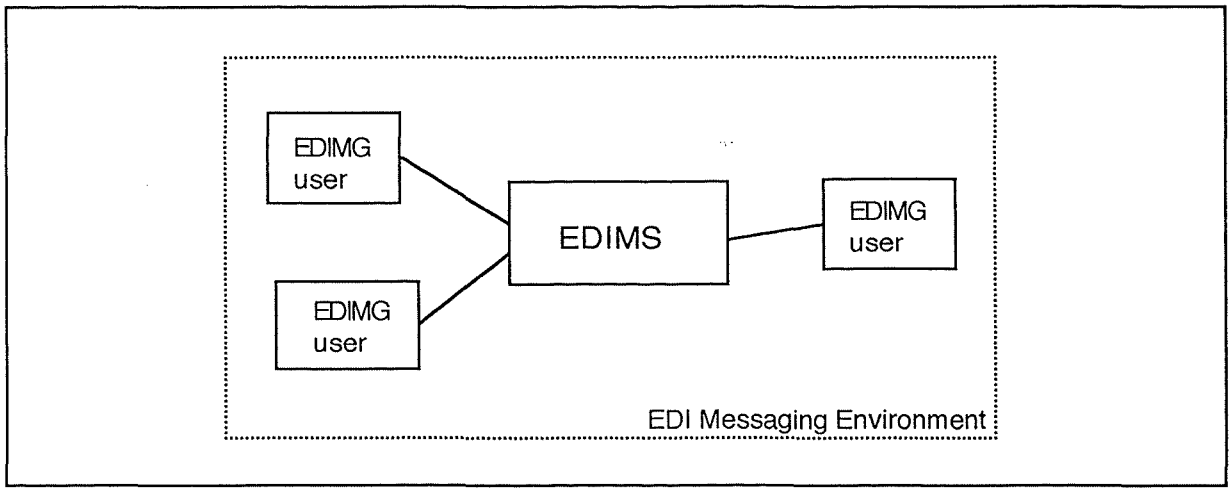

Fig 4.2.1 : EDI Messaging Environment

An EDI Messaging User (EDIMG User) is a computer process also called an EDI application. An EDIMG User receives, originates, or both receives and originates EDIMs and EDINs. An EDIMG User access the EDIMS through Abstract Ports (Origination and Reception ports). See figure 4.2.3.

The EDIMS is the subject of the next point.

4.2.2. EDI Messaging System

The X.435 recommendation defines the EDI Messaging System (EDIMS) as an Abstract Object by means of which all EDIMG Users communicate with one another in the EDI Messaging Environment.

The EDIMS is made of Abstract Objects which interact with each other. These objects are : EDI User Agent (EDI-UA), EDI Message Store (EDI-MS), Message Transfer System and EDI

Access Unit (EDI-AU). EDI-UAs, EDI-MSs and EDI-AUs are the objects by which EDIMS provides Abstract Service to EDIMG Users. The figure 4.2.2 shows them in the EDI Messaging System.

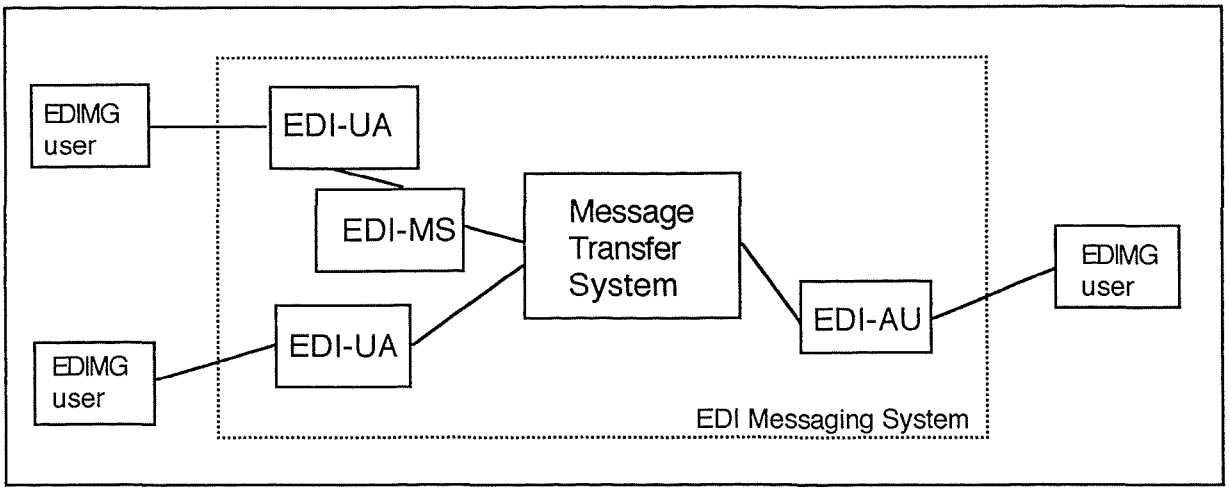

Fig 4.2.2 : EDI Messaging System

- EDI User Agent

An EDI User Agent (EDI-UA) is a UA in the X.400 terminology. An EDI-UA is a dedicated user agent specially designed for assisting a single EDIMG User to exchange EDIMs and EDINs through the EDI Messaging System. The EDIMS can contain any number of EDI-UAs.

An EDI-UA is joined to an EDIMG User by Abstract Ports. And an EDI-UA can be joined directly to the MTS or indirectly via the EDI-MS using its Abstract Ports. The EDI-UA accesses the MTS through its Delivery, Submission and Administration Ports. The EDI-UA accesses the EDI-MS through its Retrieval, Indirect Submission and Administration Ports. See figure 4.2.3.

An EDI-UA must employ MTS in order to provide the EDIMS Abstract Service to its user. A full list of EDI User Agent Abstract Operations can be found in [X.435 90].

- EDI Message Store

An EDI-MS is tailored so as to assist a single EDI-UA. It helps the EDI-UA to submit, take delivery, store or retrieve EDIMs and EDINs.

The EDI-MS is an optional component in a MHS. It is located between EDI-UA and MTS. It is linked to both by Abstract Ports. And EDI-MS is linked to MTS by Delivery, Direct Submission and Administration Ports. See figure 4.2.3.

MSs provide the store and retrieval Abstract Operations described in recommendation X.413. These are also available for EDI Messaging, but specific Abstract Operations, described in

[X.435 90], would qualify the MS as a specific dedicated MS for EDI Messaging. More explanations are given over these specific operations in part 4.4.4.

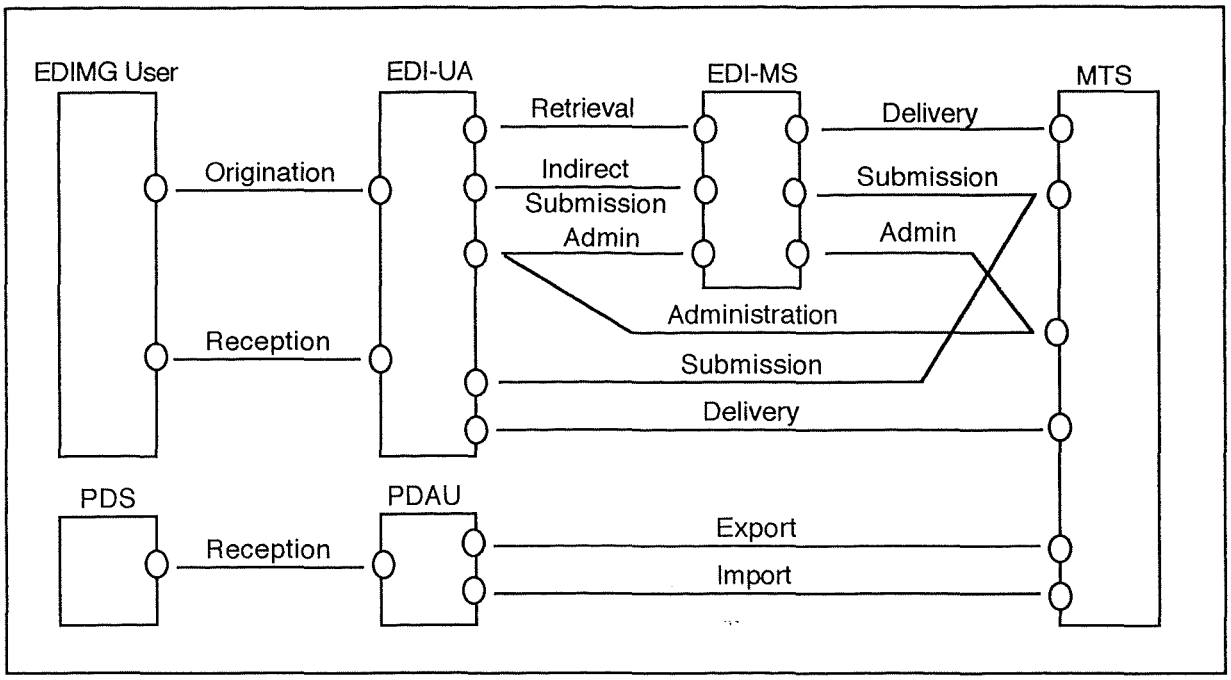

Figure 4.2.3 : EDI Messaging Abstract ports

- EDI Access Unit

An EDI Access Unit (EDI-AU) is a way by which a particular type of external EDIMG User can be reached by the EDI Messaging System. One type of access is the Physical Delivery Access Unit (PDAU) which produces a hard copy (paper, disk, tape,...) and the envelope for sending the message physically. The PDAU helps indirect users to receive EDIMs from the MTS through the Export Abstract Port. See figure 4.2.3. The PDAU does not provide the ability to originate EDIMs.

An EDI-AU can generate EDI Notifications if requested. The notifications are sent through the Import Abstract Port. See figure 4.2.3.

Other types of EDI-AU, like Facsimile Access Units, may be on further standardization.

- Message Transfer System

The Message Transfer System (MTS), as defined in 2.3.8, is in this context used for conveying EDIMs and EDINs between EDI-UAs, EDI-MSs and EDI-AUs. There is only one MTS in the EDIMS. The MTS used in the EDI Messaging Service is the one defined in X.400 recommendations. EDI-UAs, EDI-MSs and EDI-AUs are using the MT Services. Those are specified in the X.411 recommendation.

4.2.3. EDI Messaging System Functional Model

Cooperating EDI-UAs allow EDIMG Users to exchange EDIMs and EDINs. EDI-MSs provide usual MS Abstract Operations and specific EDI Abstract Operations. The Physical Delivery Access Unit allows the sending of EDIMs outside the EDI Messaging System. The telematic agent (not shown in figure 4.2.4), will allow access to telematic services through dedicated AUs and may be under further standardization.

The figure 4.2.4 shows the functional model of an EDI Messaging System.

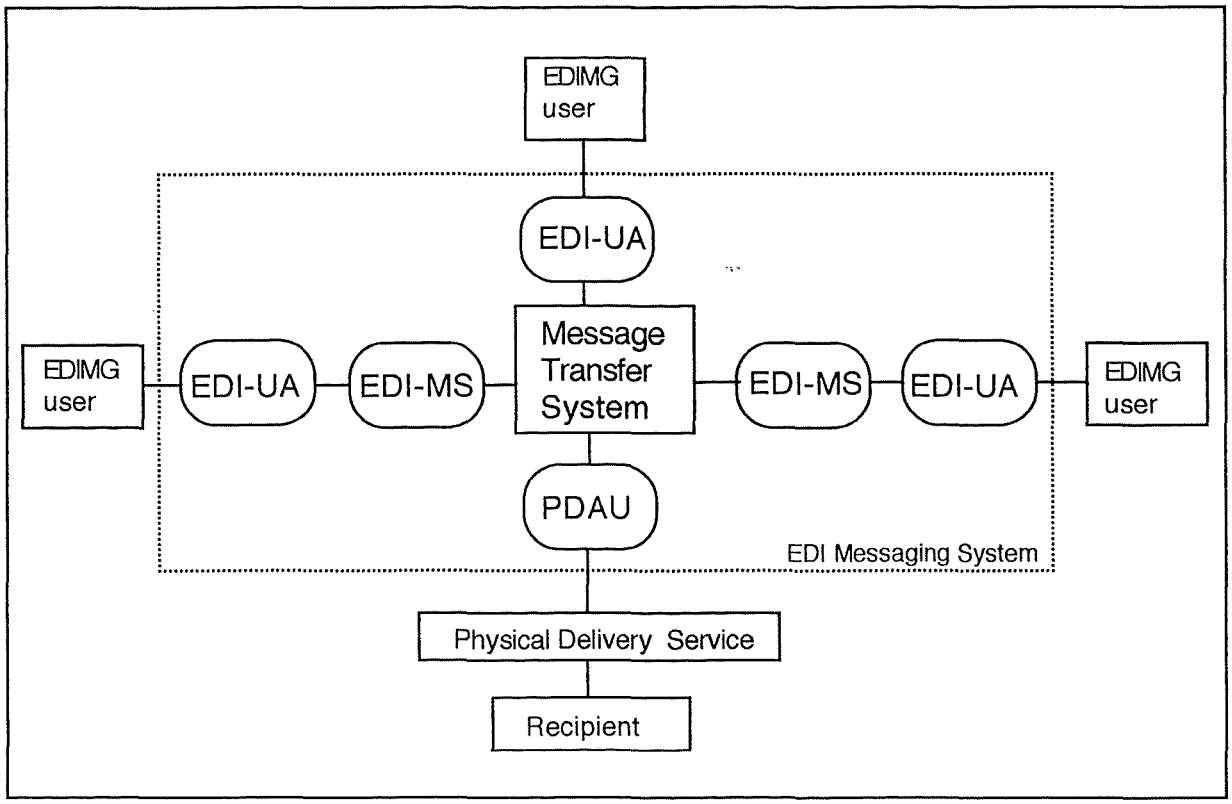

Fig 4.2.4 : EDI Messaging System Functional Model

4.2.4. EDI Message

Definition of an EDI Message.

An EDI Message (EDIM) is a X.400 message carrying one EDI Interchange. An EDI Message (EDIM) is a message type in the X.400 terminology. An EDIM carries one EDI Interchange which can contains one or more EDI message(s) in the UN/EDIFACT terminology. In all chapters, an EDI Message is a message in the X.435 terminology and an EDI message is a

message in the UN/EDIFACT terminology. Examples of use of EDIMs can be found in part 4.3. Optionally, an EDIM can carry other data related to the EDI Interchange (graphics, figures, drawings, free memos, ...).

An EDIM is usually sent by one EDIMG User to one other EDIMG User. Indeed, a recipient can forward the EDIM to another EDIMG User. The EDI Forwarding will be introduced in part 4.2.6.

Structure of an EDI Message

An EDIM consists of a heading and a body. The heading is made of numerous fields containing X.400 and EDI information. The body contains one EDI body part and other optional parts. Figure 4.2.5 describes the structure of an EDI Message. In reality, there are two body part types defined : the EDI body part and the EDIM body part. Only the EDI body part is explained is this part. The EDIM body part is concerned with forwarding. It will be detailed later in part 4.2.6. It is not shown in figure 4.2.5.

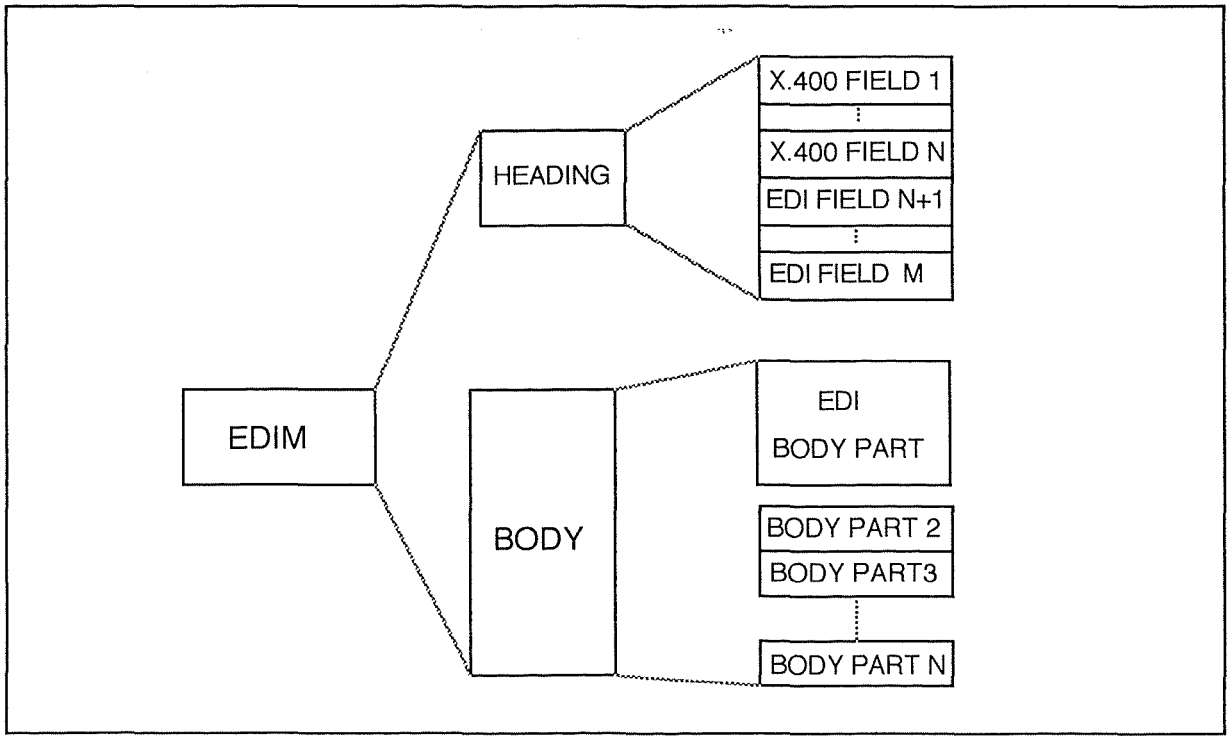

Fig 4.2.5 : EDI Message structure

The EDIM Heading

The EDIM heading contains information used by the EDI-UA in order to provide services requested by the EDIMG User. The services requested are described in [F.435 90] as elements of service. For most elements of service there is a corresponding EDIM heading field. The description of an EDIM heading in [X.435 90] divides the fields into two classes : X.400 fields

and EDI Interchange fields. The X.400 fields are concerned with EDIM information and the EDI fields are concerned with EDI Interchange information.

The X.400 fields are :

- this-EDIM : This field is a unique identifier for each EDIM. The type of this field is an EDIM Identifier which identifies an EDIM in an unambiguous, global and unique way.

- originator : This field identifies the EDIM's originator. The use of this field is optional.

- recipients : This recipients field identifies the user who is the recipient of the EDIM. Multiple recipients are allowed. This field also contains the type of notifications requested by the originator. The use of this field is optional.

- edin-receiver : This field identifies the recipient to whom the notifications have to be sent. This allows the EDIM sender to designate a specific EDI-UA to process all incoming EDINs. The use of this field is optional.

- responsibility-forwarded : This field is a boolean. Its use will be treated in part 4.3. The default value of this field is false.

- edi-bodypart-type : This field indicates which syntax has been used for coding the EDI Interchange (UN/EDIFACT, ANSI X.12, UN/TDI, PRIVATE). The default value of this field is UN/EDIFACT.

- incomplete-copy : This field indicates if body parts of the EDIM have been removed before forwarding. The default value of this field is false.

- expiry-time : This field indicates when the EDIM looses its validity. The use of this field is optional.

- related-message : This field identifies messages related to the current EDIM. The use of this field is optional.

- obsoleted-EDIMs : This field indicates one or more message(s) that the current EDIM obsoletes. The use of this field is optional.

- edi-application-security-elements : This field allows EDIMG Users to exchange agreed security elements. The use of this field is optional.

- cross-referencing-information : This field allows the EDIMG User to reference body parts in the same EDIM or within other EDIMs. The use of cross referencing will be discussed later in this part. The use of this field is optional.

The EDI Interchange fields are :

- edi-message-type : This field indicates the message type (INVOICE, ORDER, ...). This information comes from the UN/EDIFACT UNH segment. See appendix 6. The use of this field is optional.

- service-string-advice : This field indicates the separators used in the EDI Interchange (such as " '+:? ").This information comes from the UN/EDIFACT UNA segment. The use of this field is optional.

- syntax-identifier : This field indicates the syntax used and its version. (e.g. : "UNOA:1"). This information comes from the UN/EDIFACT UNB segment "syntax identifier".See appendix 6. The use of this field is optional.

- interchange-sender : This field indicates the sender of the EDI Interchange (e.g. : "56879"). This information comes from the UN/EDIFACT UNB segment "sender identification". The use of this field is optional.

- date-and-time-of-preparation : This field indicates date and time of preparation of the current EDIM (e.g. : " 910403:1651"). This information comes from the UN/EDIFACT UNB segment 'date/time of preparation'. The use of this field is optional.

- application-reference : This field provides a general reference to an application (e.g. : "invoice 91/25"). This information comes from the UN/EDIFACT UNB segment 'application reference'. The use of this field is optional.

The EDI Interchange fields are copied in the EDIM heading in ASN.1 in order to allow the receiver EDI-UA to parse these data. Otherwise the EDI-UA should be able to parse EDI syntaxes like UN/EDIFACT or ANSI X.12.

The EDIM Body

The EDIM body is made of one EDI body part and of several optional body parts. See figure 4.2.5.

The EDI body part is an octet string which carries one EDI Interchange from one EDIMG User to another one. The EDI Interchange can be coded according to any syntax (UN/EDIFACT, ANSI X.12, UN/TDI, private, ...).

The other body parts can be of any other type (i.e. : free text, drawings,...) that the originator wishes to add to the EDIM. It has been recognized that there is a number of situations where additional information which have different format types are important for business purposes. The "Cross referencing Information " field is used in the EDIM heading, in order to link the additional information and the EDI Interchange. The "Cross Referencing information" field

contains the reference number of the additional information held in a reference segment in the EDI Interchange and the body part number.

The figure 4.2.6 shows an example of optional body part.

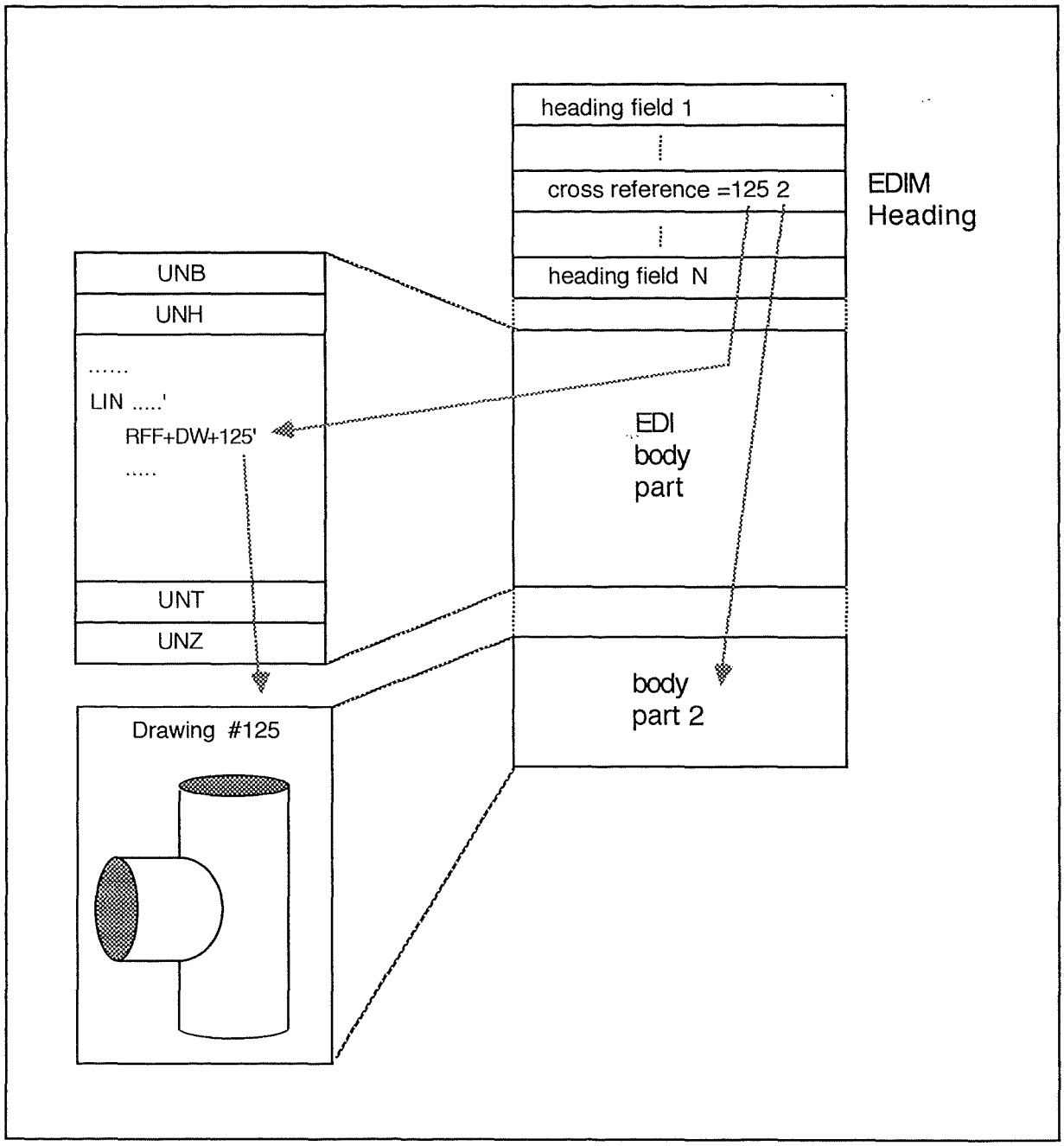

Fig 4.2.6 : Example of optional body part in an EDIM

4.2.5 EDI Notification

Definition of an EDI Notification

An EDI Notification (EDIN) is the acknowledgement that the recipient EDI-UA/EDI-MS sends to the originator EDI-UA of an EDIM. The recipient EDI-UA/EDI-MS only sends an EDIN if it is requested by the EDIM's originator. EDI Notifications consist of one of these three types :

a) Positive Notification (PN) : This is an EDIN which reports that the recipient EDI-UA/EDI-MS accepts responsibility for the received EDIM.
b) Negative Notification (NN) : This is an EDIN which reports that the recipient EDI-UA/EDI-MS rejects responsibility for the received EDIM.
c) Forwarded Notification (FN) : This an EDIN which reports that the recipient EDI-UA/EDI-MS refuses responsibility and forwards responsibility to another EDI-UA/EDI-MS both with the received EDIM.

