THESIS / THÈSE

MASTER IN COMPUTER SCIENCE

Analysis of the UN/EDIFACT Standard and Investigation of the Use of the Object-Oriented Approach to Overcome UN/EDIFACT Deficiencies

Crémer, Laurence

Award date:
1995

Awarding institution:
University of Namur

Facultés Universitaires Notre-Dame de la Paix, Namur,
Institut d'Informatique

Academic Year 1994-1995

Analysis of the UN/EDIFACT Standard and Investigation of the Use of the Object-Oriented Approach to Overcome UN/EDIFACT Deficiencies

Laurence Crémer

Thesis presented in order to obtain the degree
of *Licencié et Maître en Informatique*
by Laurence Crémer

Abstract

The objective of this thesis paper is to study the UN/EDIFACT Standard and investigate how object-oriented methodologies could be used in the design of EDIFACT messages in order to avoid possible inconsistencies and ambiguities in mapping data.

In the process we first analyse EDIFACT structures. Next follows an evaluation of the methods used in message design to identify problem areas requiring solutions. Finally, four methods to overcome EDIFACT deficiencies are detailed: the first two are adopted by two message development groups who use the object-oriented approach to model the data to be mapped into a new message. The two last ones consist in redefining EDIFACT structures in batch and interactive EDI. Before closing this paper, we assess the situation and give our opinion about the whole problem.

Résumé

L'objectif de ce mémoire est d'étudier le standard UN/EDIFACT et d'analyser de quelle façon les méthodes orientées objet pourraient être utilisées dans le développement des messages EDIFACT, afin d'éviter les incohérences et ambiguïtés qui peuvent survenir lors du mapping des données.

Pour ce faire, nous analysons d'abord les structures EDIFACT. Ensuite, nous évaluons les méthodes utilisées pour le développement des messages afin d'identifier les problèmes à résoudre. Finalement, quatre méthodes pour remédier aux déficiences d'EDIFACT sont présentées: les deux premières méthodes sont celles adoptées par deux groupes de développement de messages qui utilisent l'approche orientée objet pour modéliser les données à placer dans un nouveau message. Les deux autres méthodes consistent en une redéfinition des structures EDIFACT en batch et interactif EDI. Enfin, nous faisons le point sur la situation et exprimons notre avis sur les méthodes proposées.

Acknowledgements

We would like to first thank our mentor, Professor Philippe van Bastelaer, for his availability and his guidance all along this work.

We also wish to thank Ray Walker, SITPRO Chief Executive for giving us the opportunity to do a training period in his company, as well as John Berge, EDI Standards Department Director, and his Assistant Director, Peter Wilson, for advising us during our stay in SITPRO between August 1994 and January 1995. Many thanks too to each staff member who helped us in one way or other along the way.

We would like to give our sincere thanks to Richard Williams, Information Systems Consultant, and Edgar Gluck, EDI Consultant, for their contribution to this work and the time they gave us.

Finally, many thanks to each person who took part directly or indirectly in the accomplishment of this thesis paper.

Table of Contents

List of Figures

1 Introduction to the UN/EDIFACT Standard

2 Analysis of UN/EDIFACT Structures

3 Evaluation of UN/EDIFACT Structures

4 Introduction to the Object-Oriented Approach

5 Data Modelling to Design EDIFACT Messages

6 New EDIFACT Structures

List of Tables

1 Introduction to the UN/EDIFACT Standard

2 Analysis of UN/EDIFACT Structures

3 Evaluation of UNEDIFACT Structures

Introduction

The 92.1 UN/EDIFACT Directory includes fifty-two approved or recommended messages and many others at draft stage, designed by message development groups (MDGs). Each MDG covers one sector (e.g. trade, construction, health care...) and is responsible for the design of the messages in that sector. Within Western Europe there are twelve MDGs.

Although message design guidelines and rules exist, some MDGs have their own approach to designing messages and segments: each may have its own interpretation of the rules and their methods are influenced by the sector they work in.

> *"... there has been no clear distinction between the information requirements and the technical solution when designing our messages. The same information needs can be satisfied in many different ways. The solution chosen till now for the financial messages is just one of many alternatives." [1]*

This sometimes results in inconsistencies and ambiguities in mapping data. It could also lead to instability in message and segment structures in the future.

The objective of this thesis paper is to study the UN/EDIFACT standard and investigate how object-oriented methodologies could be used in the design of EDIFACT messages so that all messages would have a common relationship and would be understood by everybody.

Although there are seven chapters, the thesis paper consists of three principal phases:

- an analysis of EDIFACT structures;
- an evaluation of the methods used in message design to identify problem areas that require solutions; and
- an investigation of the use of the object-oriented approach in order to find a better way to design messages.

After a brief introduction to the EDIFACT standard, the purpose of the first phase is to analyse the message, segment and composite structures in relation to Message Design Guidelines and Rules, in order to understand the different approaches used to design messages. For this analysis, the 92.1 UN/EDIFACT Directory will be the source of reference. This will also include a comparative study of how the usage varies between the MDGs, for some of the more common segment groups.

The examples below illustrate that different approaches to designing messages still occur despite existing generic guidelines.

- Some segments are so simple that they contain only one composite data element, which is never used in another segment of any message either of status 1 or 2 of the 92.1 UN/EDIFACT Directory. An example is the ARD AMOUNTS RELATIONSHIP DETAIL segment. It consists of one C549 MONETARY FUNCTION composite data element only. ARD is the only segment that uses C549. There are at least fifteen other user segments which only contain one composite.
- As RULE 19 of Message Design Guidelines and Rules [2] specifies, a segment must have a function, however some segments appear to relate to several concepts: the NAD NAME AND ADDRESS segment can be considered to contain two or even three concepts: the name, the structured address and the unstructured address.

The second phase consists of an evaluation of the different approaches that will have been investigated during the first step of the work.

The following questions about the examples of the different structures found above require clarification.

- Segment and composite structures are very similar. The major difference is that the composite data element structure uses component data elements instead of data elements. This begs the question: when should a segment or a composite structure be used? The answer is currently not clear.
- Can a segment have one function and contain several concepts?

The consequences of these different approaches will be examined and this will be used to identify a number of design principles that the next step should meet.

The last step consists of an investigation of how the object-oriented approach could be used to solve the problems arising from current approaches. This might lead to new building block definitions and new design rules.

After an introduction to the object-oriented approach, we first detail the methods of the health care (MD9) and tourism, travel and leisure (MD8) message development groups. They use object-oriented data modelling principles to model the data which need to be transmitted in a new message.

Another way of applying the object-oriented approach applied to EDIFACT message design would mean defining classes of objects, attributes and relations between these classes. This leads to redefining EDIFACT concepts in terms of attributes, objects, classes and relations.

In this perspective, we present two new methodologies to design new EDIFACT structures: one applied to batch EDI and elaborated by Niels Rasmussen, the second one for I-EDI, established in the framework of a distinct project in which Richard Williams participates.

In the last chapter, we assess the situation and give our opinion about the whole problem.

Before closing this introduction, we would like to add that following each chapter is a glossary of the terms and the references used within the chapter in question. Moreover a global glossary of terms and collected references appearing in all the chapters can be found at the end of this work, as well as appendices.

Glossary of Terms

EDI ..... Electronic Data Interchange
EDIFACT ..... Electronic Data Interchange For Administration, Commerce and Transport
MDG ..... Message Development Group
MD8 ..... Health Care Message Development Group
MD9 ..... Tourism, Travel and Leisure Message Development Group
UN/ECE ..... United Nations Economic Commission for Europe
UN/EDIFACT ..... United Nations Rules for Electronic Data Interchange For Administration, Commerce and Transport

References

[1] Ebba Lonaeus & Henning Klwitter
An Alternative Approach to Message Design
SWEDIFACT Finans, Sweden, 26 October 1994

[2] UN/ECE
Design of UN/EDIFACT Messages: Guidelines and Rules
TRADE/WP.4/R.840/Rev.2
UN/ECE, Geneva (Switzerland), 11 January 1994

1

Introduction to the UN/EDIFACT Standard

1. Introduction

This chapter introduces briefly the UN/EDIFACT standard. Since we consider that this work is addressed to people used to EDIFACT, only the main features are treated.