The concept of EDI Responsibility will be defined in part 4.3.

Structure of EDI Notification

Each of the three types of notification (PN, NN, FN) contains common fields. Furthermore there are specific fields for each notification. The common fields are described first and below explanations for each kind of notification fields are given.The figure 4.2.7 shows the structure of an EDIN. Examples of the use of EDINs can be found in part 4.3.

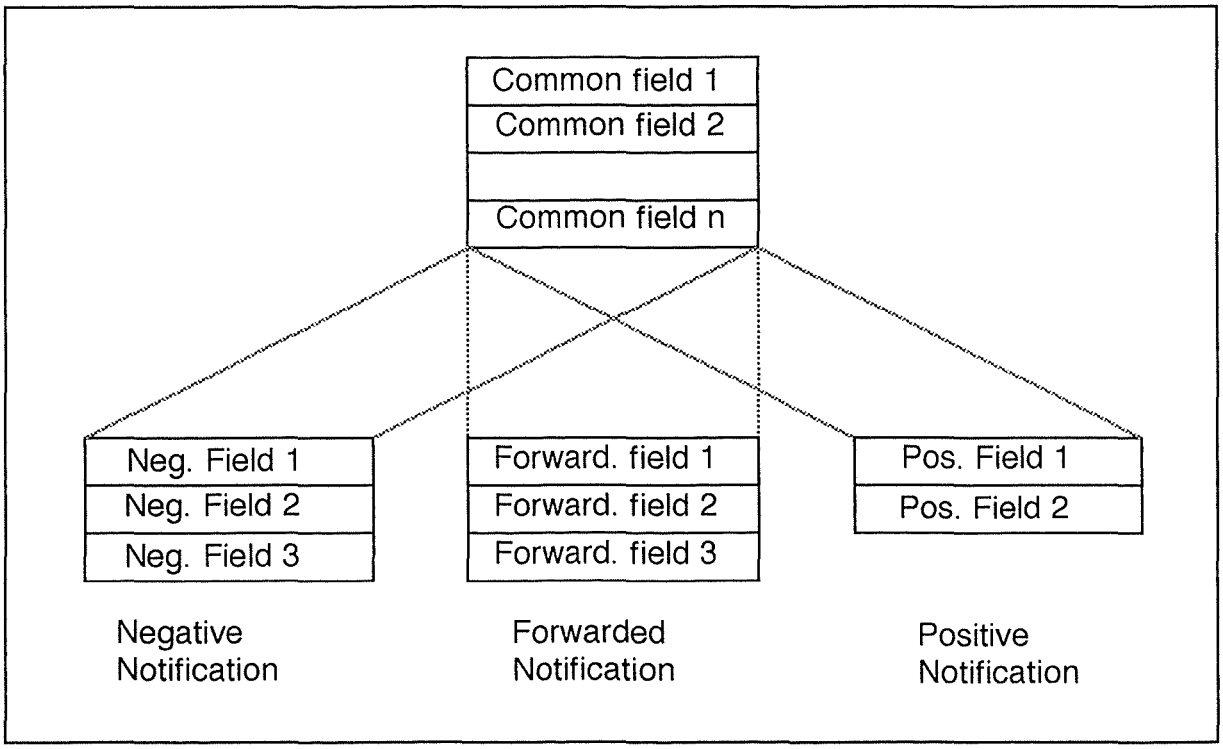

Fig 4.2.7: EDI Notification structure

EDIN Common Fields

The EDIN common fields are defined in [X.435 90]. They are :

- subject-edim : The subject-edim field is the identifier of the EDIM which has been received and for which the EDIN is being sent.

- edin-originator : The edin-originator field is the O/R Name of the EDI-UA/EDI-MS sending the EDI Notification.

- first-recipient : The first-recipient field is the O/R Name of the first recipient of the forwarding chain. This field allows the EDIN recipient to find out to which EDIM the EDIN is linked. The use of this field is optional.

- notification-time : The notification-time field is the time at which the notification was generated by its originator.

- notification-security-elements : The notification-security-elements field is used to prove non-repudiation of content received by the 'EDI application security' services. The use of this field is optional.

- edin-initiator : The edin-initiator field is the type of 'originator' of the EDIN. The EDIN can be generated by the EDI-UA without interaction with the EDIMG User, by the EDI-UA at the request of the EDIMG User, or by the EDI-MS without interaction with the EDIMG User. This field type has to be defined in bilateral agreements between the sender and the receiver.

- notification-extensions : The notification-extensions field is made for future extensions of the EDIN. No extension is presently defined. The use of this filed is optional.

PN, NN and FN specific fields

The fields that are specific for each type of EDI Notification are the reason code, supplementary information and extension. There is no reason code field in the Positive Notification, but it is contained in the Negative Notification and the Forwarding Notification. The supplementary information field can be used to send supplementary information to clarify the notification. The extension field allows further extension to the notification. Supplementary information and extension fields are optional.

The Negative Notification Reason code can be generated by the EDI-UA, the EDI-MS, the EDIMG User (EDI application) or the Physical Delivery Access Unit.

- The codes generated by the EDI-UA or the EDI-MS are either basic codes specified in [X.435 90] for the service element 'EDI Notification Request' or diagnostic codes included in the Negative Notification definition in [X.435 90].

- The codes generated by the EDIMG User are either basic codes specified in Negative Notification definition in [X.435 90] or Diagnostic codes which are EDIMG User defined reasons.

- The codes generated by the Physical Delivery Access Unit are either basic codes specified in Negative Notification definition in [X.435 90] or Diagnostic codes used to specify the error signaled in the basic reason.

Forwarded Notifications can be generated by an EDI-UA or by an EDI-MS and if the originator has requested a FN. The structure of Forwarded Notification reason codes is essentially the same as the NN Reason codes. The FN Reason Codes are listed in [X.435 90].

The use of notifications is the subject of part 4.3.

4.2.6 EDI Forwarding

EDI Forwarding is defined in the F.435 recommendation as "the onward transfer of received EDIM to one or more recipients determined by the forwarding EDI-UA/EDI-MS. EDI Forwarding takes place when an EDI Message, having been delivered to an EDI-UA/EDI-MS is forwarded onward to another EDI-UA or EDI-MS." The EDIN can be generated by the EDI-UA without interaction with the EDIMG User, by the EDI-UA at the request of the EDIMG User or by the EDI-MS without interaction with the EDIMG User. A distinction has to be made between X.400 forwarding and EDI Forwarding. X.400 forwarding is the relay between two cooperating MTAs, thus at the MTS level. The EDI Forwarding is the onward transfer between EDI-UAs or EDI-MSs, thus at the MTS user level. In order to understand the utility of the forwarding, three examples are given below [HILL 90] :

- An EDI-UA plays the role of a gateway for a company. This means that all the EDIMs for a company are addressed to the EDI-UA gateway. For each EDIM received the gateway forwards it to its final recipient. The aim of this feasibility is to send specific body parts to specific EDIMG Users (EDI applications) inside the company. This scenario would allow the organization to collect, in a centralised way, information about all incoming EDI Messages for archiving, auditing, etc... purposes.

- A private or public company provides Value Added Network (VAN) services. The EDI-UA is a gateway and is part of the VAN provider implementation. For each EDIM received the Gateway forwards it to its final recipient. This can be done through the MTS using X.435 protocol or using a proprietary protocol directly from the VAN to the recipient. The aim of this feasibility is to provide X.435 facilities to VAN users, even if they are not connected to X.435.

- An EDI-UA is used by a company that can only process a limited number of EDIM types, and thus delegates all the other EDIMs to an outside company . So the EDI-UA accepts certain types of EDIM and forwards the others to another EDIMG User.

The EDI-UA actions are various. See figure 4.2.8. It shows the basic operations of an EDI-UA. [HILL 90]

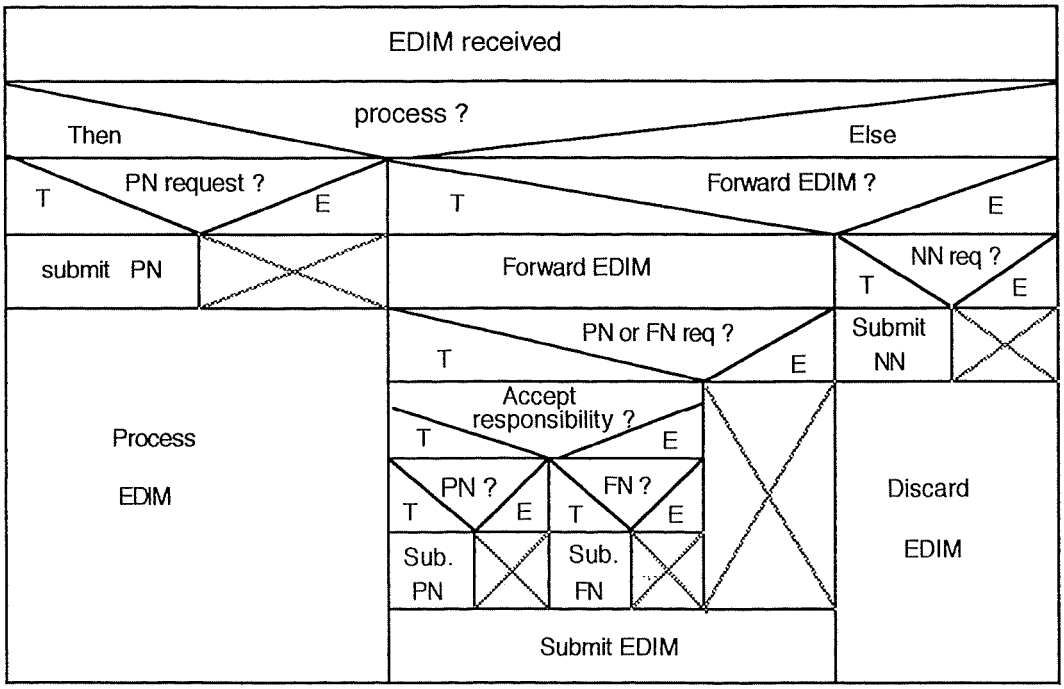

Fig 4.2.8 : EDI-UA operations flow chart

Forwarding is completely optional. So if the EDI-UA should forward the EDIM received and that there is no station of forwarding, then the EDIM is discarded.

If the EDI-UA can forward the EDIM, it does it. If there are no requests for notification then the EDI-UA can submit to its MTA the forwarded EDIM.

If the EDI-UA can forward the EDIM and does not accept the responsibility then it must forward the EDIM unchanged.

If the EDI-UA can forward the EDIM and accepts the responsibility then it can forward the EDIM with or without changes.

The concept of EDI responsibility will be explained in part 4.3. The changes that can be operated on the received EDIM in the previous case will be detailed later in this part. In order to understand how EDI Forwarding operates, three cases are described below :

CASE 1 : no forwarding

The figure 4.2.9 illustrates the case where there is no forwarding.

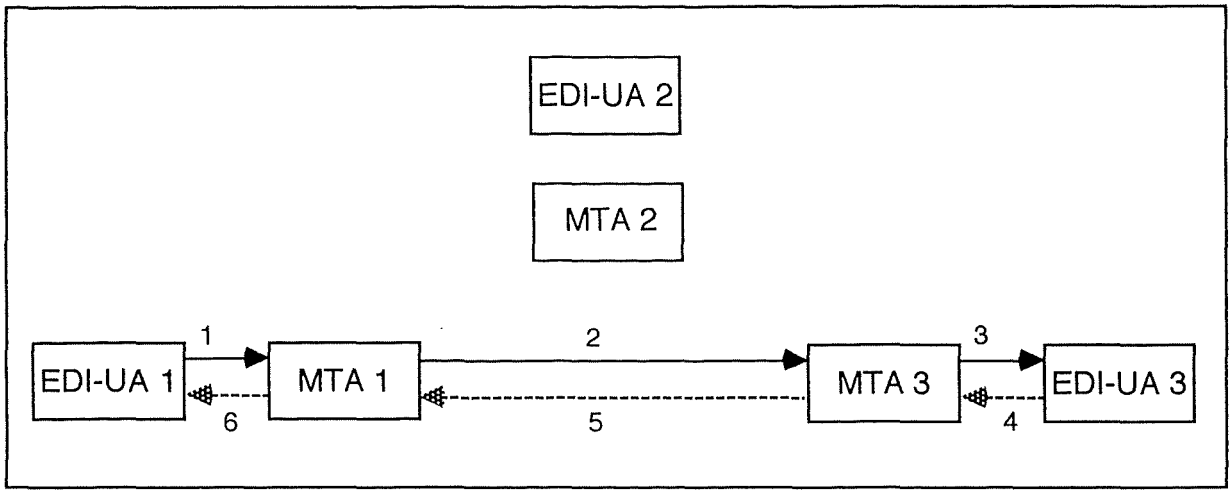

Fig 4.2.9 : No forwarding

In the figure 4.2.9,

1 - EDI-UA 1 submits an EDIM to MTA 1
2 - MTA 1 transfers the EDIM to MTA 3
3 - MTA 3 delivers the EDIM to EDI-UA 3
4 - If requested by EDI-UA 1, EDI-UA 3 submits an EDIN (PN or NN) to MTA 3
5 - MTA 3 transfers the EDIN to MTA 1
6 - MTA 1 delivers the EDIN to EDI-UA 1

CASE 2 : forwarding and responsibility not accepted

The figure 4.2.10 illustrates the case where there is forwarding and responsibility is not accepted.

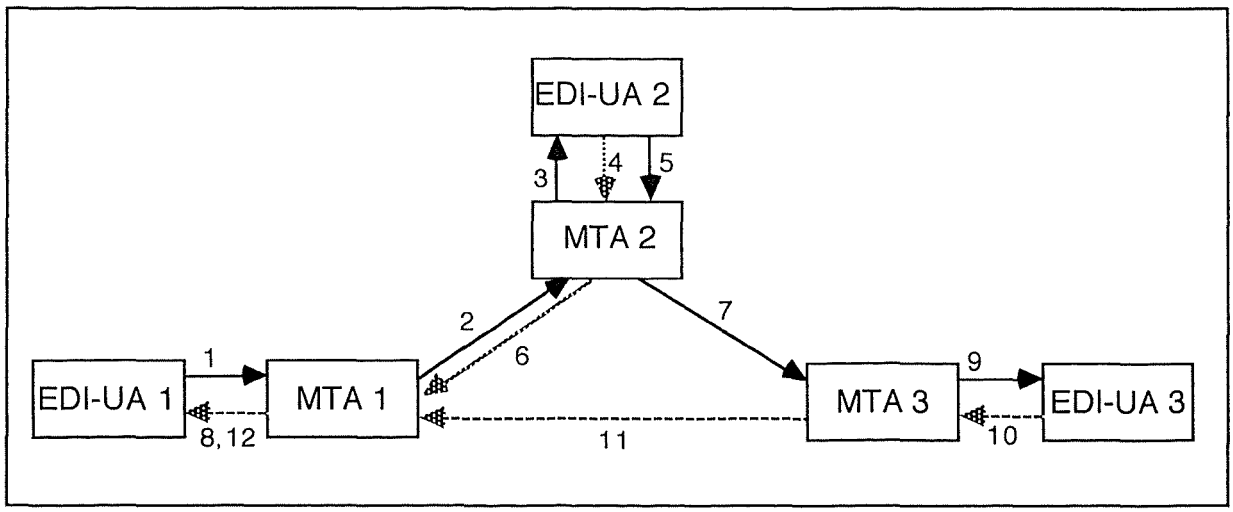

Fig 4.2.10 : Forwarding, responsibility not accepted

In the figure 4.2.10,

1 - EDI-UA 1 submits an EDIM to MTA 1
2 - MTA 1 transfers the EDIM to MTA 2
3 - MTA 2 delivers the EDIM to EDI-UA 2
4 - EDI-UA 2 decides to forward and to refuse responsibility.
If requested by EDI-UA 1, EDI-UA 2 submits a FN to MTA 2
5 - EDI-UA 2 submits the forwarded EDIM to MTA 2
6 - MTA 2 transfers the FN to MTA 1
7 - MTA 2 transfers the forwarded EDIM to MTA 3
8 - MTA 1 delivers the FN to EDI-UA 1
9 - MTA 3 delivers the EDIM to EDI-UA 3
10 - If requested by EDI-UA 1, EDI-UA 3 submits an EDIN (PN or NN) to MTA 3
11 - MTA 3 transfers the EDIN to MTA 1
12 - MTA 1 delivers the EDIN to EDI-UA 1

CASE 3 : forwarding and responsibility accepted

The figure 4.2.11 illustrates the case where there is forwarding and responsibility is accepted.

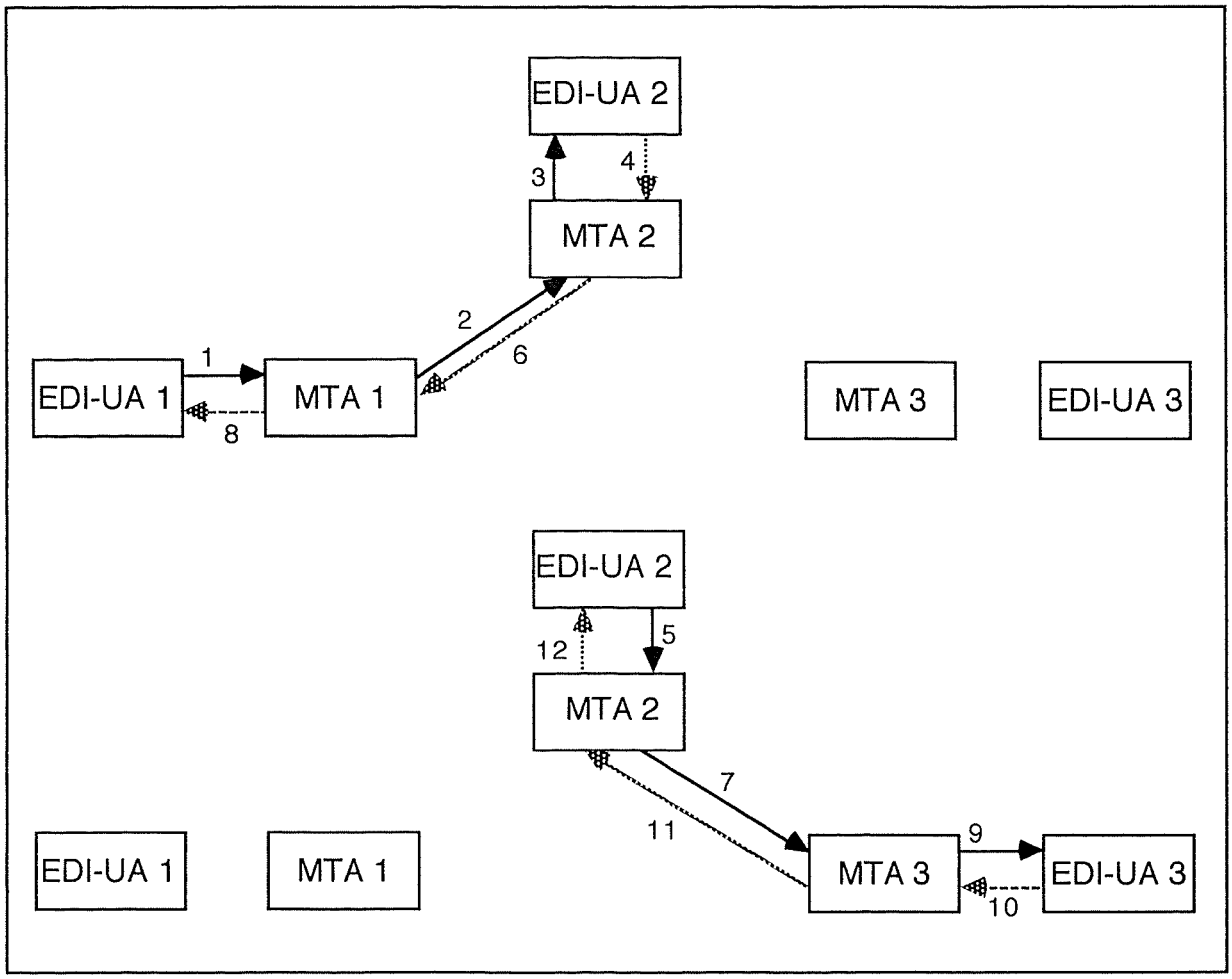

Fig 4.2.11 : Forwarding, responsibility accepted

In the figure 4.2.11,

1 - EDI-UA 1 submits an EDIM to MTA 1
2 - MTA 1 transfers the EDIM to MTA 2
3 - MTA 2 delivers the EDIM to EDI-UA 2
4 - EDI-UA 2 decides to forward and to accept responsibility.
If requested by EDI-UA 1, EDI-UA 2 submits an EDIN (PN or NN) to MTA 2
5 - EDI-UA 2 submits the forwarded EDIM to MTA 2
6 - MTA 2 transfers the EDIN to MTA 1
7 - MTA 2 transfers the forwarded EDIM to MTA 3
8 - MTA 1 delivers the EDIN to EDI-UA 1
9 - MTA 3 delivers the EDIM to EDI-UA 3
10 - If requested by EDI-UA 2, EDI-UA 3 submits an EDIN (PN or NN) to MTA 3
11 - MTA 3 transfers the EDIN to MTA 2
12 - MTA 2 delivers the EDIN to EDI-UA 2

As indicated in part 4.2.4 about EDI Messages, there are two body part types defined : an EDI body part, already explained, and an EDIM body part. The EDIM body part type is concerned with forwarding. The EDIM body part is the subject of the following.

As explained previously, an EDIM can be forwarded with or without changes, according to the rules explained.

In the case with no changes, the EDIM is packed into an EDIM body part, which is the only body part of the new EDIM, as shown in figure 4.2.12. A new set of heading fields is created. An example of this case is given in part 4.3.3.

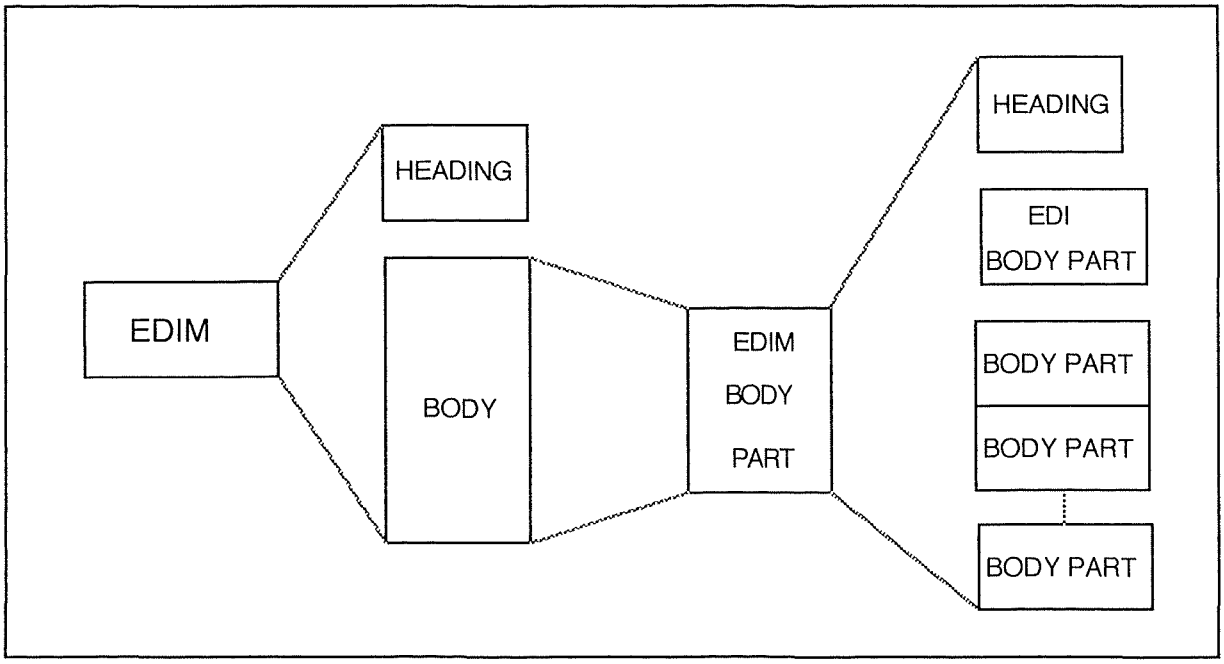

Figure 4.2.12 : Unchanged forwarded EDIM

In the case of an EDIM forwarded with changes, the body structure is more complicated. The only changes allowed on the forwarded EDIM are to drop body parts and to add new ones. Thus no changes can be made to the body parts themselves. Otherwise it would be impossible for the final recipient to see if changes have been made or not in an EDI Message since it left its originator. See figure 4.2.13. A new set of heading fields is created. An example of this case is given in part 4.3.4.