The concept of the UN/EDIFACT standard is first explained. Afterwards, EDIFACT building blocks are detailed and some more explanations on the syntax are given. Finally, interactive EDI is introduced.

2. What is UN/EDIFACT?

EDI, Electronic Data Interchange, is defined as:

"the electronic transfer of computer processable data relating to a business or administrative transaction using an agreed standard to structure the data". [1]

UN/EDIFACT, which stands for United Nations rules for Electronic Data Interchange For Administration, Commerce and Transport, is an international standard to structure the data in EDI.

UN/EDIFACT was created in 1987. Its origin is the fusion of two standards:

- UN/GTDI (United Nations Guidelines for Trade Data Interchange), which was the first European EDI standard supported by the United Nations Economic Commission for Europe (UN/ECE); and
- ANSI X12 (American National Standard Institute), which is the national standard for North America.

The rules for UN/EDIFACT are a set of internationally agreed standards, directories and guidelines for electronic interchange of structured data that are related to trade in goods and services, between independent computerised information systems.

UN/EDIFACT is approved and published by the UN/ECE in the United Nations Trade Data Interchange Directory (TDID) and is maintained under agreed procedures.

UN/EDIFACT includes:

- the UN/EDIFACT Syntax Rules (ISO 9735). This document is the formal definition of UN/EDIFACT application level syntax rules;
- Message Design Guidelines, which are intended for message designers;
- Syntax Implementation Guidelines, which are intended for use by UN/EDIFACT implementation personnel. It expands the document above;
- the UN/EDIFACT Data Element Directory (EDED). It contains a subset of the United Nations Trade Data Element Directory (TDED);
- the UN/EDIFACT Code Lists (EDCL);
- the UN/EDIFACT Composite Data Element Directory (EDCD). It is composed of a list of the composite data elements with their component data elements;
- the UN/EDIFACT Segment Directory (EDSD), which is a list of all standard segments used in United Nations Standard Messages (UNSMs);
- the UN/EDIFACT United Nations Standard Message Directory (EDMD). It contains a full description of all UNSM types;
- Uniform Rules of Conduct for the Interchange of Trade Data by Teletransmission (UNCID). These rules are meant to provide a background for users of UN/EDIFACT and other systems of electronic trade data interchange; and
- Explanatory Material as appropriate.

3. EDIFACT Building Blocks

The building blocks of an EDIFACT message are:

1. data elements (DEs);
2. composite data elements (CDEs);
3. segments;
4. the structure of the message itself.

Figure 1 illustrates the hierarchical structure of a message.