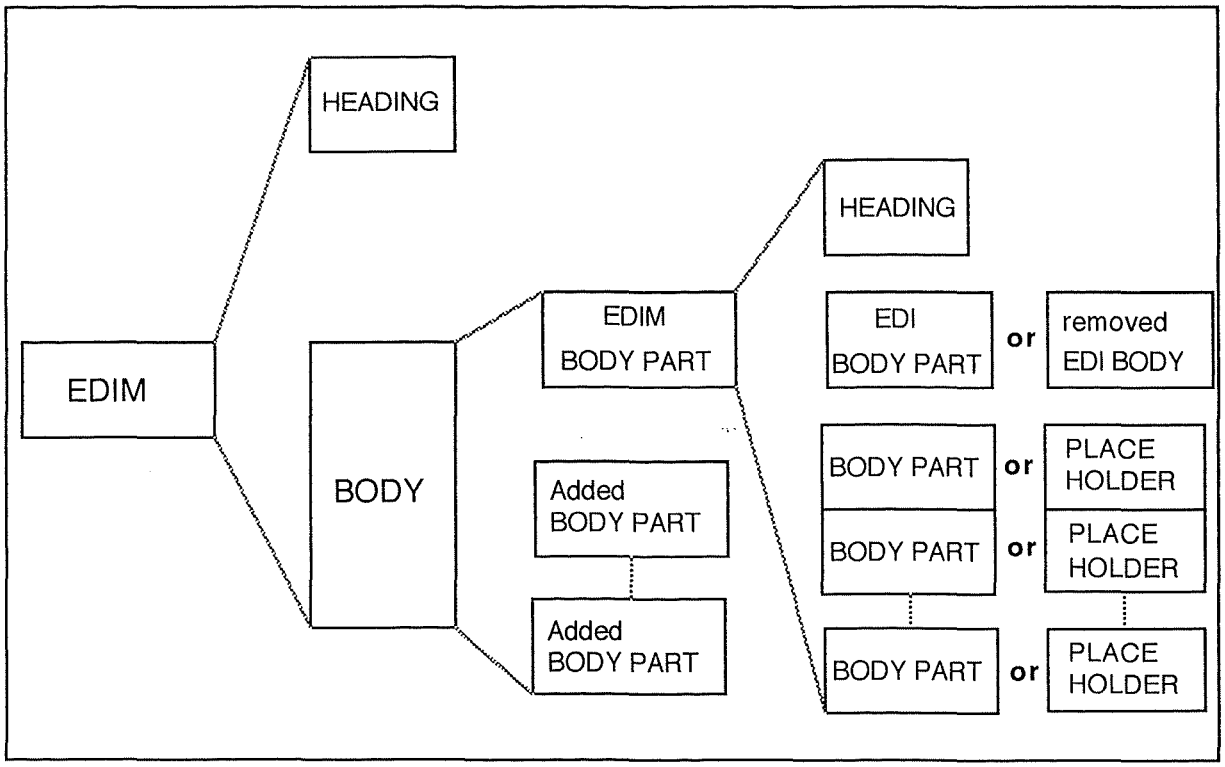

Fig 4.2.13 : EDIM forwarded with changes

The following describes the most important rules relating to the dropping or the adding of body parts in a forwarded EDIM with changes :

If the EDI body part is dropped from a forwarded EDIM then it must be replaced by a Removed EDI Body.

If a body part is dropped from a forwarded EDIM then it must be replaced by a Place Holder .

If a body part is added to a forwarded EDIM then it appears in a Added body part below the EDIM body part. The body part that is added can not be an EDI Interchange. One EDIM can carry only one EDI Interchange.

A body part containing a forwarded EDIM can not be dropped. This is because the original EDIM heading is inside the forwarded EDIM body part.

4.3 EDI Responsibility and Notifications

We will first introduce the concept of EDI Responsibility, after that, for each of the cases of point 4.2.6, we will first describe the heading fields, the possible evolution of the EDIM body and then the sending and receipt of notifications.

4.3.1 EDI Responsibility

The definition given in [F.435 90] is, "EDI Message Responsibility indicates whether the subject EDI Message has been made available to a specific user by its EDI user agent/message store". But a clearer definition can be given in terms of receiving and sending EDI Notifications.

EDI Notification (EDIN) is the acknowledgement that the recipient EDI-UA/EDI-MS sends to the originator EDI-UA of an EDIM. The notification must be asked by the originator in its EDIM. As indicated in part 4.2.5, there are three types of EDIN : Positive Notification (PN), Negative Notification (NN), Forwarded Notification (FN).

The receiving EDI-UA/EDI-MS generates notifications, if asked, following these rules [HILL 90] :

- Positive Notification whenever the receiving EDI-UA/EDI-MS does any of the following :

  holds the EDIM for processing by the resident EDIMG User ;

  adds or drops body parts from the received EDIM when forwarding ;

  forwards the EDIM and wishes to notify to the originator that the end of the forwarding chain is reached even if the EDIM is still forwarded. This means for the originator that the EDIM has reached the receiver and that the further evolution of the EDIM is out of its concern.

- Negative Notification whenever the receiving EDI-UA/EDI-MS rejects the EDIM if the requirements fixed by the EDIMG User are not met. The reason code field in the Negative Notification contains the source of the rejection.

- Forwarded Notification whenever the receiving EDI-UA/EDI-MS forwards the EDIM and has made no change on the EDIM (drop or add body parts) and no previous PN or NN has been sent.

So the EDI Responsibility definition becomes :

> If PN, NN or FN are requested, an EDI-UA/EDI-MS accepts EDI responsibility if it sends a PN, it refuses EDI Responsibility if it sends a NN and forwards EDI Responsibility if it sends a FN. If no notification is requested then there is no responsibility.

4.3.2 CASE 1 : No forwarding

The description of this case is given in part 4.2.6. The situation is illustrated in figure 4.3.1.

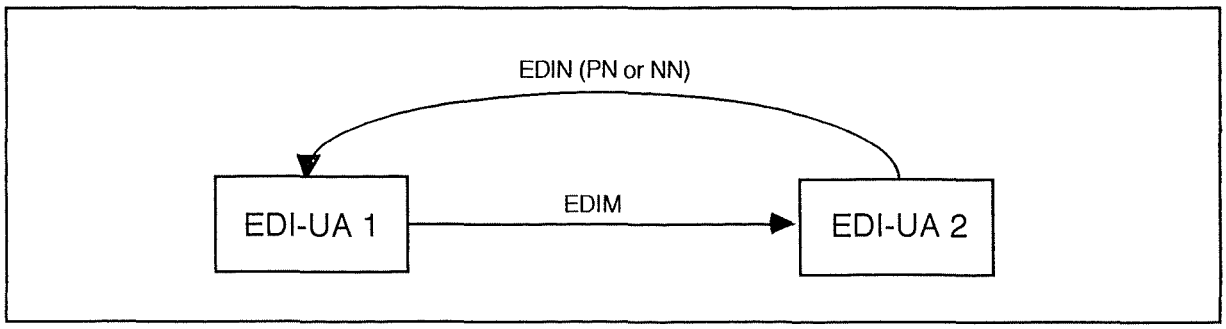

Fig 4.3.1 : No forwarding

A. EDIM heading

Explanations are given for fields which, we think are worth noting.

- this-EDIM : The 'this-EDIM' field is the identifier of the EDIM.
- originator : The 'originator' field of the EDIM is EDI-UA 1's O/R Name.
- recipients : The 'recipients' field contains EDI-UA 2's O/R Name. In the subfield 'edi-notification-requests', the originator must specify the type of notification requested.
- edin-receiver : The 'edin-receiver' field is the recipient to whom notification must be sent. This field is optional, if it is not present, the recipient will be the originator.
- responsibility-forwarded : The 'responsibility-forwarded' is a boolean field. If it is not defined, it takes its default value 'FALSE'.

See example in figure 4.3.2.

B. Sending notification

The recipient EDI-UA in this case can send two types of EDI Notifications : Positive Notifications and Negative Notifications. An EDI-UA sends a PN only if PN is requested in 'edi-notification-requests' field and the EDI-UA accepts responsibility for the EDIM. An EDI-UA sends a NN only if NN is requested in the 'edi-notification-requests' field and the EDI-UA rejects responsibility for the EDIM.

The EDIN common fields of PN (or NN) will be as specified below :

- subject-edim : The 'subject-edim' will be the EDIM Identifier 'this-EDIM' field of the incoming EDIM.
- edin-originator : The 'edin-originator' will be the recipient specified in the 'recipients' field of the incoming EDIM. In this case it is EDI-UA 2's O/R Name.
- first-recipient : The 'first-recipient ' field is optional. This field is not present in this case.

The NN reason code field will specify the error.

See example in figure 4.3.3.

In this case there is only one recipient. If there are multiple recipients then the "first-recipient " field is used and allows the EDIN recipient to distinguish the incoming EDINs.

C. Receiving notification

At most, one PN or NN is received coming from the EDIM recipient if NN and PN were requested by EDI-UA 1.

D. Example

Figures 4.3.2 and 4.3.3 give an example of type 'Case 1'.

EDIM Heading	
Field	Value
this-EDIM	BE/ RTT/ FUNDP/ INFO/ STUDENT/ EDI 91/04/04/18:37:10
originator	BE/ RTT/ FUNDP/ INFO/ STUDENT/ EDI
recipients recipient edi-notification-requests	BE/ RTT/ FUNDP/ DROIT/ CRID/ GREDIN NN PN
edin-receiver	BE/ RTT/ FUNDP/ INFO/ STUDENT/ EDI

Fig 4.3.2 : EDIM heading fields and content

EDIN Common fields	
Field	Value
subject-edim	BE/ RTT/ FUNDP/ INFO/ STUDENT/ EDI 91/04/04/18:37:10
edin-originator	BE/ RTT/ FUNDP/ DROIT/ CRID/ GREDIN

Fig 4.3.3 : EDIN common fields and content

4.3.3 CASE 2 : forwarding and responsibility not accepted

The description of this case is given in part 4.2.6. The situation is illustrated in figure 4.3.4.

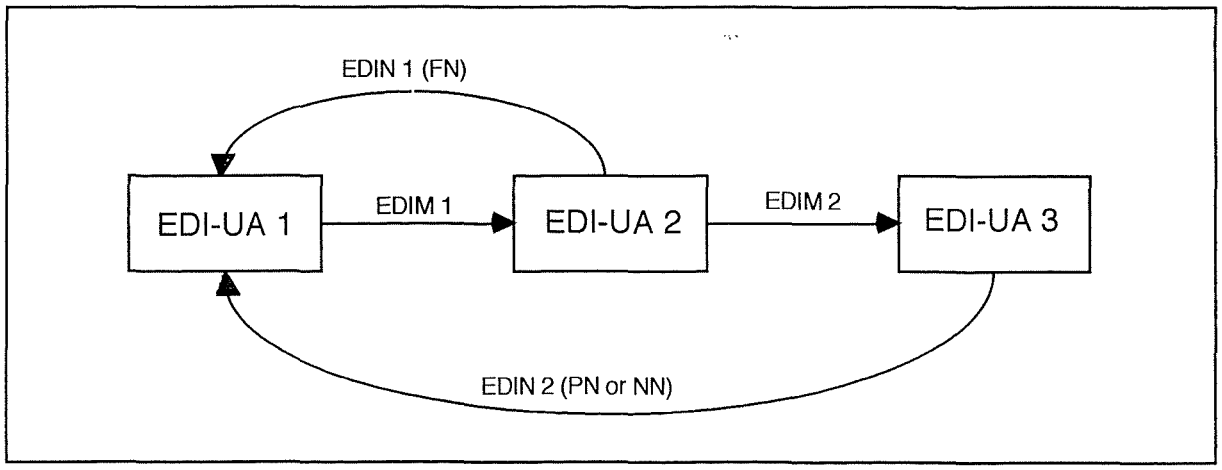

Fig 4.3.4 : Responsibility forwarded

A. EDIM heading

The EDI-UA 2 which forwards the EDIM 1 must create a new EDIM (EDIM 2). As indicated in part 4.2.6 about the EDI body part, we are in the case of forwarding without changes. The EDI-UA 2 will simply encapsulate the incoming EDIM 1 inside another EDIM (EDIM 2). Fields described below are those of the EDIM 2 heading which, we think, are worth noting.

- this-EDIM : The 'this-EDIM' is the EDIM identifier field. A new value is created by the EDI-UA 2.
- originator : The 'originator' field contains the EDI-UA 2's O/R Name.

- recipient : The 'recipient' field is created by the EDI-UA 2. If the EDIM 1 contains a 'edi-notification-requests' then one, and only one of the new recipients must have the same request for notification. Otherwise it will be impossible for EDI-UA 1 to find out the origin of the EDINs that it will receive. So if only one recipient sends an EDIN, the EDI-UA 1 will be able to match the EDIN received with the EDIM sent.
- edin-receiver : The 'edin-receiver' field contains information about the recipient to whom notification must be sent. In this case, it must contain the O/R Name of the 'edin-receiver' field of the EDIM 1. This information is required by the process controlling EDIN in order to correlate each EDIN with its 'originating' EDIM.
- responsibility-forwarded : The 'responsibility-forwarded' boolean field is set to 'TRUE'.

See example in figures 4.3.5 and 4.3.6. The figure 4.3.5 shows the EDIM 1 heading and the figure 4.3.6 shows the EDIM 2 heading.

B. Sending notification

In this case, two EDI-UAs must send notifications if requested : EDI-UA 2 has to send a Forwarded Notification (FN) and EDI-UA 3 has to send a Positive Notification (PN) or a Negative Notification (NN).

For the Forwarded Notification, EDI-UA 2 must set the following common fields :

- subject-edim : The 'subject-edim' will be the EDIM Identifier 'this-EDIM' field of the EDIM 1.
- edin-originator : The 'edin-originator' will be the recipient specified in the 'recipients' field of the EDIM 2. In this case it is EDI-UA 2's O/R Name.
- first-recipient : The 'first-recipient ' field contains the O/R Name of the first recipient of the forwarding chain. So in this case it will be EDI-UA 2's O/R Name.

For the Positive or Negative Notification, EDI-UA 3 must set the following common fields :

- subject-edim : The 'subject-edim' will be the EDIM Identifier of the 'original' EDIM (EDIM 1).
- edin-originator : The 'edin-originator' will be the recipient specified in the 'recipients' field of the EDIM 2. In this case it is EDI-UA 3's O/R Name.
- first-recipient : The 'first-recipient ' field contains the O/R Name of the first recipient of the forwarding chain. So in this case it will be EDI-UA 2's O/R Name.

The NN reason code field will specify the error.

See example in figures 4.3.7 and 4.3.8. The figure 4.3.7 shows the EDIN 1 common fields and the figure 4.3.8 shows the EDIN 2 common fields.

C. Receiving notification

If FN is requested and the EDIM is forwarded then a FN will be received by the EDI-UA 1. The 'subject-edim' field indicates which EDIM is linked with the FN and the 'first-recipient' indicates which EDI-UA has forwarded the EDIM.

If both PN and NN were requested, then EDI-UA 1 should receive either a PN or a NN. The 'subject-edim' field indicates which EDIM is linked with the notification and the 'first-recipient' indicates which EDI-UA has forwarded the EDIM.

So it is easy for the process controlling incoming notifications to correlate them.

D. Example

Figures 4.3.5 to 4.3.8 give an example of type 'Case 2'

original EDIM Heading created by EDI-UA 1	
Field	Value
this-EDIM	BE/ RTT/ FUNDP/ INFO/STUDENT/ EDI 91/04/04/18:37:10
originator	BE/ RTT/ FUNDP/ INFO/ STUDENT/ EDI
recipients recipient edi-notification-requests	BE/ RTT/ FUNDP/ DROIT/ CRID/ GREDIN NN PN FN
edin-receiver	BE/ RTT/ FUNDP/ INFO/ STUDENT/ EDIN

Fig 4.3.5 : EDIM 1 heading fields and content

forwarded EDIM Heading created by EDI-UA 2	
Field	Value
this-EDIM	BE/ RTT/ FUNDP/ INFO/STUDENT/ EDI 91/04/04/20:17:18
originator	BE/ RTT/ FUNDP/ DROIT/ CRID/ GREDIN
recipients recipient edi-notification-requests	 BE/ RTT/ FUNDP/ DROIT/ CRID/ ORDER NN PN
edin-receiver edin-receiver-name original-edim-identifier first-recipient	 BE /RTT/ FUNDP/ INFO/ STUDENT/ EDIN BE/ RTT/ FUNDP/ INFO/ STUDENT/ EDI 91/04/04/18:37:10 BE/ RTT/ FUNDP/ DROIT/ CRID/GREDIN
responsability-forwarded	TRUE

Fig 4.3.6 : EDIM 2 heading fields and content

EDIN Common fields (forwarding notification)	
Field	Value
subject-edim	BE/ RTT/ FUNDP/ INFO/ STUDENT/ EDI 91/04/04/18:37:10
edin-originator	BE/ RTT/ FUNDP/ DROIT/ CRID/ GREDIN
first-recipient	BE/ RTT/ FUNDP/ DROIT/ CRID/ GREDIN

Fig 4.3.7 : EDIN 1 FN common fields and content

EDIN Common fields (Negative or Positive Notification)	
Field	Value
subject-edim	BE/ RTT/ FUNDP/ INFO/ STUDENT/ EDI 91/04/04/18:37:10
edin-originator	BE/ RTT/ FUNDP/ DROIT/ CRID/ ORDER
first-recipient	BE/ RTT/ FUNDP/ DROIT/ CRID/ GREDIN

Fig 4.3.8 : EDIN 2 PN or NN common fields and content

4.3.4. CASE 3 : forwarding and responsibility accepted

The description of this case is given in part 4.2.6. The situation is illustrated in figure 4.3.9.

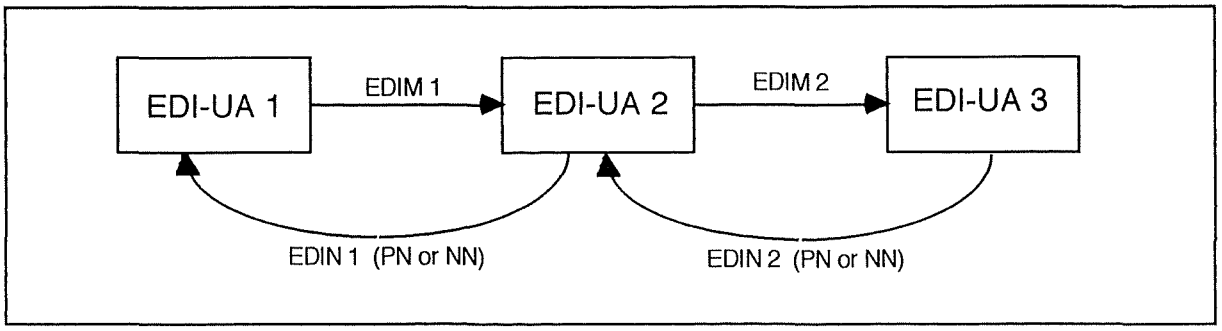

Fig 4.3.9 : Responsibility accepted

A. EDIM heading

The EDI-UA 2 which forwards the EDIM 1 must create a new EDIM (EDIM 2). As indicated in part 4.2.6 about EDI body part, we are in the case of forwarding with possible changes. The changes allowed are to drop body parts including EDI body part and to add new body parts. These changes have to be done conforming to the rules described in part 4.2.6. The EDI-UA 2 will encapsulate the incoming EDIM 1 or the modified EDIM and added body parts inside another EDIM (EDIM 2). Fields described below are those of the EDIM 2 heading which, we think, are worth noting.

- this-EDIM : The 'this-EDIM' is the EDIM identifier field. A new value is created by the EDI-UA 2.
- originator : The 'originator' field contains the EDI-UA 2's O/R Name.
- recipient : The 'recipient' field is created by the EDI-UA 2. The 'edi-notification-requests' field can contain any value.
- edin-receiver : The 'edin-receiver' field contains the recipient O/R Name to whom notification must be sent. This field is optional ; if it is not present, the recipient will be the originator (EDI-UA 2) otherwise it will be the recipient specified. It is clear that the EDI-UA should not be the same as the EDIN receiver defined in EDIM 1, in order to avoid confusion.
- responsibility-forwarded : The 'responsibility-forwarded' is a boolean field. If it is not defined, it takes its default value 'FALSE'.

See example in figures 4.3.10 and 4.3.11. The figure 4.3.10 shows the EDIM 1 heading and the figure 4.3.11 shows the EDIM 2 heading.

B. Sending notification

In this case, two EDI-UAs must send notifications if requested : EDI-UA 2 and EDI-UA 3 can send a PN or a NN.

EDI-UA 2 will send either a PN or a NN to EDI-UA 1. The EDIN common fields of PN (or NN) will be as specified below :

- subject-edim : The 'subject-edim' will be the EDIM Identifier 'this-EDIM' field of the EDIM 1.
- edin-originator : The 'edin-originator' will be the recipient specified in the 'recipients' field of the EDIM 1. In this case it is EDI-UA 2's O/R Name.
- first-recipient : The 'first-recipient ' field is optional and it is not present in this case.

EDI-UA 3 will send either a PN or a NN to EDI-UA 2. The EDIN common fields of PN (or NN) will be as specified below :

- subject-edim : The 'subject-edim' will be the EDIM Identifier 'this-EDIM' field of the EDIM 2.
- edin-originator : The 'edin-originator' will be the recipient specified in the 'recipients' field of the EDIM 2. In this case it is EDI-UA 3's O/R Name.
- first-recipient : The 'first-recipient ' field is optional and it is not present in this case.

The NN reason code field will specify the error.

See examples in figure 4.3.12 and 4.3.13. The figure 4.3.12 shows the EDIN 1 common fields and the figure 4.3.13 shows the EDIN 2 common fields.

C. Receiving notification

At most one PN or NN is received coming from the EDIM 1 recipient if NN and PN were requested by EDI-UA 1. The situation is the same for EDI-UA 2.

D. Example

Figures 4.3.10 to 4.3.13 give an example of type 'Case 3'.

original EDIM Heading created by EDI-UA 1	
Field	Value
this-EDIM	BE/ RTT/ FUNDP/ INFO/ STUDENT/ EDI 91/04/04/18:37:10
originator	BE/ RTT/ FUNDP/ INFO/ STUDENT/ EDI
recipients recipient edi-notification-requests	 BE/ RTT/ FUNDP/ DROIT/ CRID/ GREDIN NN PN FN
edin-receiver	BE/ RTT/ FUNDP/ INFO/ STUDENT/ EDIN

Fig 4.3.10 : EDIM 1 heading fields and content

forwarded EDIM Heading created by EDI-UA 2	
Field	Value
this-EDIM	BE/ RTT/ FUNDP/ INFO/ STUDENT/ EDI 91/04/04/20:17:18
originator	BE/ RTT/ FUNDP/ DROIT/ CRID/ GREDIN
recipients recipient edi-notification-requests	 BE/ RTT/ FUNDP/ DROIT/ CRID/ ORDER NN PN
edin-receiver edin-receiver-name	 BE/ RTT/ FUNDP/ INFO/ STUDENT/ EDIN

Fig 4.3.11 : EDIM 2 heading fields and content

EDIN Common fields (PN or NN) created by EDI-UA 2	
Field	Value
subject-edim	BE/ RTT/ FUNDP/ INFO/ STUDENT/ EDI 91/04/04/18:37:10
edin-originator	BE/ RTT/ FUNDP/ DROIT/ CRID/ GREDIN
first-recipient	BE/ RTT/ FUNDP/ DROIT/ CRID/ GREDIN

Fig 4.3.12 : EDIN 1 common fields and content

EDIN Common fields (PN or NN) created by EDI-UA3	
Field	Value
subject-edim	BE/ RTT/ FUNDP/ INFO/ STUDENT/ EDI 91/04/04/18:37:10
edin-originator	BE/ RTT/ FUNDP/ DROIT/ CRID/ ORDER
first-recipient	BE/ RTT/ FUNDP/ DROIT/ CRID/ ORDER

Fig 4.3.13 : EDIN 2 common fields and content

4.4 OTHER TOPICS

4.4.1. EDI Security

MHS Security capabilities as defined in part 2.3.5 (1988 X.400 Security enhancements) are used to provide the security features for the EDI Messaging System. In addition, X.435 recommendation provides additional security capabilities for EDI. The X.435 security capabilities are explained below. The way these facilities are provided is beyond the scope of this work. Nevertheless, the reader can find more information in [X.435 90] and in [CRYP 90].

Extended Security capabilities

Proof of EDI Notification :

This security facility must be able to prove to the EDIM sender that the EDIM receiver is indeed the originator of the EDIN received.

This facility can be viewed as a signed receipt returned to the message sender.

So this facility must provide a proof of the origin of the EDIN received.

Non-repudiation of EDI Notification

This security facility must be able to prove to the EDIM sender that the EDIM receiver is indeed the originator of the EDIN received and must prevent against any attempt of the EDIN originator to falsely deny the sending of it.

This facility can be viewed as a legalized signed receipt returned to the message sender.

So this facility must provide a proof of the origin of the EDIN, strong enough to avoid any attempt of the EDIN originator to falsely deny the sending of it.

Proof of Content Received

This security facility must be able to prove to the EDIM originator that the message sent has not been changed throughout the system.

This facility can be viewed as a photocopy of the message sent back to its originator by the recipient.

So this facility must provide a means to prove to the EDIM originator that the content received by the recipient is the same as the message content sent.

Non-repudiation of Content Originated

This security facility must be able to prove to the recipient of a message that the message content is unchanged throughout the system and to protect the message recipient against any attempt by the originator to falsely deny the sending of it.

This facility can be viewed as a sealed envelope with an official seal.

So this facility must provide to the recipient a proof of the origin of the EDIM and the likeness of the content received, strong enough to avoid any attempt of the EDIM originator to deny the sending of it.

Non-repudiation of Content Received

This security facility must be able to prove to the EDIM originator that the message sent has not been changed throughout the system and to protect the message originator against any attempt by the recipient to falsely deny the recept of the message.

This facility can be viewed as a legalized photocopy of the message sent back to its originator by the recipient.

So this facility must provide to the originator a proof that the content received by the recipient is the same as the EDIM content originated. This proof has to be strong enough to avoid any attempt of the EDIM recipient to falsely deny the receipt of the EDIM content.

4.4.2. Naming , Addressing and Use of Directory

This part describes how EDIMG Users are identified, how an EDIMG User can obtain the O/R Address of another EDIMG User, how an EDIMG User can obtain specific EDI information about another EDI-UA. The use of the Directory Service, as defined in CCITT X.500 recommendation, can benefit EDIMG Users as follows :

- by obtaining the O/R Address of another trading partner (EDIMG User)

- by obtaining specific information about a given EDIMG User, in order to give the possibility to exchange EDI Interchange. This set of specific information is also called the 'EDI profile'.

The use of the X.500 Directory Service is necessary for the further development of EDI relations. Indeed, nowadays, EDI relations are subject to bilateral agreements between partners but no external partner can enter in the system without negotiating those agreements. But 'open' trading relation is the norm of traditional business transactions. 'Open' trading means

that any company can do business with any other. Directory Service can provide O/R Address and EDI capabilities information without bilateral agreements. So the use of Directory Service will be a necessity for 'open' EDI relations.

EDI Naming

EDIMG Users identify each other by a name. This name is contained in the UN/EDIFACT UNB segment. It is unique in a particular EDI community but it may not be globally unique.

Each EDI community has its own organization for naming. EDI communities can be :

- industrial groups (EDIFICE, CEFIC, ...) which have formalized names recognized by an internationally naming authority (DUNS, EAN, ...) which is globally unique.

- large private corporations which have formalized names given by a multinational company which acts as a naming authority ; these names are only unique within the company's trading community.

- trading partners group around a VAN service provider (INS, AT&T ISTEL, ...) which have a free form name assigned by the trading partners themselves, with a check of the uniqueness by the operator. These names are obviously locally unique.

With all existing name forms, it is clear that it will take time for migrating to globally unique naming [LI 90].

The EDI standards, as UN/EDIFACT, define the recipient identification as an alphanumeric data element with optional qualifier and routing address data elements. Globally unique EDI Naming can be achieved using the qualifier data elements. It is assumed that the EDI Naming should contain as little addressing information as possible. Indeed the name can be seen as a static information for quite a long time, on the contrary the address can change from time to time.

Obtaining an O/R Address

The EDI Name of the recipient is contained in the UN/EDIFACT UNB "Interchange recipient". But before the EDI Interchange can be submitted, the originator EDI-UA must obtain the EDI Name of the recipient EDI-UA. This is considered as a local matter.

A first query to the Directory Service can map the EDI Name and O/R Name. And a second query to the Directory service can provide the O/R Address from the O/R Name.

The EDI-UA will access the Directory Service through a Directory User Agent (as defined in part 2.3.5 of this work). A sub-tree administered by a naming authority organization (EAN, DUNS,...) can contain EDI Names and their corresponding O/R Names.

The example described below is shown in figure 4.4.1. An EDIMG User want to send an EDI Interchange to another one. It is assumed that the originator knows the EDI Name of the recipient.

If the EDI Name is globally unique the country name and the naming authority organization name can be derived from a qualifier code within the EDI Name. Otherwise with the EDI Name, the country name and the naming authority organization name, the O/R Name corresponding to the EDI Name can be found using the Directory Service. In this example the country name is BE, the naming authority organization is EAN and the EDI Name is 56879. The corresponding O/R Name is FUNDP/INFO/STUDENT.

In the second query with the O/R Name, the Directory Service gives the O/R Address (country = BE, ADMD = RTT, PRMD = FUNDP, organization = INFO, organization unit = STUDENT and the common name = EDI).

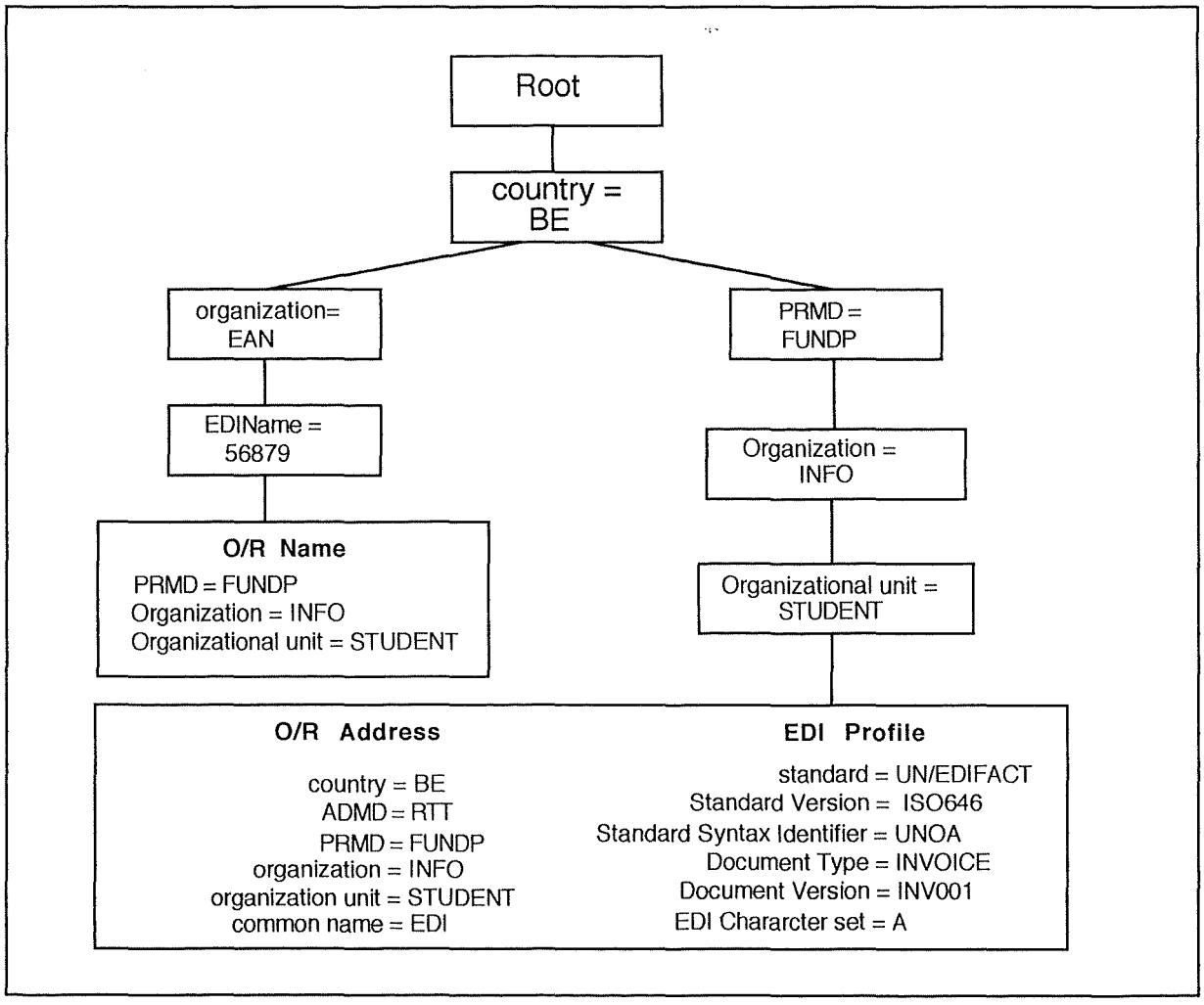

Fig 4.4.1 : Example of Directory accesses

Obtaining the EDI Profile

The Directory Service can be used for providing information about an EDI-UA. This information is contained in Directory Attributes representing the EDI capabilities of an EDI-UA. The following are worth noting :

- standard : defining the standard used by the EDI-UA
- standard version : defining the version of the standard used.
- standard syntax identifier : defining the syntax used
- document type : defining the type of EDI messages supported
- document version : defining the version of the EDI messages supported
- EDI character set : defining the character set supported

See example 4.4.1. In this example the EDI profile, reached using the same process as for O/R Address is : standard = UN/EDIFACT, Standard version =ISO646, standard syntax identifier = UNOA, document type = ORDER, document version = ORD001 and the EDI character set = A.

4.4.3 Physical implementation [HILL 90]

There are many ways to implement EDI Applications. Below, we will describe five scenarios :

- The EDI-UA is associated with one EDIMG User
- The EDI-UA is a gateway which does not accept responsibility and forwards the responsibility and the EDIM to its final recipient.
- The EDI-UA is a gateway which accepts responsibility for all incoming EDI messages and acknowledges the originator if requested.
- The EDI-UA is part of a Value Added Network.
- The EDI-UA is used by a company which forwards some EDIM type to another company for processing.

The EDI-UA serves one application

This situation is described in figure 4.4.2

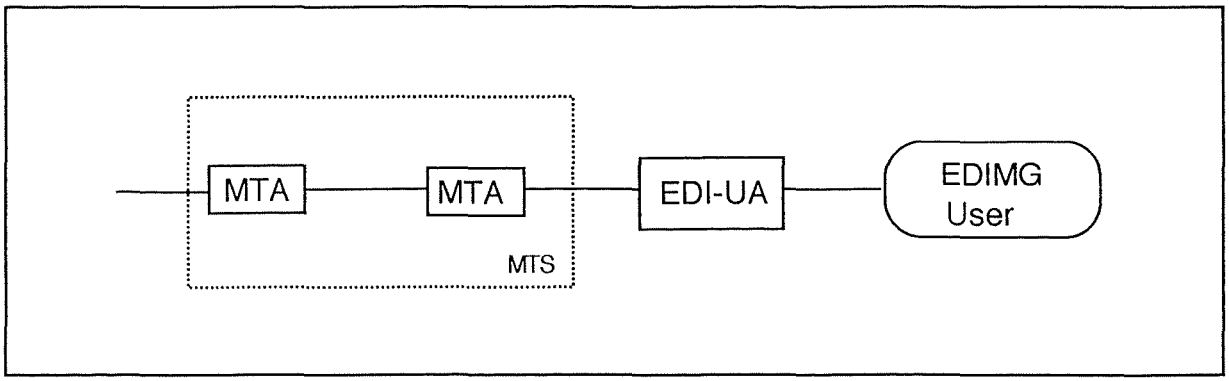

Fig 4.4.2 : EDI-UA serves one application

In this situation there is a one-to-one relationship between the EDI-UA and the EDIMG User. The EDI-UA is able to refuse or accept the responsibility and creates EDIN following the case, if asked.

The EDI-UA is a Gateway and does not accept responsibility

In this situation, an EDI-UA plays the role of a gateway. This means that all the EDIMs for a company are addressed to the Gateway EDI-UA. Thus only the EDI-UA Gateway is known to the outside world. The EDI-UA 1 and EDI-UA 2 are only known by the Gateway. The Gateway forwards the responsibility and creates EDIN (FN), if asked. The Gateway will never send Positive or Negative Notifications. The PN or NN are sent by the final recipient EDI-UA (EDI-UA 1 or EDI-UA 2). For each EDIM received the Gateway forwards it to its final recipient with the responsibility. This has to be done through the MTS using X.435 Protocol. The gateway in this case can not operate changes on the incoming EDIMs.

This situation is described in figure 4.4.3.

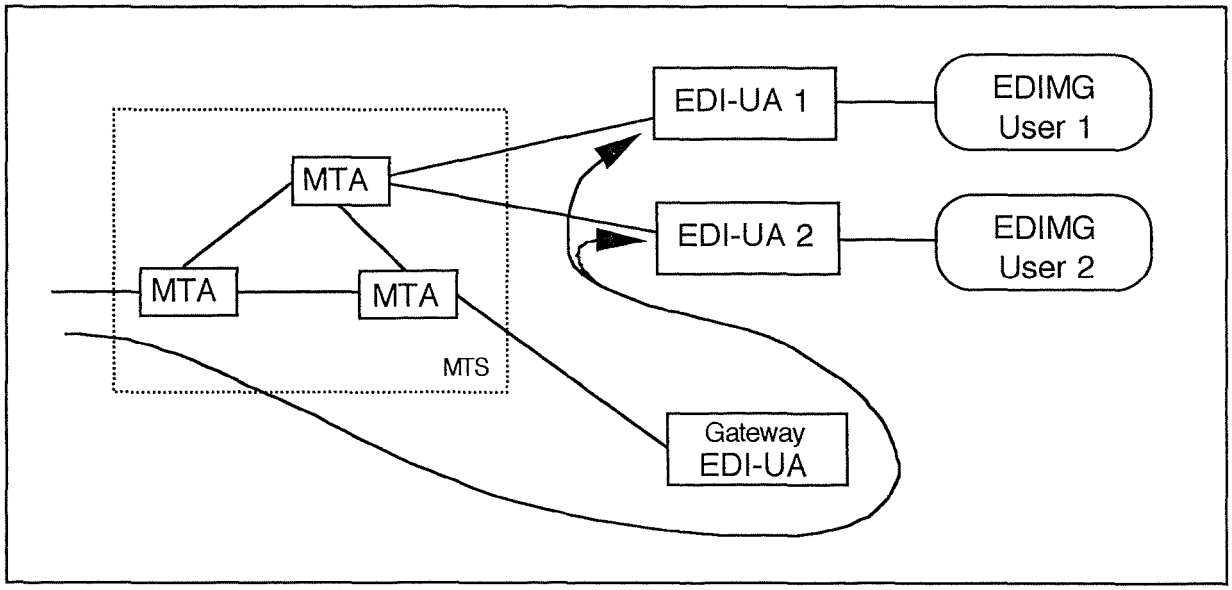

Fig 4.4.3 : EDI-UA is a gateway and does not accept responsibility

The EDI-UA is a gateway and accepts responsibility

This situation is described in figure 4.4.4

In this situation, an EDI-UA once more plays the role of a gateway. This means that all the EDIMs for a company are addressed to the Gateway EDI-UA. Thus only the EDI-UA Gateway is known to the outside world. The EDI-UA 1 is only known by the Gateway. The Gateway is able to refuse or accept the responsibility and creates EDIN (PN or NN following the case), if asked. The Gateway will never send Forwarded Notification. For each EDIM received the Gateway forwards it to its final recipient. This can be done through the MTS using X.435

protocol (EDI-UA 1 and EDIMG User 1) or using a proprietary protocol directly from the Gateway to the EDIMG User (EDIMG User 2). The Gateway can operate changes on the incoming EDIM like dropping and addition of body parts. The aim of this feature is to send specific body parts such as invoices or other body part types to specific EDIMG Users (EDI applications).

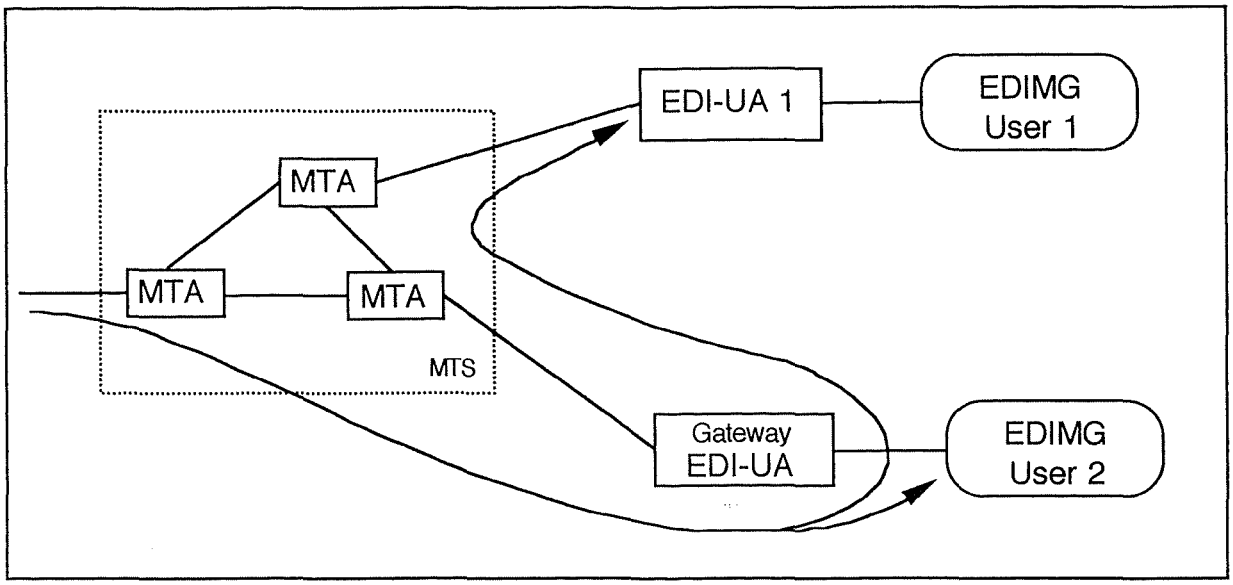

Fig 4.4.4 : EDI-UA is a gateway and accepts responsibility

The EDI-UA is a Value Added Network

This situation is described in figure 4.4.5.

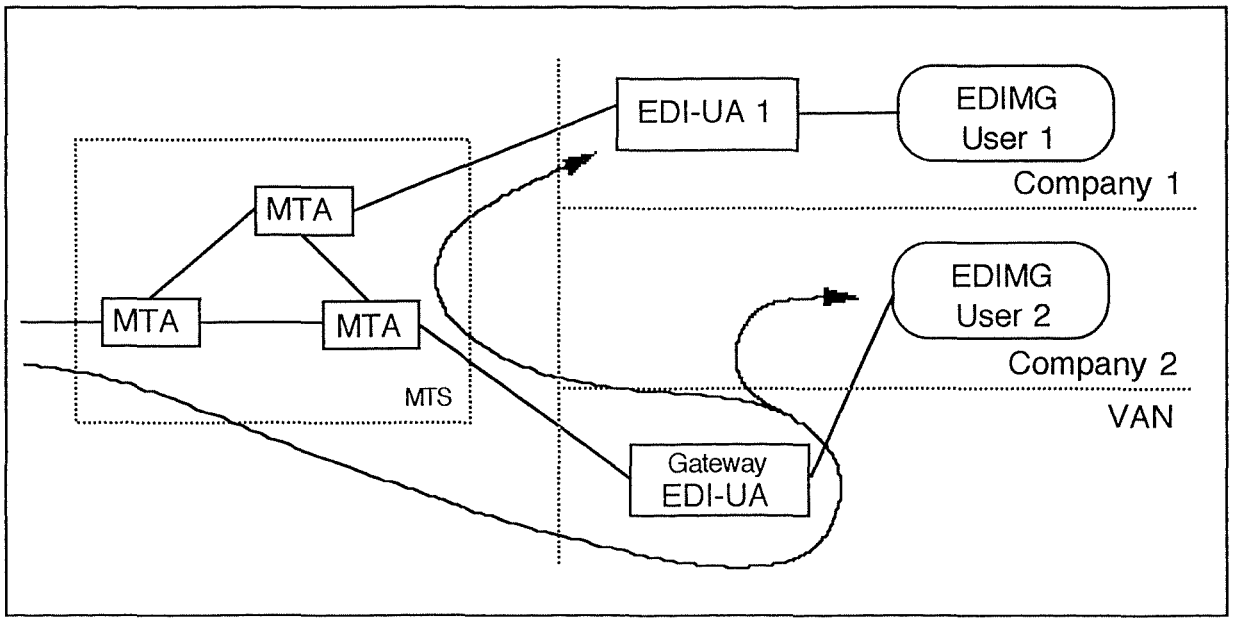

Fig 4.4.5 : EDI-UA is a Value Added Network

In this situation, a private or public company provides Value Added Network (VAN) services. The EDI-UA is a gateway and is part of the VAN provider implementation. The Gateway can accept responsibility or not accept it, the Gateway can send acknowledgements or not depending on contractual agreement between the VAN provider and the user. For each EDIM received the Gateway forwards it to its final recipient. This can be done through the MTS using X.435 Protocol (company 1, EDI-UA 1 and EDIMG User 1) or using a proprietary protocol directly from the Gateway to the EDIMG User (company 2, EDIMG User 2).

The EDI-UA is a company gateway and forwards some EDIM.

This situation is described in figure 4.4.6.

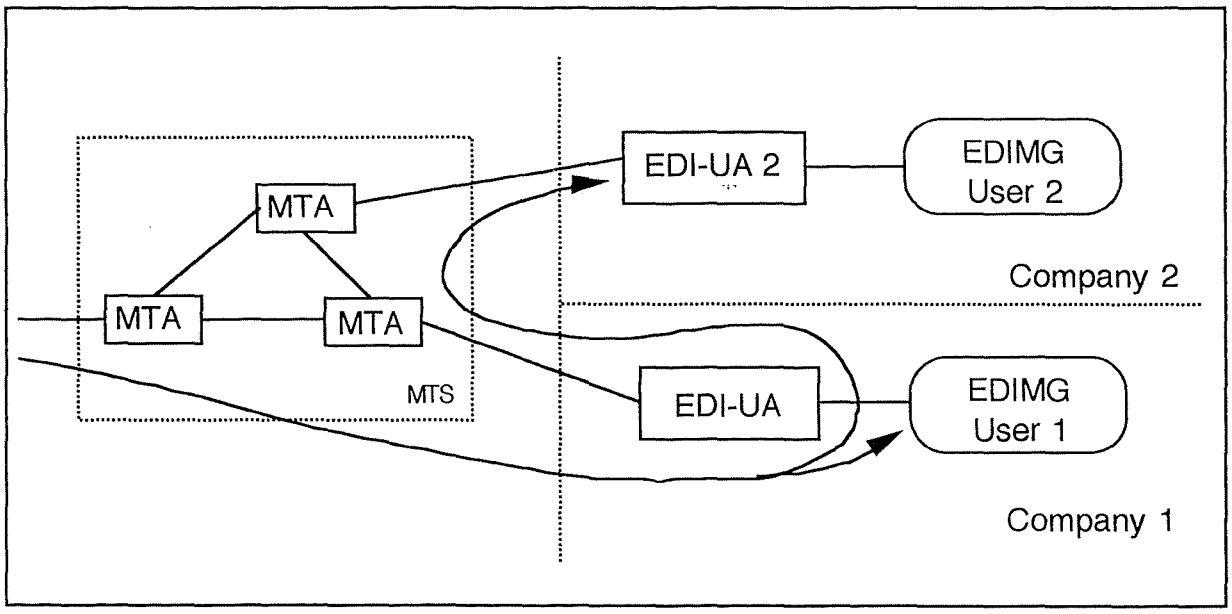

Fig 4.4.6 : EDI-UA is used by a company which does not process all EDIM types

In this scenario, the EDI-UA is used by a company which can process determined EDIM types (such as order) with EDIMG User 1, but delegates others (such as invoice) to an outside company (company 2). So the EDI-UA accepts responsibility for certain types of EDIM (EDIMG User 1) and in other cases, forwards responsibility to another EDIMG User (EDIMG User 2). The EDI-UA sends Positive or Negative Notification in the first case and Forwarded Notifications in the second one.

4.4.4 EDI Message Store

The recommendation X.413 defines the Message Store (MS). The MS is an optional component of the MHS as explained in part 2.3.5 of this work. X.435 supports additional features to those specified in X.413. The EDI Message Store (EDI-MS) is a X.413 MS providing additional specific EDI Abstract Operations. They are :

- maintenance of EDI-MS information ;
- forwarding with responsibility not accepted ;
- forwarding with responsibility accepted.

These three features are explained below .

Maintenance of EDI-MS information

When the MTS delivers an EDIM or an EDIN, the Message Store transforms the heading fields of the incoming EDI Message or EDI Notification into MS attributes. This has to be done in order to use these attributes for selective retrieval by the EDI-UA.

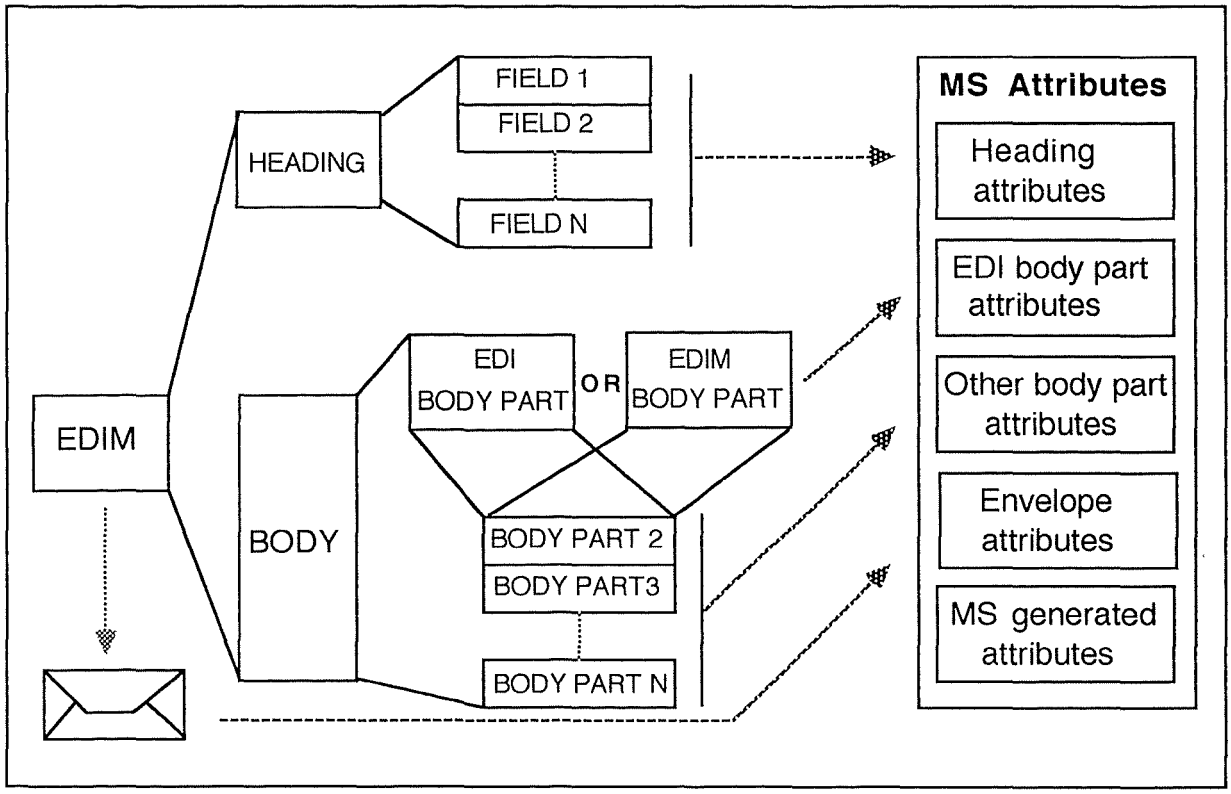

Fig 4.4.7 : Mapping EDIM and MS fields

All the EDIM heading attributes become EDI-MS 'Heading attributes'. The EDI-MS 'EDI body part attributes' contain the whole EDI interchange and signal which EDI standard has been used in the EDIM. Other body parts are mapped into one or several 'Other body part attributes'. The

EDI-MS generates attributes in 'Envelope attributes' based on the MHS envelope elements. 'MS generated attributes' are for administration purposes such as time of entry, sequence number or interchange length. See figure 4.4.7.