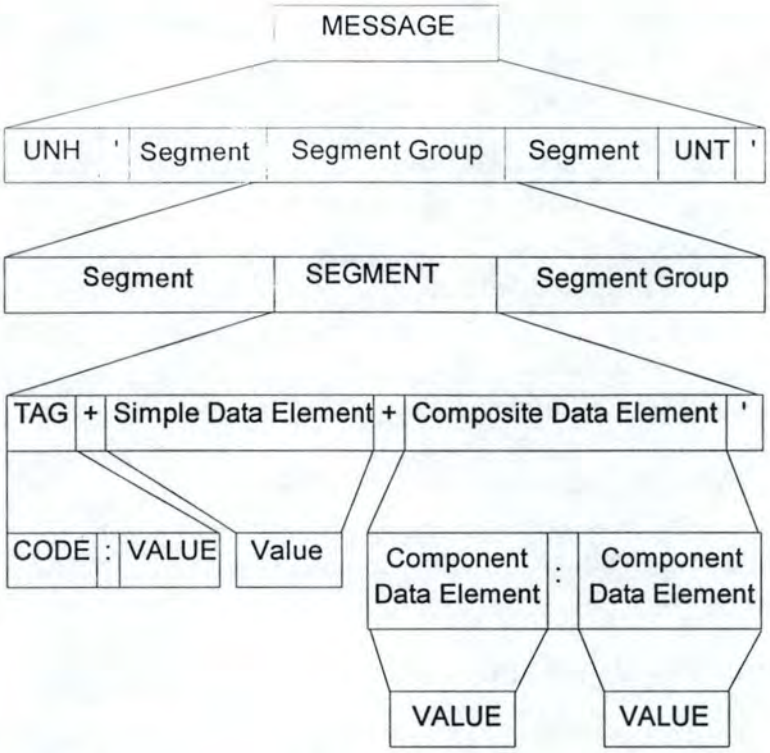


Figure 1: Hierarchical Structure of a Message [2]

3.1. Data Elements

"A data element is the smallest unit of information within the structure of a message ..." [2]

There are two types of data elements: simple data elements and component data elements. A simple data element can either be:

- a specific simple data element, which defines a precise business function;
- a generic simple data element, which defines a global business function. A data element qualifier is associated with it to give it a specific meaning;
- a data element qualifier, which gives a specific business function to the generic data element which it directly follows.[1]

[1]We would like to draw attention to the fact that a qualifier can qualify either:

- a generic data element. In this case, in a composite or in a segment, the qualifier follows directly the generic data element that it qualifies; or
- a whole composite. The *composite qualifier* is then the first component of the composite and qualifies all the other components of the composite; or
- a whole segment in the same way as a composite. This qualifier is called *segment qualifier*.

A component data element can be one of the three categories described above but is used in a composite data element.

Data elements can be expressed in clear or coded: in clear means that the value of the DE is explicitly, totally expressed while coded means that the DE has as its value a code, described in a code list directory.

Data elements are grouped into ten categories. The first figure of their four-numeric tag indicates the category to which they belong.

0 - Service Elements
1 - Documentation, references
2 - Dates, times
3 - Parties, addresses
4 - Clauses, conditions
5 - Amounts, percentages
6 - Measure identifiers
7 - Goods descriptions
8 - Transport modes
9 - Other

3.2. Composite Data Elements

"A composite data element is two or more component data elements grouped together to permit related information to be expressed in a structured way." [2]

"RULE 5: Composite data element shall have a single function, with each component data element relating directly to the function of the composite." [2]

When the first component data element of a composite is a qualifier, this qualifier, called *composite qualifier*, qualifies the whole composite. In other words it qualifies all the other data elements contained in the composite.

3.3. Segments

A segment is a defined set of simple and/or composite data elements. Segments are the essential level of grouping for UN/EDIFACT because to transmit data elements, they must be part of a segment.

"RULE 19: A new segment shall have a single function (which can be qualified if necessary, to identify its usage). A segment shall contain sufficient simple and/or composite data elements to fulfil its functional definition and the contents shall relate directly to the function of the segment." [2]

A segment can be qualified in the same way as a composite.

In a message, segments that are related can be grouped in segment groups.

"Grouping of segments also permits information carried in individual segments to be related in a structured way." [2]

"The first segment of a group must occur only once per occurrence of the group and is designated as the "trigger" segment for the group which it heads. The trigger segment determines the function of the group." [2]

In order to bring control in respect to the identification of what is being transmitted and from whom/to whom etc., user application data segments are enveloped by service segments. There are six service segments:

- UNB, interchange header identifying a number of functions or features which will apply to all the messages that follow (for example, the syntax identifier),
- UNZ, interchange trailer which ensures a tidy end to the interchange,
- UNG, functional group[1] header,
- UNE, functional group trailer,
- UNH, message header, whose function is to identify which message is to be processed and its version and release. It also contains the message reference number that identifies the message uniquely, and
- UNT, message trailer. It contains a count of the number of segments in the message and the reference number of the message transmitted in UNH.