Forwarding with responsibility not accepted

X.435 recommendation has defined two forward operations including the concept of responsibility. This section describes the 'forward with responsibility' not accepted and the following part explains the 'forward with responsibility accepted'. For both, the rules for forwarding are the same as for EDI-UAs. Criteria based on EDI-MS attributes must be specified by the EDIMG User for selecting which forward operation has to be performed by the EDI-MS.

The EDI-MS Forwarding with responsibility not accepted is described in figure 4.4.8

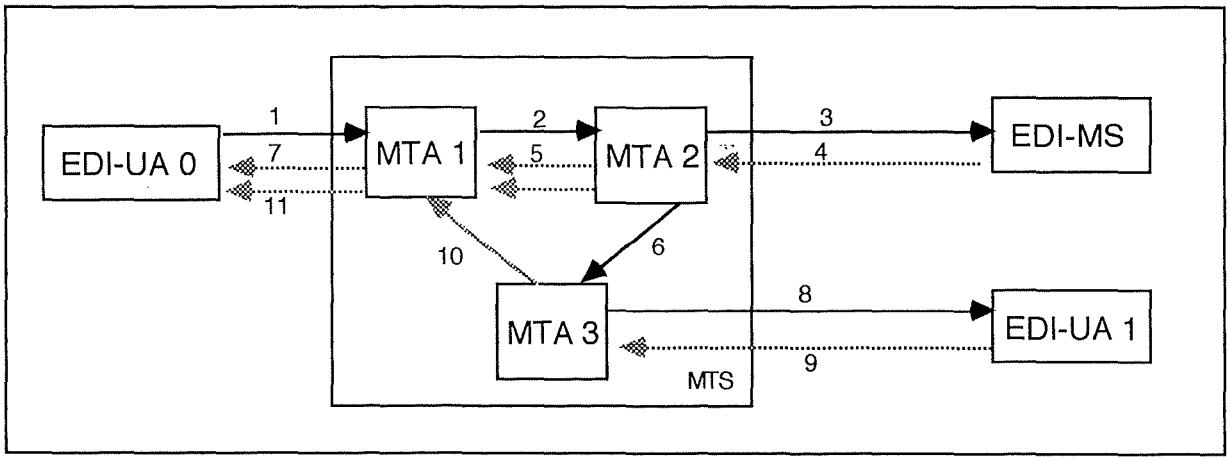

Fig 4.4.8 : EDI-MS Forwarding with responsibility not accepted

When EDIMG User instructs the EDI-MS to perform forwarding with responsibility not accepted, the EDI-MS will always act as follows. As illustrated on the figure 4.4.8 :

1 - EDI-UA 0 submits an EDIM to MTA 1
2 - MTA 1 transfers, through the MTS, the EDIM to MTA 2
3 - MTA 2 delivers the EDIM to EDI-MS
4 - EDI-MS submits the EDIM and if requested by EDI-UA 0, a Forwarded Notification (FN) to MTA 2.
5 - MTA 2 transfers, through the MTS, the EDIN (FN) to MTA 1
6 - MTA 2 transfers, through the MTS, the EDIM to MTA 3
7 - MTA 1 delivers the EDIN (FN) to EDI-UA 0
8 - MTA 3 delivers the EDIM to EDI-UA 1
9 - EDI-UA 1 submits a Positive or Negative Notification (PN or NN) to the MTA 3, if requested by EDI-UA 0.
10 - MTA 3 transfers, through the MTS, the EDIN (PN or NN) to MTA 1.
11 - MTA 1 delivers the EDIN (PN or NN) to EDI-UA 0.

Forwarding with responsibility accepted

The EDI-MS Forwarding with responsibility accepted is described in figure 4.4.9.

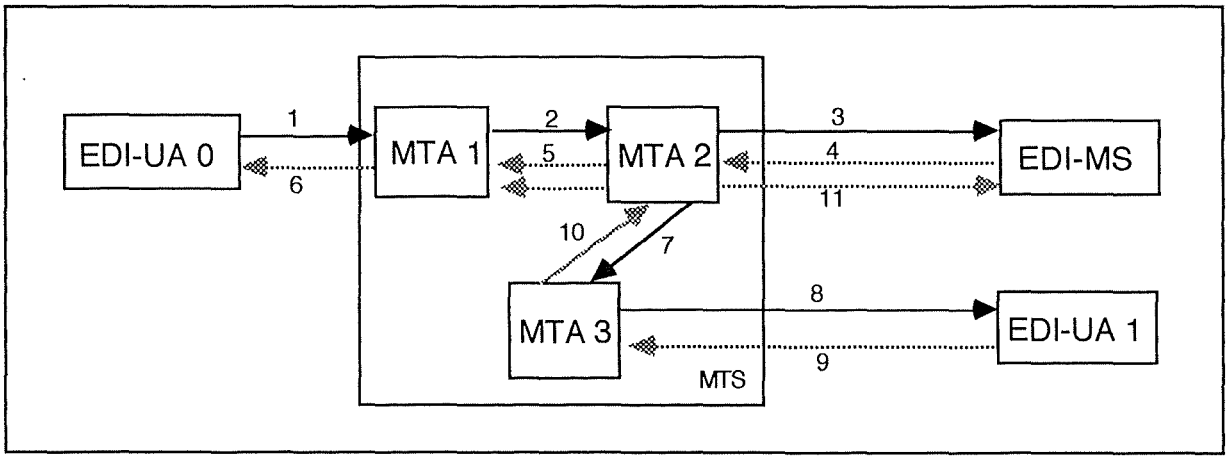

Fig 4.4.9 : EDI-MS Forwarding with responsibility accepted

When EDIMG User instructs the EDI-MS to perform forwarding with responsibility accepted, the EDI-MS will always act as follows. As illustrated on the figure 4.4.10 :

1 - EDI-UA 0 submits an EDIM to MTA 1
2 - MTA 1 transfers, through the MTS, the EDIM to MTA 2
3 - MTA 2 delivers the EDIM to EDI-MS
4 - EDI-MS submits the EDIM and if requested by EDI-UA 0, a Positive or Negative Notification (PN or NN) to MTA 2.
5 - MTA 2 transfers, through the MTS, the EDIN (PN or NN) to MTA 1
6 - MTA 1 delivers the EDIN (PN or NN) to EDI-UA 0
7 - MTA 2 transfers, through the MTS, the EDIM to MTA 3
8 - MTA 3 delivers the EDIM to EDI-UA 1
9 - EDI-UA 1 submits the Positive or Negative Notification (PN or NN) to the MTA 3, if requested by EDI-MS.
10 - MTA 3 transfers, through the MTS, the EDIN (PN or NN) to MTA 2.
11 - MTA 2 delivers the EDIN (PN or NN) to EDI-MS.

Chapter 5 : The use of ASN.1 for EDI

5.1 Introduction

As explained in the previous chapters, X.400 is the OSI solution for transferring EDI Interchange in a store-and-forward manner. X.409 recommendation defines an abstract syntax notation, called ASN.1 in the OSI world. X.409 is used in the X.400 Message Handling System as a notation for describing the X.400 protocols and as encoding rules used for producing a string of octets suitable for transmission.

Since EDI Interchanges are defined in part 1.2.2 as tree structures, there could be two solutions for encoding them. The first one, used by UN/EDIFACT and by other main syntaxes (UN/TDI, ANSI X.12, ...), encodes each EDI Interchange as an octet string. That solution requires delimiters between data. Thus characters like ":+.?" or character strings like "UNB, UNH, NAD, LIN ..." are used for defining what data is related to what information, but contain no "useful" information. The other solution is to code EDI Interchanges using structured languages and encoding schemes using ASN.1. This method is closer to the structured definition of EDI Interchange and does not need character delimiters.

In this chapter, we will first give a brief introduction to ASN.1 notation and encoding rules. The use of a structured language is often considered as an improper solution for EDI in the literature ([TAYL 90] and [O'CON 89]). The ASN.1 detractors arguments are based on a

wrong understanding of its use.This usual method will be firstly explained and we will explain why it is not correct. After that we will define what we consider a correct method. A comparison of encodings of an EDI Interchange using the UN/EDIFACT solution and the two ASN.1 solutions is given. Before a list of advantages and disadvantages of the use of ASN.1 for EDI, we will analyze how ASN.1 can fit X.400 for EDI.

5.2 ASN.1 notation and encoding rules [TANE 89]

5.2.1 Introduction

The Abstract Syntax Notation.1 (ASN.1) is the OSI solution for "representing, encoding, transmitting and decoding data structures" [TANE 89]. ASN.1 is defined in the presentation layer of the OSI model. Even if historically the abstract syntax notation was a part of the X.400 recommendations and thus of the application layer.

The application layer contains many applications which exchange numerous different data structures. Each datum is transmitted from the application with its ASN.1 data structure name to the presentation layer as APDU (Application Protocol Data Unit). Using the ASN.1 definition, the application layer knows the type, the length and can encode the data for transmission. The receiver presentation layer finds out the type and the length of the ASN.1 data structure and thus can decode the data and passes the APDU to the receiving application.

The ASN.1 notation is explained in part 5.2.2 and is described in [X.208 88] and ISO 8824. The ASN.1 encoding rules are described in part 5.2.3 and are given in [X.209 88] and ISO 8825.

5.2.2 Notation

The ASN.1 notation is used for defining data types. The ASN.1 description of data types is called Abstract Notation because no specific representation is affixed.

The general rules of writing use the Backus Naur Form method (BNF). The explanation of this method is beyond the scope of this paper.

The primitive types allow definition of more complex types. They are always written in upper case letters. See example in figure 5.2.1. The primitive types defined by ASN.1 are :

INTEGER	: Arbitrary length integer.
BOOLEAN	: TRUE or FALSE.
BIT STRING	: List of 0 or more bits.
OCTET STRING	: List of 0 or more bytes.
ANY	: Union of all types.
NULL	: No type at all.
OBJECT IDENTIFIER	: Object name (such as a library).

```
number          ::= INTEGER,
unit-price      ::= REAL,
city            ::= OCTET STRING
```

Figure 5.2.1 : Example of primitive types

ASN.1 also defines constructors for building new types with the primitive ones. See example in figure 5.2.2. They are :

SEQUENCE	: Ordered list of various types.
SEQUENCE OF	: Ordered list of a single type.
SET	: Unordered collection of various types.
SET OF	: Unordered collection of a single type.
CHOICE	: Any one type taken from a given list.

```
Purchase_order_item     ::= SEQUENCE {
        number                          INTEGER,
        quantity                        REAL,
        unit-price                      REAL }
```

Figure 5.2.2 : Example of constructors

To allow a situation where several fields are optional, ASN.1 has defined the following types :

OPTIONAL	: Allows optional fields.
DEFAULT	: Gives a default value to a field if it is not specified.

See example in figure 5.2.3.

```
Purchase_order_item     ::= SEQUENCE {
        number                          INTEGER,
        quantity                        REAL,
        unit-price                      REAL,
        pricing-quantity-basis          REAL OPTIONAL }
```

Figure 5.2.3 : Example of Optional

The use of OPTIONAL and DEFAULT can cause problems for identifying to what field a data is linked. To solve this problem ASN.1 has defined the concept of tagging. The tag types are :

UNIVERSAL	: These tags are standardized and are used by various applications.
APPLICATION	: These tags are specific to one application.
PRIVATE	: These tags are used by a restricted number of users, often within an organization or a company.
CONTEXT	: These tags are used in specific context. They allow definitions of sub-elements in a context. The context tags only have a meaning within that context.

The tags are written in upper case letters and between square brackets. See example in figure 5.2.4.

```
*Application tag*
Priority                ::= [APPLICATION 7] INTEGER

*Context tag*
EDI_Interchange         ::= SEQUENCE {
                            [0] Interchange_Heading,
                            [1] Interchange_Body}
```

Figure 5.2.4 : Example of tags

5.2.3 Encoding rules

ASN.1 defines exactly how data has to be encoded. Those rules are called Basic Encoding Rules (BER). The encoding of each value is composed of various fields : type, length , data and optionally end-of-content flag, if the data length is unknown. These fields are explained below. This method of encoding is known under the name TLV (Type, Length and Value).

As explained before, the data structure can be primitive (see figure 5.2.5.a) or can be constructed (see figure 5.2.5.b).

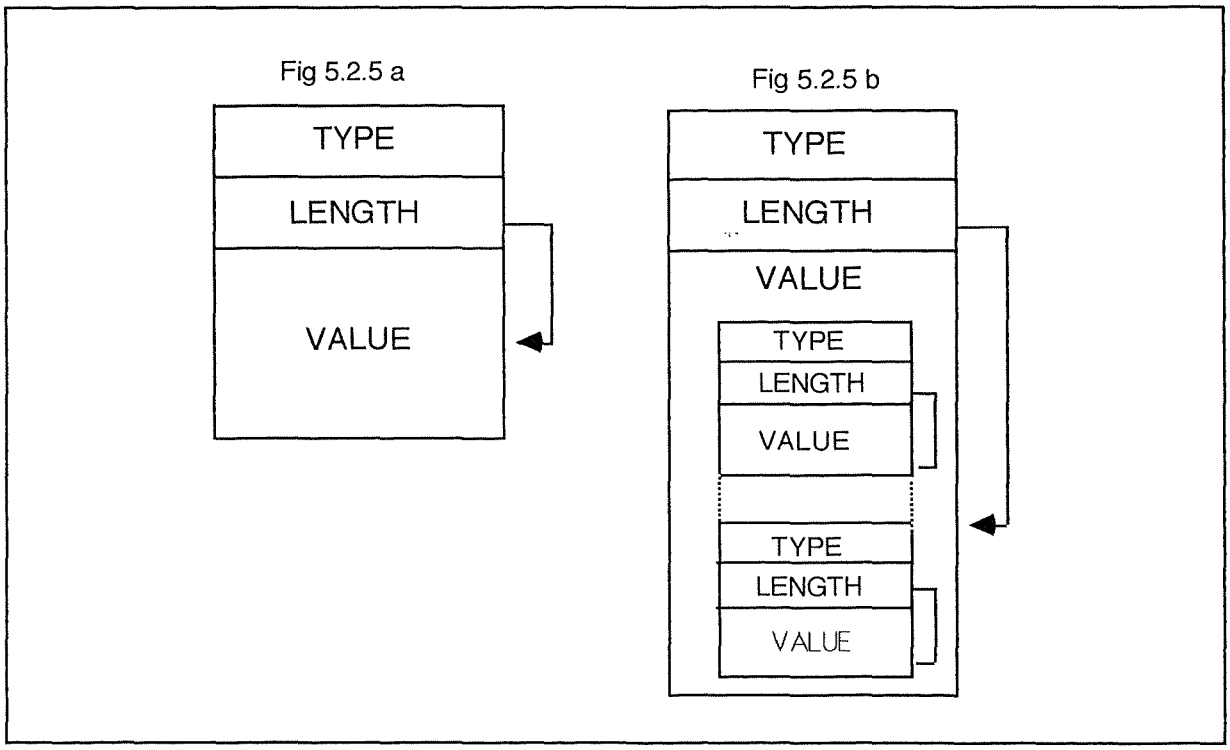

Figure 5.2.5 : ASN.1 encoded data

- The Type (or Tag) :

The type identifies the data structure that follows. It is coded in one byte. The two first bits define the type of the tag. The third bit indicates if the data structure is primitive or constructed. The five last bits contain the identifier tag. The identifier can be written on several bytes but in order to simplify, we will only consider the one-byte solution.

See figure 5.2.6.

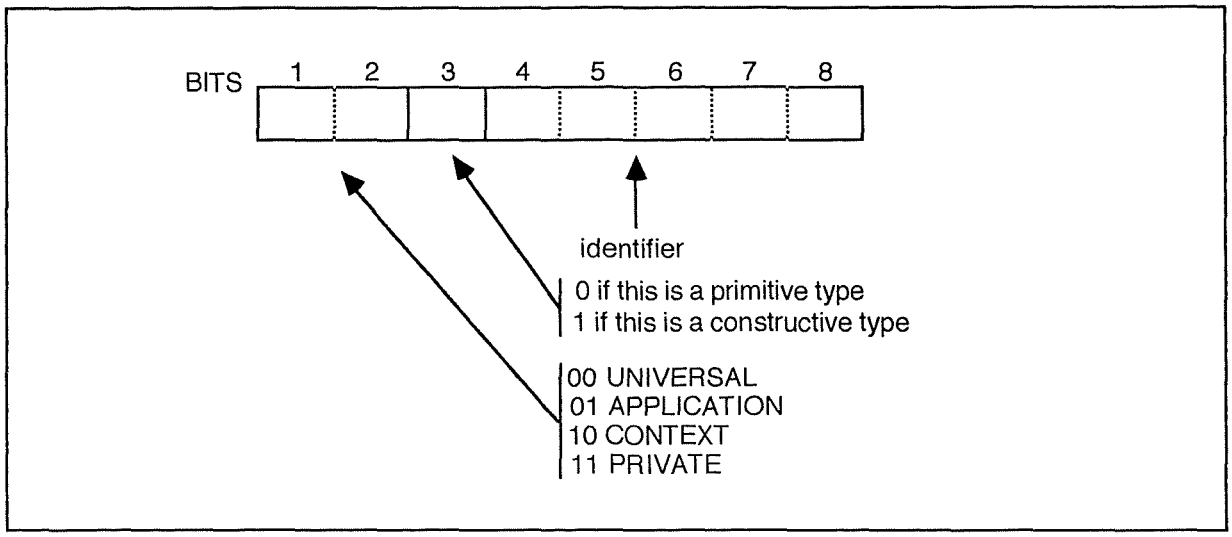

Figure 5.2.6 : ASN.1 tag

- The Length :

The length field determines the length of the value field. The length is encoded on one byte, if the length of the value is longer than 128 bits then the length is encoded on several bytes. If the length is unknown then the field length contains a special code and a end-of-contents flag indicates the end of the value field.

- The Value :

The value field is the value itself. The field type and length indicate how to read this field.

5.3 Definition of EDI Interchanges using ASN.1

The usual approach, relating ASN.1 and EDI, is based on a straight translation of existing EDI standards (such as UN/EDIFACT) into ASN.1. We think that this method is the result of a wrong understanding of structured languages. [TAYL 90] and [O'CON 89] In this part, we will first explain the usual approach and why it is not adequate. In a second section, a proper use of ASN.1 for EDI is shown.

As explained in part 1.2, particularly in figure 1.2.2, an EDI Interchange is defined as a tree structure. In UN/EDIFACT, the root is the interchange, its sheets are the Functional Groups or the messages. The Functional Groups and the messages are non-terminal sheets. For example, the message sheets are the segments. The terminal sheets are the data elements and the coded data elements' codes.

The aim of UN/EDIFACT is to flatten this tree. The result of this operation is a list of characters. And in order to rebuild the initial tree, the recipient needs special marks to find out

to what particular meaning a received data is related. The special segment tags (UNB, UNH, ...), the data segment tags (LIN, NAD,...) , the delimiters ('+:?) are those marks.

The usual method starts from the list of characters. Thus instead of using the tree structure of an EDI Interchange, this approach uses the list of characters which contain all the special segment tags and data segment tags that are not useful for a structured language. In this definition of an EDI message definition in ASN.1, there are characters without any utility.

A correct use of ASN.1 does not require the special segments and does not use the tagging system. Some information held in the UN/EDIFACT UNB and UNH is used in an ASN.1 EDI message definition, but the separation function of the segment tags is useless. The Appendix 8 gives the definition of an ORDER message in ASN.1. Only the first two segments of the message are developed in that definition.

The following example illustrates the UN/EDIFACT, the usual ASN.1 and the correct ASN.1 method for defining an EDI Interchange. The interchange that is defined is fictitious but it has the advantage to be very simple. The goal of this interchange is to send associated names and addresses. Only the mandatory data elements are showed in these definitions. See figure 5.3.1, 5.3.2 and 5.3.3.

TAG	NAME	S	REPT
UNB	Interchange Header	M	1
UNH	Message Header	M	1
NAD	Name and Address	M	5
UNT	Message Trailer	M	1
UNZ	Interchange Trailer	M	1

TAG	DATA ELEMENT		NAME	S	TYPE
UNB	S001		Syntax Identifier	M	
		0001	Syntax Identifier	M	a4
		0002	Syntax Version Number	M	n1
	S002		Interchange Sender	M	
		0004	Sender Identification	M	an..35
	S003		Interchange Recipient	M	
		0010	Recipient Identification	M	an..35
	S004		Date/Time of Preparation	M	
		0017	Date	M	n6
		0019	Time	M	n4
	0020		Interchange Control Reference	M	an..14
UNH	0062		Message Reference Number	M	an..14
	S009		Message Identifier	M	
		0065	Message Type	M	an..6
		0052	Message Version Number	M	n..3
NAD	3035		Party Qualifier	M	an..3
	C058		Name & Address	C	
		3124	Name and Address Line	M	an..35

UNT	0074	Number of Segments in Message	M	n..6
	0062	Message Reference Number	M	an..14
UNZ	0036	Interchange Control Count	M	n..6
	0020	Interchange Control Reference	M	an..14

Figure 5.3.1 : Example of UN/EDIFACT message definition

```
EDI_Interchange                 ::= SEQUENCE {
        interchange-header              [0] Unb,
        messages                        [1] SEQUENCE OF {Message},
        interchange-trailer             [2] Unz}

Message                         ::= SEQUENCE {
        message-header                  [0] Unh,
        message-type                    [1] Message_type,
        message-trailer                 [0] Unt}

Message_type                    ::= CHOICE {
                                        [0] Purchase_order,        * not defined here *
                                        [1] Invoice,               * not defined here *
                                        [2] Name_and_address}

Name_and_address                ::= SEQUENCE {
        segment-tag                     [0] "NAD",
        party-qualifier                 [1] T61String(SIZE(1..3)),
        name-and-address                [2] SEQUENCE (1..5) OF
                                            {Name_and_address_line} OPTIONAL}

Name_and_address_line           ::= T61String(SIZE(1..35))

Unb                             ::= SEQUENCE {
        segment-tag                     [0] "UNB",
        syntax-identifier               [1] SEQUENCE {
                syntax-identifier               T61String(SIZE(1..4)),
                syntax-version-number           INTEGER},
        interchange-sender              [2] SEQUENCE {
                sender-identification           T61String(SIZE(1..35)) },
        interchange-recipient           [3] SEQUENCE {
                recipient-identification        T61String(SIZE(1..35))},
        date-time-of-preparation        [4] SEQUENCE {
                date                            INTEGER,
                time                            INTEGER},
        interchange-control-reference   [5] T61String(SIZE(1..14))  }

Unh                             ::= SEQUENCE {
        segment-tag                     [0] "UNH",
        message-reference-number        [1] T61String(SIZE(1..14)),
        message-identifier              [2] SEQUENCE {
                message-type                    T61String(SIZE(1..6)),
                message-version-number          INTEGER}}
```

```
Unt                                 ::= SEQUENCE {
        segment-tag                         [0] "UNT",
        number-of-segments-in-message       [1] INTEGER,
        message-reference-number            [2] T61String(SIZE(1..14))}

Unz                                 ::= SEQUENCE {
        segment-tag                         [0] "UNZ",
        intechange-control-count            [1] INTEGER,
        interchange-control-reference       [2] T61String(SIZE(1..14))}
```

Figure 5.3.2 : Example of usual ASN.1 message definition

```
EDI_Interchange                     ::= SEQUENCE {
                                            [0] Interchange_heading,
                                            [1] Interchange_body }

Interchange_heading                 ::= SEQUENCE {
        syntax-identifier                   [0] Syntax_identifier
        interchange-sender                  [1] Interchange_sender,
        interchange-recipient               [2] Interchange_recipient,
        date-time-of-preparation            [3] UCTTime,
        interchange-control-reference       [4] T61String(SIZE(1..14)) }

Interchange_body                    ::= SEQUENCE { EDI_Message }

Syntax_identifier                   ::= SEQUENCE {
        syntax-identifier                   [0] T61String(SIZE(1..4)),
        syntax-version-number               [1] INTEGER,

Interchange_sender                  ::= T61String(SIZE(1..35)),

Interchange_recipient               ::= T61String(SIZE(1..35)),

EDI_Message                         ::= CHOICE {
                                            [0] Purchase_order,        * not defined here *
                                            [1] Invoice,               * not defined here *
                                            [2] Name_and_address}

Name_and_address                    ::= SEQUENCE {
                                            [0] Message_heading,
                                            [1] SEQUENCE (1..5) OF
                                                {Parties_name/address } }

Message_heading                     ::= SEQUENCE {
        message-reference-number            [0] T61String(SIZE(1..14)),
        message-type                        [1] T61String(SIZE(1..6)),
        message-version-number              [2] T61String(SIZE(1..3))}
```

```
Parties_name/address          ::= SEQUENCE {
      party-qualifier                 [0] T61String(SIZE(1..3)),
      name-and-address                [1] SEQUENCE (1..5) OF
                                          {Name_and_address_line} OPTIONAL}

Name_and_address_line         ::= T61String(SIZE(1..35))
```

Figure 5.3.3 : Example of correct ASN.1 message definition

5.4. Coding of EDI Interchanges using ASN.1

In order to illustrate those definitions, we will take an example of the previously defined interchange. The UN/EDIFACT coding of the interchange is shown in figure 5.4.1 :

```
UNB+UNOA:1+7265739+526529+910807:1030+12175'UNH+10
505+ORDERS:1'NAD+FUNDP 51 RUE DE BRUXELLES 5000 N
AMUR'UNT+ 1+10505'UNZ+1+12175'
```

Figure 5.4.1 : Example of UN/EDIFACT message coding

As specified below, an UN/EDIFACT interchange can be seen as an octet string or as a list of bytes. Thus each character is an octet. In this example, the UN/EDIFACT interchange is made of 129 octets.

ASN.1 is an abstract syntax notation, so there is no representation defined. The coding is made from a concrete syntax into a flow of bytes. In order to give an exact idea of the number of octets used, the structure of the same example is given below. As explained in point 5.2.3, a datum is coded using the Type, Length and Value method. So for each datum, at least one octet is necessary for coding the Type and an other one for coding the Length. These are shown between brackets in the following. If an element has sheets, the number of bytes used by its sheets is added between the second pair of brackets and its value field contains "V". The number of octets used by the root 'EDI_Interchange' is the number of octets used for coding the interchange.

Figure 5.4.2 is the coding of the example shown in 5.4.1, using the usual ASN.1 definition of the interchange previously described and figure 5.4.3 is the coding of the same example using the correct ASN.1 definition.

	Number of octets	Value
EDI_Interchange	179+(2) (64+98+17)	V
[0] Interchange_header	62+(2) (62)	V
[0] Unb	60+(2) (5+11+11+10+16+7)	V
[0] segment-tag	3+(2)	UNB
[1] syntax-identifier	9+(2) (6+3)	V
[0] syntax-identifier	4+(2)	UNOA
[1] syntax-version-number	1+(2)	1
[2] interchange-sender	9+(2) (9)	V
sender-identification	7+(2)	7265739
[3] interchange-recipient	8+(2) (8)	V
recipient-identification	6+(2)	526529
[4] date-time-of-preparation	14+(2) (8+6)	V
date	6+(2)	910807
time	4+(2)	1030
[5] interchange-control-reference	5+(2)	12175
[1] message	96+(2) (25+51+20)	V
[0] Unh	25+(2) (5+7+13)	V
[0] segment-tag	3+(2)	UNH
[1] message-reference-number	5+(2)	10505
[2] message-identifier	11+(2) (8+3)	V
message-type	6+(2)	NAMADD
message-version-number	1+(2)	1
[1] Message-type	11+(2) (11)	V
[2] Name_and_address	49+(2) (5+4+40)	V
[0] segment-tag	3+(2)	NAD
[1] party-qualifier	2+(2)	BY
[2] name-and-address	38+(2) (38)	V
name-and-address-line	36+(2)	FUNDP 51 RUE DE BRUXELLES 5000 NAMUR
[2] Unt	18+(2) (5+3+7)	V
[0] segment-tag	3+(2)	UNT
[1] number-of-segments-message	1+(2)	1
[2] message-reference-number	5+(2)	10505
[2] Unz	15+(2) (5+3+7)	V
[0] segment-tag	3+(2)	UNZ
[1] interchange-control-count	1+(2)	1
[2] interchange-control-reference	5+(2)	12175

Figure 5.4.2 : Example of usual ASN.1 message coding

	Number of octets	Value
EDI_Interchange	123+(2) (51+72)	V
[0] Interchange_Heading	49+(2) (8+11+10+13+7)	V
[0] syntax-identifier	6+(2)	UNOA 1
[1] interchange-sender	9+(2) (9)	V
Interchange -sender	7+(2)	7265739
[2] interchange-recipient	8+(2) (8)	V
Interchange-recipient	6+(2)	526529
[3] date-time-of-preparation	11+(2)	910807:1030
[4] interchange-control-reference	5+(2)	12175
[1] Interchange_Body	70+(2) (70)	V
EDI_Message	68+(2) (68)	V
[2] Name_and_address	66+(2) (20+46)	V
[0] message_heading	18+(2) (7+8+3)	V
[0] message-reference-number	5+(2)	10505
[1] message-type	6+(2)	NAMADD
[2] message-version-number	1+(2)	1
[1] Parties_name/address	44+(2) (4+40)	V
[0] party-qualifier	2+(2)	BY
[1] Name-and-address-line	38+(2)	V
Name-and-address-line	36+(2)	FUNDP 51 RUE DE BRUXELLES 5000 NAMUR

Figure 5.4.3 : Example of correct ASN.1 message coding

The usual ASN.1 definition needs 181 octets for coding the same example. The correct use of ASN.1 needs 125 octets for coding the same interchange. It is clear that the usual method is not interesting in terms of space needed for coding an interchange. We think that this example is a complementary proof of the poor efficiency of the usual ASN.1 method.

Even if the correct ASN.1 coding is shorter than the UN/EDIFACT coding in number of octets, no firm conclusions can be extracted on this advantage of using ASN.1. In an other example given in Appendix 9, the UN/EDIFACT coding needs 450 octets and the correct ASN.1 coding only needs 386 octets. The ASN.1 solution is a bit more than 14% shorter than the UN/EDIFACT solution.

5.5 Use of ASN.1 for EDI with X.400

Several X.400 solutions carrying EDI character string syntaxes, as UN/EDIFACT, are currently defined : P0/1 US interim solution, P2 European interim solution and X.435 / F.435. These solutions were the subject of the chapters 3 and 4. But in each case, there is a need for

duplicating information contained in the UN/EDIFACT header fields into the X.400 envelope and message headers. The use of ASN.1 for defining structures and encoding the EDI interchange avoids repetition of information. In addition to this advantage, pure EDI information (EDI message type, party-qualifier, date-time of preparation, ...) could be used for selective retrieval from X.400 Message Store.

X.400 already requires the ASN.1 compiler. Thus, in this solution there is more need of a standard formatting software. Using ASN.1, for encoding EDI interchange, gives the opportunity to the receiver to check, using ASN.1 compiler, the syntactical correctness of received interchange and to automatically send an error message if any error occurs.

We think that the use of ASN.1 provides a fully integrated OSI approach for EDI using X.400 as communication protocol.

The Appendix 7 gives a list describing how X.435 recommendation using ASN.1 can suit for EDI requirements for communication.

5.6 Advantages and disadvantages

Advantages

- ASN.1 is precise.
- ASN.1 is widely accepted and known in the OSI world.
- ASN.1 can be used to transmit agreed data types and data field length.
- ASN.1 compiler check syntactical correctness of received EDI Interchange.
- ASN.1 definition structure and encoding is close to EDI interchange structure.
- An EDI interchange using ASN.1 needs less bytes to be coded.
- The concrete representation of a structured language is simple to implement.

Disadvantages

- ASN.1 is too sophisticated than necessary for EDI interchange.
- ASN.1 is not widely known outside the OSI community.
- ASN.1 is not human-readable.
- ASN.1 is error sensitive. If one bit in the type field or in the length field is changed, the rest of the interchange is unreadable.

We think that ASN.1 solution can suit perfectly EDI. Even if the disadvantages are true, it is clear to us that technical aspects of EDI are the scope of technical specialists and not of business men. It is sure that ASN.1 is error sensitive, but the aim of EDI is to provide an error-less system. So if errors occur, we think that the ability to point them out and ask retransmission is more profitable.

Conclusion

We gave first a definition of Data Electronic Interchange and a description of the implementing process for EDI. The implementing process pointed out the importance of consulting user associations before 'go EDI'. The technical needs including EDI standards, translation softwares, hardware and communication media must be fixed during the implementing process. Amongst all the EDI standards, we have briefly described UN/EDIFACT standard because it is supported by the United Nations and it is internationally recognized. We have seen the functionalities provided by the translation softwares. Functionalities which begin by the extraction of data from in-house files, followed by the reformatting stored in a flat-file, by the translation according an EDI standard into an EDI Interchange and is ended by the sending of it. We have described the four possible hardware implementations using personal computers, mini computers and mainframe computers. The last technical need described is the communication media. We have pointed out two main solutions : the use of Value Added Network services and the use of open system solutions. The pro's and the con's of both were given.

Amongst the open system standards, the X.400 recommendations are seen by the EDI communities as the solution for carrying EDI Interchange in a store-and-forward manner. We described the 1984 and 1988 set of recommendations and we gave a list of EDI requirements for a communication medium in order to find out why X.400 suits for EDI. Because we considered of the EDI requirements and the X.400 capabilities as consistent, we gave three solutions for carrying EDI Interchange using the X.400 recommendations.

The first solution, called the P0/1, is defined by ANSI and is for United States. P0/1 simply encapsulates the EDI Interchange inside a MTS envelope providing no specific EDI services. Due to the lack of facilities P0/1 is considered as an interim solution. The second solution, called the P2, is defined by the Commission of the European Communities and is for European countries. P2 uses the interPersonal Messaging System which allows exchanges of message between humans. P2 encapsulates the EDI Interchange inside the body part of an InterPersonal Message. Even if P2 allows the use of the Message Store for selective retrieval, it is considered as interim solution because it is based on the InterPersonal Messaging System. The third solution is called X.435/F.435 recommendation and is a dedicated Messaging System allowing exchanges of EDI messages between computer applications. We described the X.435/F.435 EDI Messaging System and its main features. We think that the use of X.435/F.435 recommendation is adequate for the transfer of EDI Interchange in a store-and-forward manner.

Because X.400 recommendations already use an abstract syntax notation (ASN.1) for describing and coding its parameters, we studied the possible use of this abstract syntax notation for describing and coding EDI Interchange. But ASN.1 is yet another syntax for EDI and UN/EDIFACT is agreed internationally. Thus we think that, in spite of all its advantages, ASN.1 will probably never be widely used.

Even if X.400 recommendations suit for EDI, we must keep in mind that currently only 1984 X.400 becomes available in the world, that 1988 X.400 and X.435 implementations are not available and that X.435 will only take advantages when it will be implemented on 1988 MTAs. The future use of X.435/F.435 recommendation for EDI depends on the world wide acceptance of Open System Interconnection standards. We can hope that OSI standards are constrained by the lack of implementable applications. Perhaps X.435/F.435 will push this expected emergence...

[CEC 88] Commission of the European Communities, *Guidelines for P2 implementation*, Brussels, 1988.

[CROU 90] R. CROUSE, PC, Midrange, or Mainframe : Choosing the Right EDI Platform, *EDI Forum, The Journal of Electronic Data Interchange*, pp. 117-120, EDI Group Ltd, United States, 1990.

[CRYP 90] CRYPTOMATHICA/S, Digital Signatures in EDIFACT, *The TEDIS Programme*, Brussels, November 1990.

[FENT 89] N. FENTON, Atechnical introduction to the UN/TDI standards, *EDI technology*, Blenheim Online, pp. 41-61, London, 1989.

[F.435 90] CCITT, *Message Handling : EDI Messaging Service, Draft Recommendation F.435*, Version 5.0, Lausanne, June 15, 1990.

[GENI 88] G. GENILLOUD, Proposal for Interim Guidelines for the Use of X.400 for EDI, *Interim Rapporteurs Group on EDI/X.400*, Burlingame Meeting, 7 - 11 November 1988.

[GOSIP 90] Governement OSI Profile, *Electronic Data Interchange (EDI)*, Draft for comment, London, November, 1990.

[GUIL 87] D. GUILBAULT, *X.400 Communication Guideline for EDI*, ANSI/ASC/X12C/TG5/87-052, Canada, 1987.

[HASE 89] M. HASENBERG,Standardisation Issues, *Tedis Workshop, EDI and Telecommunications*, Brussels, June,1989.

[HILL 90] R. HILL, *EDI and X.400 using Pedi*, Technology Appraisals, Islesworth, 1990.

[HILL 91] R. HILL, Use of ASN.1 for EDI, *Discussion Paper, First Draft*, Switzerland, January 8, 1991.

[ISO 8571] International Organization for Standardization, *Information processing systems - open Systems Intercommunication - File Transfer, Access and Management*, Geneva, 1988.

[LI 90] M. LI, EDI addressing and directories, *Proceedings of the UK National Conference on Paperless Trade, EDI 90*, Bleenheim Queensdale, pp. 573-580, London,1990.

[MACP 87] S. MACPHERSON, *What are VADS ? A Guide to Value Added and Data Services*, National Computing Centre Limited Publications, England, 1987.

[MANR 89] C.U. MANROS, *The X.400 Blue Book Companion*, Technology Appraisals, Islesworth, 1989.

[MOO 88] P. E. MOO, *X12-ISO 9735 Syntax Comparison and Recommendations*, ASC X 12C / TG6/88 - 101, United States, May, 1988.

[NUSS 91 III] H. NUSSBAUMER, *Téléinformatique III, Session, Presentation, Compression des données, couche d'application*, pp. 461-469, Collection informatique, Presses Polytechnique et Universitaires Romandes, Suisse, 1991.

[NUSS 91 IV] H. NUSSBAUMER, *Téléinformatique IV, Mesagerie électonique X.400, Messagerie industrielle MMS, Transfert de fichiers FTAM*, pp. 32-34, Collection informatique, Presses Polytechnique et Universitaires Romandes, Suisse, 1991.

[O' CON 89] R. M. O' CONNOR, Use of ASN.1/BER instead of EDIFACT Syntax, *Edifice - Communication Task Group*, Switzerland, May 1989.

[ROSE 90] M.T. ROSE, *The Open Book : A Practical Perspective on OSI*, Prentice-Hall International, Inc., Englewood Cliffs, 1990.

[SIMO] S. SIMONET, *X.400 technical course*, InformaBel "Advanced Technics" Departement, Belgium.

[SITP 90] SITPRO, *The EDIFACT Service from SITPRO*, London, 1990.

[TANE 89] ANDREW S. TANENBAUM, *Computer Networks*, Prentice-Hall International, Inc., Englewood Cliffs, 1989.

[TAYL 90] G. J. TAYLOR, A. E. HOWARD, *A study to draw a correlation between EDIFACT and ASN. 1*, Department of Trade and Industry, Information Technology Standards Unit, London, July, 1990.

[TOYE 89] P. TOYE, Review of OSI standards, *EDI technology*, Blenheim Online, pp. 87-107, London, 1989.

[VANG 88] VANGUARD, *EDI and X.400 Study*, Department of Trade and Industry, Her Majesty's Stationery office, London, 1988.

[VANG 89 I] VANGUARD, *EDI Standards : a guide for Existing and Prospective Users*, Department of Trade and Industry, Her Majesty's Stationery office, London, 1989.

[VANG 89 II] Vanguard, *Value-Added and Data Services Interworking Study*, Department of Trade and Industry, Her Majesty's Stationery Office, London, 1989.

[WARD 89] K. WARD, EDI and Translation Software Products, *EDI Forum, The Journal of Electronic Data Interchange*, pp. 138-142, EDI Publications Inc, United States, 1989.

[X.208 88] CCITT, *Specification of Abstract Syntax Notation One (ASN.1), Recommendation X.208*, CCITT, Geneva, March 1988.

[X.209 88] CCITT, *Specification of Basic Encoding Rules for Abstract Syntax Notation One (ASN.1), Recommendation X.209*, CCITT, Geneva, March 1988.

[X.400 84] CCITT, *Data Communication Network, Message Handling System, Recommendations X.400-X.430*, CCITT, Geneva, 1984.

[X.400 88] CCITT, *Data Communication Network, Message Handling System, Recommendations X.400-X.420*, CCITT, Geneva, 1988.

[X.435 90] CCITT, *Message Handling Systems : EDI Messaging System, Draft Recommendation X.435*, Version 5.0, Lausanne, June 15, 1990.

[X.500 88] CCITT, *The Directory - Overview of Concept, Models and Sevices, Recommendation X.500*, CCITT, Geneva, December 1988.

ACSE	Association Control Service Element
AE	Application Entity
ADMD	Administration Management Domain
AM	Association Manager
ANSI	American National Standards Institution
AP	Application Process
APDU	Application Protocol Data Unit
ASE	Application Service Element
ASN.1	Abstract Syntax Notation one
BER	Basic Encoding Rules
BNF	Backus Naur Form
CASE	Common Application Service Element
CCITT	Comité Consultatif International Télégraphique et Téléphonique
DAP	Directory Access Protocol
DDA	Domain Defined Attributes
DIB	Directory Information Base
DSA	Directory Service Agent
DSP	Directory System Protocol
DUA	Directory User Agent
DUNS	Dun and Bradstreet
EAN	International Article Number Association
EDCD	EDIFACT Composite Data Elements Directory
EDCL	EDIFACT Code List
EDED	EDIFACT Data Element Directory
EDI	Electronic Data Interchange
EDI-AU	Electronic Data Interchange Access Unit
EDIFACT	See UN/EDIFACT
EDIM	Electronic Data Interchange Message
EDIME	Electronic Data Interchange Messaging Environment
EDIMG	Electronic Data Interchange Messaging
EDIMG User	Electronic Data Interchange Messaging User
EDIMS	Electronic Data Interchange Messaging System
EDI-MS	Electronic Data Interchange Message Store
EDI-UA	Electronic Data Interchange User Agent
EDIN	Electronic Data Interchange Notification
EDMD	EDIFACT Message Directory
EDSD	EDIFACT Standard Segments Directory
FN	Forwarded Notification
FTAM	File Transfer, Access and Management
GOSIP	Government OSI Profile
IA	Interchange Agreement
IPM	InterPersonal Messaging
IPMS	InterPersonal Messaging System
IPM-UA	InterPersonal Messaging User Agent
IPM-UAE	InterPersonal Messaging User Agent Entity
ISDN	Integrated Services Digital Network
ISO	International Standards Organization
LAN	Local Area Network
MASE	Message Administration Service Element
MD	Message Dispatcher
MDSE	Message Delivery Service Element
MHS	Message Handling System
MOTIS	Message-Oriented Text Interchange Systems
MRSE	Message Retrieval Service Element
MS	Message Store
MSSE	Message Submission Service Element
MTA	Message Transfer Agent
MTAE	Message Transfer Agent Entity

MTL	Message Transfer Sublayer
MTS	Message Transfer System
MTSE	Message Transfer System Element
NN	Negative Notification
O/R	Originator/Recipient
OSI	Open System Interconnection
PAD	Packet Assembler/Disassembler
PC	Protocol Content
PDAU	Physical Delivery Access Unit
Pedi	Protocol for Electronic Data Interchange
PN	Positive Notification
PRMD	Private Management Domain
P-SAP	Presentation Service Access Point
PSDN	Public Switched Data Network
PSTN	Public Switched Telephone Network
PTT	Postal, Telegraph and Telephone Administration
ROSE	Remote Operation Server
RTS	Reliable Transfer Server
SASE	Specific Application Service Element
SDE	Submission and Delivery Entity
TLV	Type, Length and Value
UA	User Agent
UAE	User Agent Entity
UAL	User Agent Sublayer
UN/EDIFACT	United Nations/Electronic Data Interchange For Administration, Commerce and Transport
UNSM	United Nations Standard Message
UNTDED	United Nations Trade Data Elements Directory
UN/TDI	United Nations/Trade Data Interchange
VADS	Value Added Data Services
VAN	Value-Added Network

INTERFACTS - Produced 90/02/28

UN/EDIFACT Data Elements Directory 90.1 Date Created: 90/02/15

3164 CITY NAME

Function : Name of a city (a town, a village) for addressing purposes

Format : an..35

Data Element 3164 is used in Segment : NAD 88.1

INTERFACTS - Produced 90/02/28

UN/EDIFACT Data Elements Directory 90.1 Date Created : 90/02/15

3207 COUNTRY, CODED

Function : Name of a country or other geographical entity as specified in ISO 3166.

Format : a2

Note : ISO 2-alpha Country code, ISO 3166.

Data Element 3207 is used in Segment: NAD 88.1

INTERFACTS - Produced 90/02/28

UN/EDIFACT Code Lists Directory 90.1 Date Created : 90/02/15

3207 COUNTRY CODE

Description : ISO 3166 two alpha country code

AD	Andorra
AE	United Arab Emirates
AF	Afghanistan
AG	Antigua and Barbuda
AI	Anguila
AL	Albania
AN	Netherlands Antilles
AO	Angola
AQ	Antarctica
AR	Argentina
AS	American Samoa
AT	Austria
AU	Australia
AW	Aruba
BB	Barbados
BD	Bangladesh
BE	Belgium
BF	Burkina Faso
BG	Bulgaria
BH	Bahrain
BI	Burundi
BJ	Benin
BM	Bermuda
BN	Brunei Darussalam
BO	Bolivia
BR	Brazil

BS	Bahamas
BT	Bhutan
BU	Burma
BV	Bouvet Island
BW	Botswana
BY	Byelorussian SSR
BZ	Belize
CA	Canada
CC	Cocos Islands
CF	Central African Republic
CG	Congo
CH	Switzerland
CI	Cote d'Ivoire
CK	Cook Islands
CL	Chile
CM	Cameroon
CN	China
CO	Colombia
CR	Costa Rica
CS	Czechoslovakia
CU	Cuba
CV	Cape Verde
CX	Christmas Island
CY	Cyprus
DD	(German Democratic Republic)
DE	Germany, Federal Republic of
DJ	Djibouti
DK	Denmark
DM	Dominica
DO	Dominican Republic
DZ	Algeria
EC	Ecuador
EG	Egypt
EH	Western Sahara
ES	Spain
ET	Ethiopia
FI	Finland
FJ	Fiji
FK	Falkland Islands (Malvinas)
FM	Micronesia
FO	Faroe Islands
FR	France
GA	Gabun
GB	United Kingdom
GD	Grenada
GF	French Guiana
GH	Ghana
GI	Gibraltar
GL	Greenland
GM	Gambia
GN	Guinea
GP	Guadeloupe
GQ	Equatorial Guinea
GR	Greece
GT	Guatemala
GU	Guam
GW	Guinea-Bissau
GY	Guyana

HK	Hong Kong
HM	Heard and McDonald Islands
HN	Honduras
HT	Haiti
HU	Hungary
ID	Indonesia
IE	Ireland
IL	Israel
IN	India
IO	British Indian Ocean Territory
IQ	Iraq
IR	Iran (Islamic Republic of)
IS	Iceland
IT	Italy
JM	Jamaica
JO	Jordan
JP	Japan
KE	Kenya
KH	Kampuchea, Democratic
KI	Kiribati
KM	Comoros
KN	Saint Kitts and Nevis
KP	Korea, Dem. People's Rep. of
KR	Korea, Republic of
KW	Kuwait
KY	Cayman Islands
LA	Lao People's Democratic Republic
LB	Lebanon
LC	Saint Lucia
LI	Liechtenstein
LK	Sri Lanka
LR	Liberia
LS	Lesotho
LU	Luxembourg
LY	Libyan Arab Jamahiriya
MA	Morocco
MC	Monaco
MG	Madagascar
MH	Marshall Islands
ML	Mali
MN	Mongolia
MO	Macau
MP	Northern Mariana Islands
MQ	Martinique
MR	Mauritania
MS	Montserrat
MT	Malta
MU	Mauritius
MV	Maldives
MW	Malawi
MX	Mexico
MY	Malaysia
MZ	Mozambique
NA	Namibia
NC	New Caledonia
NE	Niger
NF	Norfolk Island
NG	Nigeria

NI	Nicaragua
NL	Netherlands
NO	Norway
NP	Nepal
NR	Nauru
NT	Neutral Zone
NU	Niue
NZ	New Zealand
OM	Oman
PA	Panama
PE	Peru
PF	French Polynesia
PG	Papua New Guinea
PH	Philippines
PK	Pakistan
PL	Poland
PM	St. Pierre et Miquelon
PN	Pitcairn
PR	Puerto Rico
PT	Portugal
PW	Palau
PY	Paraguay
QA	Qatar
RE	Reunion
RO	Romania
RW	Rwanda
SA	Saudi Arabia
SB	Solomon Islands
SC	Seychelles
SD	Sudan
SE	Sweden
SG	Singapore
SH	St. Helena
SJ	Svalbard and Jan Mayen Islands
SL	Sierra Leone
SM	San Marino
SN	Senegal
SO	Somalia
SR	Suriname
ST	Sao Tome and Principe
SU	USSR
SV	El Salvador
SY	Syrian Arab Republic
SZ	Swaziland
TC	Turks and Caicos Islands
TD	Chad
TF	French Southern Territories
TG	Togo
TH	Thailand
TK	Tokelau
TN	Tunisia
TO	Tonga
TP	East Timor
TR	Turkey
TT	Trinidad and Tobago
TV	Tuvalu
TW	Taiwan Province of China
TZ	Tanzania

UA	Ukrainian SSR
UG	Uganda
UM	United States Minor Out.Isl.
US	United States
UY	Uruguay
VA	Vatican City State
VC	Saint Vincent and the Grenadines
VE	Venezuela
VG	British Virgin Islands
VI	Virgin Islands, U.S.
VN	Viet Nam
VU	Vanuatu
WF	Wallis and Futuna Islands
WS	Samoa
YD	Democratic Yemen
YE	Yemen
YU	Yugoslavia
ZA	South Africa
ZM	Zambia
ZR	Zaire
ZW	Zimbabwe

INTERFACTS - Produced 90/02/27

UN/EDIFACT Composite Data Elements Directory 90.1 Date Created : 90/02/15

C186 QUANTITY INFORMATION

Function : Quantity information in a transaction, qualified when relelvant

6063	Quantity qualifier	C an..3
6060	Quantity	M n..15
6411	Measure unit specifier	C an..3

Composite data element C186 is used in Segment : QTY 90.1
QVA 88.1
SID (88.1) 90.1
LIN (88.1) 90.1

6063 QUANTITY QUALIFIER

Function : Code giving specific meaning to a quantity

Format : an..3

Data Element 6063 is used in Composite : C186 88.1

6060 QUANTITY

Function : Numeric value of a quantity

Format : n..15

Data Element 6060 is used in Composite : C186 88.1
C264 88.1
C280 88.1

6411 MEASURE UNIT SPECIFIER

Function : Indication of the unit of measurement in which weight (mass), capacity, length, area, volume or other quantity is expressed

Format : an..3

Note : Examples: kilogram, litre, metre, square metre, cubic metre, number CIMP codes : Kilogram = K ; Pound = LISO 1000, ISO 2955, ISO 4217, cf. 5.6. CIMP codes cf 5.9

Data Element 6411 is used in Composite : C118 88.1
C128 88.1
C134 (88.1) 90.1
C174 88.1
C178 88.1
C186 88.1
C272 88.1
C270 88.1
C280 88.1

INTERFACTS - Produced 90/02/27

UN/EDIFACT Data Segments Directory 90.1 Date Created : 90/02/15

NAD NAME AND ADDRESS

Function : To specify the name/address and their related function, either by C082 only and/or unstructured by C058 or structured by C080 thru 3207

3035 PARTY QUALIFIER	M an..3
C082 PARTY IDENTIFICATION	C
3039 Party identification, coded	M an..17
1131 Code list identifier, coded	C an2
C058 NAME & ADDRESS	C
3124 Name and address line	M an..35
3124 Name and address line	C an..35
3124 Name and address line	C an..35
3124 Name and address line	C an..35
3124 Name and address line	C an..35
C080 PARTY NAME	C
3036 Party name	M an..35
3036 Party name	C an..35
3036 Party name	C an..35
C059 STREET	C
3042 Street and number/P.O.Box	M an..35
3042 Street and number/P.O.Box	C an..35
3042 Street and number/P.O.Box	C an..35
3164 CITY NAME	C an..35
3229 COUNTRY SUB-ENTITY, CODED	C an..9
3251 POST CODE	C an..9
3207 COUNTRY, CODED	C a2

INTERFACTS - Produced 90/02/27

UN/EDIFACT Data Segments Directory 90.1 Date Created : 90/02/15

LIN LINE ITEM

Function : To specify the basic and most frequently used line item data for a transaction

1082 LINE ITEM NUMBER	C n..6
1229 ACTION REQUEST CODE	C n..2
C198 PRODUCT IDENTIFICATION	C
7020 Article number	M an..35
7023 Article number identifier	M an..3
C198 PRODUCT IDENTIFICATION	C
7020 Article number	M an..35
7023 Article number identifier	M an..3
C186 QUANTITY INFORMATION	C
6063 Quantity qualifier	C an..3
6060 Quantity	M n..15
6411 Measure unit specifier	C an..3
C118 UNIT PRICE INFORMATION	C
5110 Unit price	M n..15
5375 Price type, coded	C an..2
5284 Unit price basis / rate basis	C n..9
6411 Measure unit specifier	C an..3
6170 NUMBER OF PRICING UNITS	C n..9
5116 ITEM AMOUNT	C n..15
C134 REFERENCE PRICE INFORMATION	C
5110 Unit price	M n..15
5375 Price type, coded	C an..2
5387 Price type qualifier	C an..3
5284 Unit price basis / rate basis	C n..9
6411 Measure unit specifier	C an..3
6318 QUALITY/YIELD PERCENTAGE	C n..7

UNITED NATIONS STANDARD MESSAGE (UNSM)

PURCHASE ORDER

Message Type : ORDER
Version : 90
Release : 1
Contr. Agency : UN
Status : 2
Date : 90-03

Joint Submission by UN EDIFACT Rapporteur Groups

CONTENTS

0. INTRODUCTION

This specification provides the definition of the Purchase order message (ORDER) to be used in Electronic Data Interchange (EDI) between partners involved in administration, commerce and transport.

1. SCOPE

1.1 Functional Definition

A message specifying details for goods or services ordered under conditions agreed between the seller and the buyer.

1.2 Field of Application

This message may be applied for both national and international trade. It is based on universal practice and is not dependent on the type of business or industry.

1.3 Principles

. A buyer may order for one or more goods items or services.

. A purchase order may refer to goods items or services related to one or more delivery schedules, call-offs, etc.

. A purchase order for crossborder transactions may contain additional information for customs and/or statistical purposes.

. A purchase order may contain details for transport and destination as well as delivery patterns.

2. MESSAGE DEFINITION

2.1 Data Segment Clarification

This chapter should be read in conjunction with the Branching Diagram (not in this text) and the Segment Table which indicate mandatory, conditional and repeating requirements.

2.1.1 Heading Section

Information to be provided in the Heading Section:

UNH, Message Header

A service segment starting and uniquely identifying a message. The message type code for the UN Purchase Order Message is 'ORDER'.

BGM, Beginning of Message
A segment for unique identification of the Purchase Order Document Number, date, and type of order, and optionally a primary reference to another document, e.g. quotation or contract.

RFF, References
A segment for referencing documents relating to the whole message, e.g. contract, import/export license.

CTA, Contact
A segment for identifying contacts relevant to the whole message.

Segment group 1: NAD-LOC-RFF-DOC-CTA-FII
A group of segments identifying names, addresses andlocations, contacts and references, and required supporting documents relevant to the whole purchase order, as well as providing financial institution information for a party.

NAD, Name and address
A segment for identifying names and addresses and their functions relevant for the whole purchase order. Identification of the seller and buyer parties is mandatory for the purchase order message. They are to be given in the NAD segment.

LOC, Location identification
A segment indicating more details regarding specific locations related to the party specified in the NAD segment.

RFF, References
A segment for referencing document and other numbers related to the party and its function as specified in the NAD segment.

DOC, Documents required
A segment for providing information relating to the documents required by the party specified by the NAD segment.

CTA, Contacts
A segment giving additional contact information relating to the party specified in the NAD segment, e.g. contact person or department in a particular function.

FII, Financial institution information
A segment identifying the financial institution (e.g. bank) and relevant account numbers for the seller, buyer and other parties. e.g. the seller may provide a choice of financial institutions for payment.

DTM, Date/time reference
A segment specifying the date, and when relevant, the time of an activity identified by the date/time qualifier, e.g. shipped on date.

Segment group 2 : TRI-LOC
A group of segments specifying tax related information, and when necessary, the location(s) to which that tax information relates.

TRI, Tax related information
A segment specifying a tax type, category and rate, or exemption, relating to the whole purchase order.

LOC, Location identification
A segment indicating the location to which the tax or exemption specified in the TRI segment applies, e.g. city or state tax.

Segment group 3 : CUX-DTM
A group of segments specifying the currencies and related dates/periods valid for the whole purchase order.

CUX, Currencies
A segment identifying the currencies required in the purchase order, e.g. the order currency and other currencies required in the transaction. A rate of exchange may be given to convert the reference currency into the target currency. Currency data is usually omitted in national applications but will be required for international transactions.

DTM, Date/Time Reference
A segment specifying the date/period and, where relevant, time details related to the currency information.

ALI, Additional line information
A segment indicating special conditions related to the total purchase order owing to origin, customs preference, or other commercial factors.

FTX, Free Text
A segment with free text information, in coded or clear form, to give further clarification, when required, to the whole purchase order. In computer to computer exchanges such text will normally require the receiver to process this segment manually.

PAT, Payment Terms Basis
A segment indicating the payment terms, date/time basis, and additional terms valid for the purchase order.

PAI, Payment Instructions
A segment specifying conditions of payment, guarantee and method of payment for the whole purchase order.

Segment groups 4 + 5 : TDT-LOC-DTM
Groups of segments specifying details of the mode of transport, location, and date/time of departure and destination relating to the whole purchase order.

TDT, Details of Transport
A segment specifying the carriage, and the mode and means of transport of the goods being ordered.

LOC, Location Identification
A segment indicating locations relevant to the transport specified in the TDT segment.

DTM, Date/Time Reference
A segment indicating the date/time details of departure and/or arrival relating to the TDT segment.

TOD, Terms of Delivery
A segment indicating the terms of delivery and transfer for the whole purchase order.

Segment group 6 : PAC-MEA-PCI
A group of segments identifying the packaging, physical dimensions, and marks and numbers for goods referenced by the whole purchase order.

PAC, Package
A segment specifying the number of units per package and the physical type of packaging for the whole invoice, e.g. pallet size.

MEA, Measurements
A segment specifying physical measurements of packages described in the PAC segment.

PCI, Package identification
A segment specifying markings and labels used on individual physical units (packages) described in the PAC segment.

Segment Group 7 : EQF-EQA-HAN-FTX
A group of segments providing information relating to equipment identification and ownership, handling and notification of hazardous materials, relating to the whole purchase order.

EQF, Equipment Fixed Information
A segment to define fixed information regarding equipment to be used in conjunction with the whole purchase order, and if required, to indicate responsibility for supply of the equipment, e.g. a container identification.

EQA, Equipment Attached
A segment identifying equipment either attached to the equipment described in the EQF segment above, or equipment related to that EQF segment, and which is further defined in a subsequent EQF segment.

HAN, Handling Instruction
A segment providing information on required handling of materials in the whole purchase order, and additionally, if required, notifying hazardous materials in the whole purchase order.

FTX, Free Text
A segment with free text information, in coded or clear form, to give further clarification, when required, to the equipment and handling details. In computer to computer exchanges such text will normally require the receiver to process this segment manually.

Segment Groups 8 + 9 : SCC-QTY-DTM-FTX
A group of segments specifying requested delivery schedules elating to quantities, frequencies, and dates, required for he whole purchase order

SCC, Scheduling Conditions
A segment specifying the type and status of the schedule being given, and optionally defining a pattern to be established, e.g. firm or proposed delivery schedule for a weekly pattern.

QTY, Quantity
A segment to specify pertinent quantities relating to the schedule(s) and pattern established in the SCC segment, e.g. delivery quantity.

DTM, Date/Time Reference
A segment indicating the date/time details relating to the quantity and schedule details in the SCC/QTY segments. This segment may be repeated to indicate date/time ranges, e.g. start and end dates for a delivery pattern, or delivery window.

FTX, Free Text
A segment with free text information, in coded or clear form, to give further clarification to scheduling instructions. In computer to computer exchanges such text will normally require the receiver to process this segment manually.

API, Additional Price Information
A segment providing information concerning pricing related to type of trade, quantity, date/period, or amount.

LTM, Lead Time
A segment for providing information concerning lead times valid for the whole order with regard to the availability of products and services

IMD, Item Description
A segment for describing product or service related information valid for the whole order. This segment should be used for describing products or services that cannot be identified through codes.

UNS, Section Control
A service segment placed after the last user segment in the heading section to indicate the end of that section.

2.1.2 Detail Section

Information to be provided in the detail section:

Segment Group 10
A group of segments providing details of the individual ordered items.

LIN, Line Item
A segment identifying the details of the product or service being ordered, e.g. product identification, quantity ordered, price. All other segments in the detail section following the LIN segment refer to the item line.

RFF, References
A segment for referencing documents relating to the line item, e.g. a contract and its appropriate line item.

PIA, Additional Product Identification
A segment providing additional product identification.

IMD, Item Description
A segment for describing the product or service being ordered. This segment should be used for products that cannot be identified by a product code or article number.

MEA, Measurements
A segment specifying physical measurements of the ordered item, in original or unpacked form.

Segment Group 11: PAC-MEA-PCI

A group of segments identifying the packaging, physical dimensions, and marks and numbers for goods referenced in the ordered line item.

PAC, Package
A segment specifying the number of units per package and the physical type of packaging for the ordered line item.

MEA, Measurements
A segment specifying physical measurements of packages described in the PAC segment.

PCI, Package Identification
A segment specifying markings and labels used on individual physical units (packages) described in the PAC segment.

Segment Group 12 : TRI-LOC
A group of segments specifying tax related information, and when necessary the locations, to which that tax information relates.

TRI, Tax Related Information
A segment specifying the tax type, category, and rate, or exemption, relating to the order line item.

LOC, Location Identification
A segment indicating the location to which the tax or exemption specified in the TRI segment applies.

Segment Group 13 : NAD-LOC-RFF-DOC-CTA
A group of segments identifying names, addresses, and locations, contacts and references, and required supporting documents relevant to the order line item.

NAD, Name and Address
A segment for identifying names and addresses and their functions relevant to the ordered items.

For segment descriptions see segment Group 1 in the heading section.

SDQ, Destination Quantity
A segment specifying destination and related quantity details for split delivery.

ACA, Alternative Currency Amount
A segment specifying the amount and currency in a currency different to the message base currency.

ALI, Additional Information
A segment indicating that the line item is subject to special conditions owing to origin, customs preference, or commercial factors

DTM, Date/Time Reference
A segment specifying date/time details for transactions relating to the line item.

FTX, Free Text
A segment with free text in coded or clear form, to give further clarification, when required, to an ordered line item. In computer to computer exchanges such text will normally require the receiver to process this segment manually.

Segment Group 14 : ALC-ALI-TRI-ACA-FTX
A group of segments specifying allowances and charges for the ordered line item. Where relevant, tax and alternate currency details are to be indicated in the TRI and ACA segments.

ALC, Allowances and Charges
A segment indicating any charges or allowances for the ordered line item, e.g. rebate, packing charge, delivery charge.

ALI, Additional Information
A segment indicating that allowances or charges related to the line item are subject to special conditions owing to origin, customs preference or commercial factors.

TRI, Tax Related Information
A segment specifying the tax type, category, and rate, or exemption, related to the allowance or charge.

ACA, Alternative Currency Amount
A segment specifying the amount and currency of the allowance or charge in a currency different to the message base currency.

FTX, Free Text
A segment with free text information, in coded or clear form, relating to the allowance or charge. In computer to computer exchanges such text will normally require the receiver to process this segment manually.

Segment Group 15 : API-FTX
A group of segments specifying additional pricing information used in conjunction with ordered line item.

API, Additional Price Information
A segment providing information concerning pricing related to type of trade, quantity, date/period, or amount.

FTX, Free Text
A segment with free text information, in coded or clear text form, relating to the pricing information. In computer to computer exchanges such text will normally require the receiver to process this segment manually.

Segment Group 16 + 17 : TDT-LOC-DTM
A group of segments specifying details of the mode of transport, location, and date/time of departure and destination relating to the ordered line item.

For segment descriptions see segment Group 4 + 5 in the heading section.

GIN, Goods Identity Numbers
A segment providing identity numbers to be applied to thegoods being ordered, e.g. serial numbers for assembled equipment.

GIR, Goods Identification Related Numbers
A segment providing sets of related identification numbers for a line item, e.g. engine numbers, chassis number and transmission number for a vehicle

TOD, Terms of Delivery
A segment specifying the terms of delivery and transfer of title locations for the ordered line item.

Segment Group 18 : EQF-EQA-HAN-FTX
A group of segments providing information relating to equipment identification and ownership, handling, and notification of hazardous materials, relating to the ordered line item.

For segment descriptions, see Segment Group 7 in the headingsection.

Segment Group 19 + 20 : SCC-QTY-DTM-FTX
A group of segments specifying requested delivery schedules relating to quantities, frequencies, and dates, required for the ordered line item.

For segment descriptions, see Segment Group 8 + 9 in the heading section.

LTM, Lead Time
A segment for providing information concerning lead times with regard to availability of products and services.

Segment Group 21 : SID

A group of segments comprising the Sub-Detail Section,providing details of individual sub-items to an ordered line item.

SID, Sub-Line Item Details (Configuration)
A segment to permit the sub-division of a line item into its components, e.g. computer system itemised into separate parts such as CPU, terminals, disk drives, etc. The sub-line detail segment may be followed by the segments and segment groups following the LIN segment as indicated in the branch diagram.

UNS, Section Control
A service segment placed after the last user data segment in the detail section to indicate the end of that section.

2.1.3 Summary Section

Information to be provided in the Summary Section:

TMA, Total Message Amounts
A segment giving the total amounts for the whole purchase order message.

ACT, Alternative Currency Total Amount
A segment specifying the summary amount and currency in a currency different from the message base currency.

FTX, Free Text
A segment with free text information, to give further clarification, when required, for the summary section only. In computer exchanges, such text will normally require the receiver to handle this segment manually.

Segment Group 32 : ALC-ALI-TRI-ACA-FTX
A group of segments specifying allowances and charges for the whole purchase order. Where relevant, allowance, charge, tax, and currency details are to be indicated in the ALI, TRI and ACA segments.

For the segment description see Group 14 in the Detail Section

CNT, Control Totals
A segment by which control totals may be provided by the sender for checking by the receiver.

UNT, Message Trailer
A service segment ending a message, giving the total number of segments in the message and the control reference number of the message.

2.2 Segment Table

TAG	NAME	S	REPT	S	REPT
UNH	Message Header	M	1		
BGM	Beginning of message	M	1		
RFF	References	C	10		
CTA	Contact Segment	C	10		
----- Segment Group 1 ----------				C	20---------!
NAD	Name and address	M	1		!
LOC	Location identification	C	5		!
RFF	References	C	10		!
DOC	Documents Required	C	5		!
CTA	Contact Segment	C	5		!
FII	Financial institution information	C	5 ----------		!
DTM	Date/Time Reference	C	5		

----- Segment Group 2 ----------		C	5 ----------
TRI	Tax Related Information	M	1
LOC	Location identification ----------	C	5 ----------
----- Segment Group 3 ----------		C	5 ----------
CUX	Currencies	M	1
DTM	Date/Time Reference----------	C	5 ----------
ALI	Additional Information	C	5
FTX	Free Text	C	5
PAT	Payment Terms Basis	C	10
PAI	Payment Instructions	C	1
----- Segment Group 4 ----------		C	10 ----------
TDT	Details of transport	M	1
----- Segment Group 5 ----------		C	10 -------
LOC	Location identification	M	1
DTM	Date/Time Reference ----------	C	1 ----------
TOD	Terms of delivery	C	5
----- Segment Group 6 ----------		C	10 ---------
PAC	Package	M	1
MEA	Measurements	C	5
PCI	Package Identification	C	5 ----------
----- Segment Group 7 ----------		C	10 ---------
EQF	Equipment fixed information	M	1
EQA	Equipment attached	C	5
HAN	Handling instruction	C	5
FTX	Free Text	C	5 ----------
------ Segment Group 8 ----------		C	10 -------
SCC	Scheduling conditions	M	1
----- Segment Group 9 ----------		C	10 -----!
QTY	Quantity	M	1
DTM	Date/Time Reference	C	2 ----------
FTX	Free Text	C	5 ----------
API	Additional price information	C	25
LTM	Lead time	C	15
IMD	Item Description	C	10
UNS	Section Control	M	1
----- Segment Group 10 ----------		C	9999 ------
LIN	Line item	M	1
RFF	References	C	10
PIA	Additional product id	C	10
IMD	Item Description	C	10
MEA	Measurements	C	5

----- Segment Group 11 ----------		C	10 -----
PAC	Package	M	1
MEA	Measurements	C	10
PCI	Package Identification	C	10 ----------
----- Segment Group 12 ----------		C	5 --------
TRI	Tax Related Information	M	1
LOC	Location identification	C	5 ----------
----- Segment Group 13 ----------		C	10 ------
NAD	Name and address	M	1
LOC	Location identification	C	5
RFF	References	C	5
DOC	Documents Required	C	5
CTA	Contact Segment	C	5 ----------
SDQ	Destination Quantity	C	10
ACA	Alternative currency amount	C	5
ALI	Additional Information	C	5
DTM	Date/Time Reference	C	5
FTX	Free Text	C	5
----- Segment Group 14 ----------		C	10 ----------
ALC	Allowances and charges	M	1
ALI	Additional Information	C	5
TRI	Tax Related Information	C	5
ACA	Alternative currency amount	C	5
FTX	Free Text	C	5 ----------
------- Segment Group 15 ----------		C	20 ----------
API	Additional price information	M	1
FTX	Free Text	C	5 ----------
----- Segment Group 16 ----------		C	10 ----------
TDT	Details of transport	M	1
----- Segment Group 17 ----------		C	10 ----------
LOC	Location identification	M	1
DTM	Date/Time Reference	C	5 ----------
GIN	Goods Identity Number	C	100
GIR	Goods identification related numbers	C	100
TOD	Terms of delivery	C	5
----- Segment Group 18 ----------		C	10 --------
EQF	Equipment fixed information	M	1
EQA	Equipment attached	C	5
HAN	Handling instruction	C	5
FTX	Free Text	C	5 ----------
----- Seg ment Group 19 ----------		C	100 --------
SCC	Scheduling conditions	M	1

----- Segment Group 20 -----		C	10	
QTY	Quantity	M	1	
DTM	Date/Time Reference	C	2	
FTX	Free Text	C	5	
LTM	Lead time	C	15	
----- Segment Group 21 -----		C	200	
SID	Sub-line item details	M	1	
RFF	References	C	10	
PIA	Additional product id	C	10	
IMD	Item Description	C	10	
MEA	Measurements	C	5	
----- Segment Group 22 -----		C	10	
PAC	Package	M	1	
MEA	Measurements	C	10	
PCI	Package Identification	C	10	
---- Segment Group 23 ----		C	5	
TRI	Tax Related Information	M	1	
LOC	Location identification	C	5	
---- Segment Group 24 ----		C	10	
NAD	Name and address	M	1	
LOC	Location identification	C	5	
RFF	References	C	5	
DOC	Documents Required	C	5	
CTA	Contact Segment	C	5	
SDQ	Destination Quantity	C	10	
ACA	Alternative currency amount	C	5	
ALI	Additional Information	C	5	
DTM	Date/Time Reference	C	5	
FTX	Free Text	C	5	
---- Segment Group 25 ----		C	10	
ALC	Allowances and charges	M	1	
ALI	Additional Information	C	5	
TRI	Tax Related Information	C	5	
ACA	Alternative currency amount	C	5	
FTX	Free Text	C	5	
---- Segment Group 26 ----		C	20	
API	Additional price information	M	1	
FTX	Free Text	C	5	
---- Segment Group 27 ----		C	10	
TDT	Details of transport	M	1	
---- Segment Group 28 ----		C	10	
LOC	Location identification	M	1	
DTM	Date/Time Reference	C	5	

GIN	Goods Identity Number	C	100				
GIR	Goods identification related numbers	C	100				
TOD	Terms of delivery	C	5				
---- Segment Group 29 ----------				C	10	-------	
EQF	Equipment fixed information	M	1				
EQA	Equipment attached	C	5				
HAN	Handling instruction	C	5				
FTX	Free Text	C	5			----------	
---- Segment Group 30 ----------				C	100	-------	
SCC	Scheduling conditions	M	1				
---- Segment Group 31 ----------				C	10	-----	
QTY	Quantity	M	1				
DTM	Date/Time Reference	C	2			----------	
FTX	Free Text	C	5			----------	
LTM	Lead time	C	15			----------	
UNS	Section Control	M	1				
TMA	Total Message Amounts	C	1				
ACT	Alternative currency total amount	C	10				
FTX	Free Text	C	5				
---- Segment Group 32 ----------				C	10	--------	
ALC	Allowances and charges	M	1				
ALI	Additional Information	C	5				
ACA	Alternative currency amount	C	5				
TRI	Tax Related Information	C	5				
FTX	Free Text	C	5			----------	
CNT	Control totals	C	5				
UNT	Message Trailer	M	1				

2.3 Data Segments (alphabetic Sequence)

The following segments form part of the UNSM Purchase Order.

ACA Alternative Currency Amount
ACT Alternative Currency Total Amount
ALC Allowances and Charges
ALI Additional Line Information
API Additional Price Information
BGM Beginning of Message
CNT Control Totals
CTA Contacts
CUX Currencies
DOC Document Required
DTM Date/Time Reference
EQA Equipment Attached
EQF Equipment Fixed Information
FII Financial Institution Information

FTX	Free Text
GIN	Goods Identify Numbers
GIR	Goods Identification Related Numbers
HAN	Handling Instructions
IMD	Item Description
LIN	Line Item
LOC	Location Identification
LTM	Lead Time
MEA	Measurements
NAD	Name and Address
PAC	Package
PAI	Payment Instructions
PAT	Payment Terms Basis
PCI	Package Identification
PIA	Additional Product Identification
QTY	Quantity
RFF	References
SCC	Scheduling Conditions
SDQ	Destination Quantity
SID	Subline Item Details
TDT	Details of Transport
TMA	Total Message Amounts
TOD	Terms of Delivery
TRI	Tax Related Information
UNH	Message Header
UNS	Section Control
UNT	Message Trailer

INTERFACTS - Produced 90/02/27

UN/EDIFACT Data Segments Directory 90.1 Date Created : 90/02/15

UNB INTERCHANGE HEADER

Function : To start, identify and specify an Interchange

S001 SYNTAX IDENTIFIER	M
0001 Syntax identifier	M a4
0002 Syntax version number	M n1
S002 INTERCHANGE SENDER	M
0004 Sender identification	M an..35
0007 Partner identification code qualifier	C an..4
0008 Address for reverse routing	C an..14
S003 INTERCHANGE RECIPIENT	M
0010 Recipient identification code	M an..35
0007 Partner identification code qualifier	C an..4
0014 Routing address	C an..14
S004 DATE/TIME OF PREPARATION	M
0017 Date	M n6
0019 Time	M n4
0020 INTERCHANGE CONTROL REFERENCE	M an..14
S005 RECIPIENTS REFERENCE PASSWORD	C
0022 Recipient's reference/password	M an..14
0025 Recipient's reference/password qualifier	C an2
0026 APPLICATION REFERENCE	C an..14
0029 PROCESSING PRIORITY CODE	C a1
0031 ACKNOWLEDGEMENT REQUEST	C n1
0032 COMMUNICATIONS AGREEMENT IDENTIFIER	C an..35
0035 TEST INDICATOR	C n1

INTERFACTS - Produced 90/02/27

UN/EDIFACT Data Segments Directory 90.1 Date Created : 90/02/15

UNH MESSAGE HEADER

Function : To head, identify and specify a Message

0062 MESSAGE REFERENCE NUMBER	M an..14
S009 MESSAGE IDENTIFIER	M
0065 Message type	M an..6
0052 Message version number	M n..3
0054 Message release number	C n..3
0051 Controlling agency	C an..2
0057 Association assigned code	C an..6
0068 COMMON ACCESS REFERENCE	C an..35
S010 STATUS OF THE TRANSFER	C
0070 Sequence of transfers	M n..2
0073 First and last transfer	C a1

EDI Requirements / used standards capabilities	FTAM EDIFACT	P0/1 EDIFACT	P2 EDIFACT	X.435 EDIFACT	X.435 ASN.1
(1) The reliable transfer of a set of EDI messages between two computer systems	+	+	+	+	+
(2) The ability to transfer an EDI Interchange as a whole encoded in a syntax not recognised by X.400 systems	+	+	+	+	+
(3) The ability to hold EDI Interchanges in transfer for forwarding to the designated recipient when it is ready.	-	+	+	+	+
(4) The ability to select specific EDI Interchanges for receipt based on attributes of the EDI Interchange.	-		88 -	+	+
(5) The ability to select specific EDI Functional Groups within specific EDI Interchanges for receipt based on attributes of the EDI Interchange and the EDI Functional Group.	-				+
(6) The ability to list general information about EDI Interchanges awaiting retrieval from storage within the communication system.	-		88 -	+	+
(7) The ability to interwork with existing Value Added Netword for EDI.	+	+	+	+	+
(8) The ability to interwork internationally between different network providers.		+	+	+	+
(9) The ability to interwork with a private message handling network, run by an organization for its own use.		+	+	+	+
(10) The ability to interwork, possibly via an application gateway, with an InterPersonal Messaging System (IPMS) based on X.400.			+	+	+

(11) The ability to log all the parts or selected attributes of outgoing and incoming EDI Interchanges for audit purposes.				+	+
(12) The ability to correlate incoming acknowledgements with the outgoing message log.				+	+
(13) The ability to authenticate the origin and integrity of EDI Interchange to the recipient and to third parties.		88	88	+	+
(14) The ability to restrict delivery of EDI Interchanges to those containing EDI messages of a specific type from specific originators with whom there is an agreement to interchange data.	+	88 -	88 -	+	+
(15) The ability to restrict flow of EDI Interchanges between interconnected sub-networks to those of a given type between specific Senders and Recipient pairs.		88	88	+	+
(16) The ability to separate flow of EDI messages and InterPersonal Messages.		+		+	+
(17) The ability to handle EDI messages without the need to interpret the message contents.	+	+	+	+	+
(18) The ability to cipher EDI Interchanges.		88	88	+	+
(19) The ability to notify the EDI Sender of successful transfer of an EDI Interchange to the recipient's store awaiting retrieval.	+	+	+	+	+
(20) The ability to notify the EDI sender of failure to transfer an EDI Interchange.	+	+	+	+	+
(21) The ability to notify the EDI Sender of retrieval of an EDI Interchange/Functional Group by the recipient.	+		+	+	+

(22) The ability to notify the EDI Sender of failure of the recipient to retrieve an EDI Interchange or Functional Group.	+		+	+	+
(23) The ability to notify the EDI Sender of acceptance for processing of all/some messages in an EDI Interchange.					
(24) The ability to notify the EDI Sender of non-acceptance of specific messages within an EDI Interchange.					
(25) The ability for the user or a third party cleared to administer the user to up-date information used to manage the provision of messaging to the user independent of the provider of message store/transfer facilities.	+			+	+
(26) The ability to specify the urgency of delivery of a specific EDI Interchange.		+	+	+	+
(27) The ability to deliver the same interchange to several recipients.	-				+
(28) The ability to convert between different syntax encodings.		-	-	+	+

KEY :

+	Requirement fully met
-	Requirement partially met
88	Requirement only met by using 1988 X.400

source : [VANG 88]

```
EDI_Interchange                 ::= SEQUENCE {
                                        [0] Interchange_Heading,
                                        [1] Interchange_Body }

Interchange_Heading             ::= SEQUENCE {
    syntax-identifier                   [0] OBJECT IDENTIFIER,
    interchange-sender                  [1] OBJECT IDENTIFIER,
    interchange-recipient               [2] OBJECT IDENTIFIER,
    date-time-of-preparation            [3] UCTTime,
    interchange-control-reference       [4] T61String(SIZE(1..14)) }

Interchange_Body                ::= SEQUENCE { EDI_Message }

EDI_Message                     ::= CHOICE {
                                        [0] Purchase_Order,
                                            }

Purchase_Order                  ::= SEQUENCE {
    message-header                      [0] Message_Header,
    begin-of-message                    [1] Begin_Of_Message,
    reference                           [2] SEQUENCE (1..10) OF {Reference}
                                            OPTIONAL,
    contact-segment                     [3] SEQUENCE (1..10) OF {Contact_Segment }
                                            OPTIONAL,
    name-and-address                    [4] Name_And_Address,
    location-identification             [5] SEQUENCE (1..5) OF
                                            {Location_Identification} OPTIONAL,
    references                          [6] SEQUENCE (1..10) OF {References}
                                            OPTIONAL,
    document-required                   [7] SEQUENCE (1..5) OF {Document_
                                            Required} OPTIONAL,
    contact-segment                     [8] SEQUENCE (1..5) OF {Contact_Segment}
                                            OPTIONAL,
    financial-institution-information   [9] SEQUENCE (1..5) OF{Financial_Institution_
                                            Information} OPTIONAL,
    date/time-reference                 [10] SEQUENCE (1..5) OF {Date/time_
                                            Reference} OPTIONAL,
    tax-related-information             [11] Tax_Related_Information,
    location-identification             [12] SEQUENCE (1..5) OF {Location_
                                            Identification} OPTIONAL,
    currencies                          [13] Currencies,
    date/time-reference                 [14] SEQUENCE (1..5) OF {Date/time_
                                            Reference} OPTIONAL,
    additional-information              [15] SEQUENCE (1..5) OF {Additional_
                                            Information} OPTIONAL,
```

```
free-text                       [16] SEQUENCE (1..5) OF {Free_Text
                                     OPTIONAL,
payement-terms-basis            [17] SEQUENCE (1..10) OF {Payement_
                                     Terms_Basis} OPTIONAL,
payement-instruction            [18] Payement_Instruction OPTIONAL,
details-of-transport            [19] Details_Of_Transport,
location-of-transport           [20] Location_Of_Transport,
date/time-reference             [21] Date/time_Reference OPTIONAL,
terms-of-delivery               [22] SEQUENCE (1..5) OF {Terms_
                                     Of_Delivery} OPTIONAL,
package                         [23] Package,
measurements                    [24] SEQUENCE (1..5) OF {Measurements}
                                     OPTIONAL,
package-identification          [25] SEQUENCE (1..5) OF
                                     {Package_Identification} OPTIONAL,
equipment-fixed-information     [26] Equipment_Fixed_Information,
equipment-attached              [27] SEQUENCE (1..5) OF
                                     {Equipment_Attached} OPTIONAL,
handling-instruction            [28] SEQUENCE (1..5) OF
                                     {Handling_Instruction} OPTIONAL,
free-text                       [29] SEQUENCE (1..5) OF {Free_Text}
                                     OPTIONAL,
scheduling-conditions           [30] Scheduling_Conditions,
quantity                        [31] Quantity,
date/time-reference             [32] SEQUENCE (1..2) OF
                                     {Date/time_Reference} OPTIONAL,
free-text                       [33] SEQUENCE (1..5) OF {Free_Text }
                                     OPTIONAL,
additional-price-information    [34] SEQUENCE (1..25) OF {Additional_Price_
                                     Information} OPTIONAL,
lead-time                       [35] SEQUENCE (1..15) OF {Lead_Time}
                                     OPTIONAL,
item-description                [36] SEQUENCE (1..10) OF {Item_Description}
                                     OPTIONAL,
section-control                 [37] Section_Control,
line-item                       [38] Line_Item,
references                      [39] SEQUENCE (1..10) OF {References}
                                     OPTIONAL,
additional-product-id           [40] SEQUENCE (1..10) OF {Additional_
                                     Product_Id} OPTIONAL,
item-description                [41] SEQUENCE (1..10) OF {Item_Description}
                                     OPTIONAL,
measurements                    [42] SEQUENCE (1..5) OF {Measurements}
                                     OPTIONAL,
package                         [43] Package,
```

```
measurements                    [44] SEQUENCE (1..10) OF {Measurements}
                                     OPTIONAL,
package-identification          [45] SEQUENCE (1..10) OF {Package_
                                     Identification} OPTIONAL,
tax-related-information         [46] Tax_Related_Information,
location-identification         [47] SEQUENCE (1..5) OF {Location_
                                     Identification} OPTIONAL,
name-and-address                [48] Name_And_Address,
location-identifiation          [49] SEQUENCE (1..5) OF {Location_
                                     Identification} OPTIONAL,
references                      [50] SEQUENCE (1..5) OF {References}
                                     OPTIONAL,
document-required               [51] SEQUENCE (1..5) OF {Document
                                     _Required} OPTIONAL,
contact-segment                 [52] SEQUENCE (1..5) OF {Contact_Segment}
                                     OPTIONAL,
destination-quantity            [53] SEQUENCE (1..10) OF
                                     {Destination_Quantity} OPTIONAL,
alternative-currency-amount     [54] SEQUENCE (1..5) OF
                                     {Alternative_Currency_Amount}
                                     OPTIONAL,
additional-information          [55] SEQUENCE (1..5) OF
                                     {Additional_Information} OPTIONAL,
date/time-reference             [56] SEQUENCE (1..5) OF
                                     {Date/Time_Reference} OPTIONAL,
free-text                       [57] SEQUENCE (1..5) OF {Free_Text}
                                     OPTIONAL,
allowances-and-charges          [58] Allowances_And_Charges,
additional-information          [59] SEQUENCE (1..5) OF
                                     {Additional_Information} OPTIONAL,
tax-related-information         [60] SEQUENCE (1..5) OF
                                     {Tax_Related_Information} OPTIONAL,
alternative-currency-amount     [61] SEQUENCE (1..5) OF
                                     {Alternative_Currency_Amount}
                                     OPTIONAL,
free-text                       [62] SEQUENCE (1..5) OF {Free_Text}
                                     OPTIONAL,
additional-price-information    [63] Additional_Price_Information,
free-text                       [64] SEQUENCE (1..5) OF {Free_Text}
                                     OPTIONAL,
details-of-transport            [65] Details_Of_Transport,
location-identification         [66] Location_Identification,
date/time-reference             [67] SEQUENCE (1..5) OF
                                     {Date/Time_Reference} OPTIONAL,
goods-identify-number           [68] SEQUENCE (1..100) OF
                                     {Goods_Identify_Number} OPTIONAL,
```

goods-identification-related-numbers[69] SEQUENCE (1..100) OF
{Goods_Identification_Related_Numbers}
OPTIONAL,
terms-of-delivery [70] SEQUENCE (1..5) OF
{Terms_Of_Delivery} OPTIONAL,
equipement-fixed-information [71] Equipment_Fixed_Information,
equipment-attached [72] SEQUENCE (1..5) OF
{Equipment_Attached} OPTIONAL,
handling-instruction [73] SEQUENCE (1..5) OF
{Handling_Instruction} OPTIONAL,
free-text [74] SEQUENCE (1..5) OF {Free_Text}
OPTIONAL,
scheduling-conditions [75] Scheduling_Conditions,
quantity [76] Quantity,
date/time-reference [77] SEQUENCE (1..2) OF
{Date/Time_Reference} OPTIONAL,
free-text [78] SEQUENCE (1..5) OF {Free_Text}
OPTIONAL,
lead-time [79] SEQUENCE (1..15) OF {Lead_Time}
OPTIONAL,
sub-line-item-details [80] Sub_Line_Item_Details,
references [81] SEQUENCE (1..10) OF {References}
OPTIONAL,
additional-product-id [82] SEQUENCE (1..10) OF
{Additional_Product_Id} OPTIONAL,
item-description [83] SEQUENCE (1..10) OF {Item_Description}
OPTIONAL,
measurements [84] SEQUENCE (1..5) OF {Measurements}
OPTIONAL,
package [85] Package,
measurements [86] SEQUENCE (1..10) OF {Measurements
package-identification [87] SEQUENCE (1..10) OF
{Package_Identification
tax-related-information [88] Tax_Related_Information,
location-identification [89] SEQUENCE (1..5) OF
{Location_Identification} OPTIONAL,
name-and-address [90] Name_And_Address,
location-identification [91] SEQUENCE (1..5) OF
{Location_Identification} OPTIONAL,
references [92] SEQUENCE (1..5) OF {References}
OPTIONAL,
documents-required [93] SEQUENCE (1..5) OF
{Documents_Required} OPTIONAL,
contact-segment [94] SEQUENCE (1..5) OF {Contact_Segment}
OPTIONAL,

destination-quantity [95] SEQUENCE (1..10) OF {Destination_Quantity} OPTIONAL,
alternative-currency-amount [96] SEQUENCE (1..5) OF {Alternative_Currency_Amount} OPTIONAL,
additional-information [97] SEQUENCE (1..5) OF {Additional_Information} OPTIONAL,
date/time-reference [98] SEQUENCE (1..5) OF {Date/Time_Reference} OPTIONAL,
free-text [99] SEQUENCE (1..5) OF {Free_Text} OPTIONAL,
allowances-and-charges [100] Allowances_And_Charges,
additional-information [101] SEQUENCE (1..5) OF {Additional_Information} OPTIONAL,
tax-related-information [102] SEQUENCE (1..5) OF {Tax_Related_Information} OPTIONAL,
alternative-currency-amount [103] SEQUENCE (1..5) OF {Alternative_Currency_Amount} OPTIONAL,
free-text [104] SEQUENCE (1..5) OF {Free_Text} OPTIONAL,
additional-price-information [105] Additional_Price_Information,
free-text [106] SEQUENCE (1..5) OF {Free_Text} OPTIONAL,
details-of-transport [107] Details_Of_Transport,
location-identification [108] Location_Identification,
date/time-reference [109] SEQUENCE (1..5) OF {Date/Time_Reference} OPTIONAL,
goods-identify-number [110] SEQUENCE (1..100) OF {Goods_Identify_Number} OPTIONAL,
goods-identification-related-numbers [111] SEQUENCE (1..100) OF {Goods_Identification_Related_Numbers} OPTIONAL,
terms-of-delivery [112] SEQUENCE (1..5) OF {Terms_Of_Delivery} OPTIONAL,
equipement-fixed-information [113] Equipment_Fixed_Information,
equipment-attached [114] SEQUENCE (1..5) OF {Equipment_Attached} OPTIONAL,
handling-instruction [115] SEQUENCE (1..5) OF {Handling_Instruction} OPTIONAL,
free-text [116] SEQUENCE (1..5) OF {Free_Text} OPTIONAL,
scheduling-conditions [117] Scheduling_Conditions,
quantity [118] Quantity,
date/time-reference [119] SEQUENCE (1..2) OF {Date/Time_Reference} OPTIONAL,

```
        free-text                       [120] SEQUENCE (1..5) OF {Free_Text}
                                              OPTIONAL,
        lead-time                       [121] SEQUENCE (1..15) OF {Lead_Time}
                                              OPTIONAL,
        section-control                 [122] Section_Control,
        total-message-amounts           [123] Total_Message_Amount OPTIONAL,
        alternative-currency-total-amonut [124] SEQUENCE (1..10) OF
                                              {Alternative_Currency_Total_Amount}
                                              OPTIONAL,
        free-text                       [125] SEQUENCE (1..5) OF {Free_Text}
                                              OPTIONAL,
        allowances-and-charges          [126] Allowances_And_Charges,
        additional-information          [127] SEQUENCE (1..5) OF
                                              {Additional_Information} OPTIONAL,
        alternative-currency-amount     [128] SEQUENCE (1..5) OF
                                              {Alternative_Currency_Amount}
                                              OPTIONAL,
        tax-related-information         [129] SEQUENCE (1..5) OF
                                              {Tax_Related_Information} OPTIONAL,
        free-text                       [130] SEQUENCE (1..5) OF {Free_Text}
                                              OPTIONAL,
        control-totals                  [131] SEQUENCE (1..5) OF {Control_Totals}
                                              OPTIONAL,
        message-trailer                 [132] Message_Trailer,

Message_Header                  ::= SEQUENCE {
        message-reference-number        [0] T61String(SIZE(1..14)),
        message-identifier              [1] Message_Identifier,
        common-access-reference         [2] T61String(SIZE(1..35)) OPTIONAL,
        status-to-the-transfer          [3] Staus_To_The_Tranfer OPTIONAL }

Begin_Of_Message                ::= SEQUENCE {
        document                        [0] Document,
        document-number                 [1] T61String(SIZE(1..35)),
        date/time-of-document           [2] Date/Time_Of_Document,
        Message-function-code           [3] INTEGER OPTIONAL,
        Primary-reference               [4] Primary_Reference OPTIONAL,
        date/time-of-reference          [5] Date/Time_Of_Reference OPTIONAL,
        response-type                   [6] T61String(SIZE(1..2)) OPTIONAL }

Message_Identifier              ::= SEQUENCE {
        message-type                    [0] T61String(SIZE(1..6)),
        message-version-number          [1] INTEGER,
        message-release-number          [2] INTEGER OPTIONAL,
        controlling agency              [3] T61String(SIZE(1..2)) OPTIONAL,
        association-assigned-code       [4] T61String(SIZE(1..6)) OPTIONAL }
```

```
Status_To_Transfer              ::= SEQUENCE {
    sequence-of-transfers           [0] INTEGER,
    first-and-last-transfer         [1] T61String(SIZE(1..1)) OPTIONAL }

Document                        ::= SEQUENCE {
    document-name-coded             [0] INTEGER OPTIONAL,
    document-name                   [1] T61String(SIZE(1..35)) OPTIONAL}

Date/time_Of_Document           ::= SEQUENCE {
    date-coded                      [0] INTEGER OPTIONAL,
    time                            [1] INTERGER OPTIONAL}

Primary_Reference               ::= SEQUENCE {
    reference-number                [0] T61String(SIZE(1..35)),
    reference-qualifier             [1] T61String(SIZE(1..3)) OPTIONAL}

Date/Time_Of_Reference          ::= SEQUENCE {
    date-coded                      [0] INTEGER OPTIONAL,
    date                            [1] INTEGER OPTIONAL }
```

The following is the "UNSM - PURCHASE ORDER MESSAGE" document UN-ECE/TRADE/WP.4/R.534 in ASN.1.

```
EDI_Interchange                 ::= SEQUENCE {
                                        [0] Interchange_Heading,
                                        [1] Interchange_Body }

Interchange_Heading             ::= SEQUENCE {
    syntax-identifier                   [0] OBJECT IDENTIFIER,
    interchange-sender                  [1] OBJECT IDENTIFIER,
    interchange-recipient               [2] OBJECT IDENTIFIER,
    date-time-of-preparation            [3] UCTTime,
    interchange-control-reference       [4] T61String(SIZE(1..14)) }

Interchange_Body                ::= SEQUENCE { EDI_Message }

EDI_Message                     ::= CHOICE {
                                        [0] Purchase_Order,
                                        [1] Invoice }

Purchase_Order                  ::= SEQUENCE {
                                        [0] PO_Heading,
                                        [1] Parties_Involved,
                                        [2] Currency,
                                        [3] SEQUENCE OF{
                                            Purchase_Order_Lines } }

PO_Heading                      ::= SEQUENCE {
    message-reference-number            [0] T61String(SIZE(1..14)),
    document-number                     [1] T61String(SIZE(1..35)),
    date                                [2] UCTTime1,
    primary-reference-number            [3] T61String(SIZE(1..35)),
    primary-reference-qualifier         [4] T61String(SIZE(1..3)) }

Parties_Involved                ::= SEQUENCE {
    contact                             [0] Name_Address,
    buyer                               [1] Long_Address,
    seller                              [2] Long_Address,
    invoice                             [3] Long_Address OPTIONAL}

Name_Address                    ::= SEQUENCE {
    fonction                            [0] T61String(SIZE(1..2)),
    department                          [1] T61String(SIZE(1..17)),
    employee-name                       [2] T61String(SIZE(1..35)),
    communication-contact               [3] SEQUENCE (1..5) of
                                            Cm_Number_Channel }
```

```
Cm_Number_Channel          ::= SEQUENCE {
    communication-number        [0] T61String(SIZE(1..25)),
    communication-channel       [1] T61String(SIZE(1..2)) }

Long_Address               ::= SEQUENCE {
    name                        [0] T61String(SIZE(1..105)),
    street-and-number           [1] T61String(SIZE(1..105)),
    postal-code                 [2] T61String(SIZE(1..9)),
    city                        [3] T61String(SIZE(1..35)),
    country-coded               [4] T61String(SIZE(1..2)) OPTIONAL }

Currency                   ::= SEQUENCE {
    currency-reference          [0] T61String(SIZE(1..3)),
    currency-qualifier          [1] T61String(SIZE(1..3))  OPTIONAL }

Purchase_Order_Lines       ::= SEQUENCE {
                                [0] Product,
                                [1] Purchase_Order_Item,
                                [2] Addit_Product_Characteristic,
                                [3] Date_And_Time_Reference }

Product                    ::= SEQUENCE {
    seller-code                 [0] T61String(SIZE(1..35)),
    buyer-code                  [1] T61String(SIZE(1..35)) }

Purchase_Order_Item        ::= SEQUENCE {
    number                      [0] INTEGER,
    quantity                    [1] REAL,
    quantity-unit-specifier     [2] OBJECT IDENTIFIER,
    quantity-qualifier          [3] OBJECT IDENTIFIER,
    unit-price                  [4] REAL,
    unit-price-basis            [5] REAL,
    pricing-quantity-basis      [6] OBJECT IDENTIFIER,
    unit-price-qualifier        [7] OBJECT IDENTIFIER }

Addit_Product_Characteristic ::= SEQUENCE {
    description                 [0] T61String(SIZE(1..70)) }

Date_And_Time_Reference    ::= SEQUENCE {
    date-qualifier              [0] OBJECT IDENTIFIER,
    date                        [1] UCTTime1 }
```

The EDIFACT representation of the example coming from [TAYL 90] of the purchase order is shown below :

UNA:+.? 'UNB+UNOA:1+7265739+526529+870527:1030+121
75'UNH+10505+ORDERS:1'BGM+105+00355822+880701++87-
47A:CT'CTA+PD+BM-25:BARBARA MUELLER+07031/14-2448:
TE'NAD+ST+1:92++HEWLETT PACKART GMBH:WARENEINGANG
2/FACTORY+DORNIERSTR. 7+BOEBLINGEN++7030'NAD+SE+12
12:92++IBM DEUTSCHLAND GMBH+PESNERSTR. 1+SINDEL FI
NGEN++7030'CUX+DEM:OC'UNS+D'LIN+1++10833C:VP+80673
-040:BP+250:21:PCE+56.44:CT:1:PCE'IMD+ER+++COMPONE
NTS'DTM+002+880730'UNS+S'UNT+12+10505'UNZ+1+12175'

The EDIFACT message is made of 450 octets.

ASN.1 is an abstract syntax, so there is no defined representation. The coding is made from a concrete syntax into a flow of bytes. In order to give an exact idea of the number of octets used, the structure of previous example is given below. (The presentation is same as in part 5.4).

	Number of octets	Value
EDI_Interchange ::= SEQUENCE	**384+(2)**(48+336)	V
[0] Interchange_Heading	**46+(2)** (8+9+8+14+7)	V
syntax-identifier [0] OBJECT IDENTIFIER	6+(2)	"UNOA 1"
interchange-sender [1] OBJECT IDENTIFIER	7+(2)	"7265739"
interchange-recipient [2] OBJECT IDENTIFIER	6+(2)	"526529"
date-time-of-preparation [3] UCTTime	12+(2)	"870527103000"
interchange-control-reference [4] T61String(SIZE(1..14))	5+(2)	"12175"
[1] Interchange_Body	**334+(2)**(333+1)	V
EDI_Message	331+(2) (330+1)	V
[0] Purchase_Order	328+(2) (327+1)(39+187+11+90)	V
[0] PO_Heading	***37+(2)*** (7+10+8+8+4)	V
message-reference-number [0] T61String(SIZE(1..14))	5+(2)	"10505"

document-number [1] T61String(SIZE(1..35))	8+(2)	"00355822"
date [2] UCTTime1	6+(2)	"880701"
primary-reference-number [3] T61String(SIZE(1..35))	6+(2)	"87-47A"
primary-reference-qualifier [4] T61String(SIZE(1..3))	2+(2)	"CT"
[1] Parties_Involved	***187+(2)*** (186+1) (49+78+59)	V
contact [0] Name_Address	*47+(2)* (4+7+15+21)	V
fonction [0] T61String(SIZE(1..2))	2+(2)	"PD"
departement [1] T61String(SIZE(1..17))	5+(2)	"BM-25"
employee-name [2] T61String(SIZE(1..35))	13+(2)	"BARBARA MUELLER"
communication-contact [3] SEQUENCE (1..5) OF Cm_Number_Channel	19+(2) (15+4)	V
communication-number [0] T61String(SIZE(1..25))	13+(2)	"07031/14-2448"
communication-channel [1] T61String(SIZE(1..2))	2+(2)	"TE"
buyer [1] Long_Address	*76+(2)* (43+15+6+12)	V
name [0] T61String(SIZE(1..105))	41+(2)	"HEWLETT PACKART GMBH WARENEINGANG 2/FACTORY"
street-and-number [1] T61String(SIZE(1..105))	13+(2)	"DORNIERSTR. 7"
postal-code [2] T61String(SIZE(1..9))	4+(2)	"7030"
city [3] T61String(SIZE(1..35))	10+(2)	"BOEBLINGEN"
seller [2] Long_Address	*57+(2)* (22+14+6+15)	V
name [0] T61String(SIZE(1..105))	20+(2)	"IBM DEUTSCHLAND GMBH"
street-and-number [1] T61String(SIZE(1..105))	12+(2)	"POSNERSTR. 1"
postal-code [2] T61String(SIZE(1..9))	4+(2)	"7030"
city [3] T61String(SIZE(1..35))	13+(2)	"SINDEL FINGEN"

[2] Currency	***9+(2)*** (5+4)	V
currency-reference [0] T61String(SIZE(1..3))	3+(2)	"DEM"
currency-qualifier [1] T61String(SIZE(1..3)) OPTIONAL	2+(2)	"OC"
[3] SEQUENCE OF { Purchase_Order_Lines }	***88+(2)*** (21+38+14+15)	V
[0] Product	*19+(2)* (8+11)	V
seller-code [0] T61String(SIZE(1..35))	6+(2)	"10833C"
buyer-code [1] T61String(SIZE(1..35))	9+(2)	"80673-040"
[1] Purchase_Order_Item	*36+(2)* (3+5+5+4+7+3+5+4)	V
number [0] INTEGER	1+(2)	"1"
quantity [1] REAL	3+(2)	"250"
quantity-unit-specifier [2] OBJECT IDENTIFIER	3+(2)	"PCE"
quantity-qualifier [3] OBJECT IDENTIFIER	2+(2)	"21"
unit-price [4] REAL	5+(2)	"56.44"
unit-price-basis [5] REAL	1+(2)	"1"
pricing-quantity-basis [6] OBJECT IDENTIFIER	3+(2)	"PCE"
unit-price-qualifier [7] OBJECT IDENTIFIER	2+(2)	"CT"
[2] Addit_Product_Characteristic	*12+(2)* (12)	V
description [0] T61String(SIZE(1..70))	10+(2)	"COMPONENTS"
[3] Date_And_Time_Reference	*13+(2)* (5+8)	V
date-qualifier [0] OBJECT IDENTIFIER	3+(2)	"002"
date [1] UCTTime1	6+(2)	"880730"

The ASN.1 way needs 386 octets for coding the same example.

For the same information content, the EDIFACT coding needs 450 octets and the ASN.1 coding only needs 386 octets. The ASN.1 solution is a bit more than 14% shorter than the EDIFACT solution.