3.4. Messages

A standard message is a structured collection of segments, forming a logical set of information, to be transmitted between two parties during an interchange. Each message fulfils a specific function. It may often be equivalent to paper documents such as order or invoice.

> *"RULE 32: A message is a set of ordered segments and/or segment groups, starting with the message header UNH segment, and ending with the message trailer UNT segment. At least one additional segment or segment group shall appear between the header and trailer segments." [2]*

Every message has a status:

- status 0: message under development;
- status 1: draft recommendation; or
- status 2: recommendation (available for operational use).

4. The EDIFACT Syntax

The requirements of the syntax are:

1. to be independent from:
   - the application;
   - the hardware; and
   - the means of communication;
2. to have as little effect as possible on the in-house systems;

[1]Functional groups are used to group several messages of the same type in an interchange. They are not used frequently.

3. to be as flexible as possible, balanced with efficiency.

The main features of the EDIFACT syntax are:

- hierarchical structuring: data is grouped hierarchically into larger and larger groups;
- implicit data element identification: each item of data is identified by its position in a message relative to a previous data item;
- special character separation;
- flexible length data structures; and
- mandatory/conditional status of elements and segments.

Table 1 shows the different notations of the EDIFACT syntax for data element format.

Format	Explanation
a3	3 alphabetic characters, fixed length
n6	6 numeric characters, fixed length
an5	5 alphanumeric characters, fixed length
a..6	up to 6 alphabetic characters
n..9	up to 9 numeric characters
an..35	up to 35 alphanumeric characters

Table 1: Data Element Format Notation

There are three character sets: levels A (simple), B (full), and C (Latin alphabet, Cyrillic, Greek). Associated with each level are specified control characters for use in data separation. Table 2 shows recommended control characters for level A.

Control Characters		Function
Apostrophe	'	Segment terminator
Plus Sign	+	Segment tag and data element terminator
Colon	:	Component data element terminator
Question Mark	?	Release character

Table 2: Level A Control Characters

Placing the release character before one of the other control characters indicates that the apostrophe, plus sign, colon or question mark is not a control character but part of the user data.

5. Interactive EDI

The majority of EDI performed today is in an environment that is based on one way automated transmission of data by the sender and where the syntax is structured for that purpose. This is called batch EDI. Messages are often quite long and timescales for transmission and processing are not of great significance.

As usage of EDI increases, users will wish to maximize on the techniques where possible and, thus, in certain instances, more dynamic processes that will fit more closely into the business environment. One such technique has been developed - Interactive EDI (I-EDI). I-EDI is based on controlled query and response messages.

"I-EDI is characterised by the following:

- *a formalised association between two parties using a dialogue;*
- *the ability, dynamically, to direct the course of an EDI transaction depending upon the result of earlier exchanges within the dialogue;*
- *short response times;*
- *all the messages in one dialogue relate to the same business transaction;*
- *a transaction is a controlled set of dialogues which can take place between two or more parties." [3]*

6. Glossary of Terms

ANSI.....American National Standard Institute
CDE.....Composite Data Element
DE.....Data Element
EDCD.....EDIFACT Composite Data Element Directory
EDCL.....EDIFACT Code lists
EDED.....EDIFACT Data Element Directory
EDI.....Electronic Data Interchange
EDIFACT.....Electronic Data Interchange For Administration, Commerce, and Transport
EDMD.....EDIFACT Standard Message Directory
EDSD.....EDIFACT Segment Directory
I-EDI.....Interactive Electronic Data Interchange
ISO.....International Organisation for Standardisation
TDED.....Trade Data Element Directory
TDID.....Trade Data Interchange Directory
UN/ECE.....United Nations Economic Commission for Europe

UN/EDIFACT..... United Nations Rules for Electronic Data Interchange For Administration, Commerce, and Transport
UN/GTDI ..... United Nations Guidelines for Trade Data Interchange
UNCID ..... Uniform Rules of Conduct for the Interchange of Trade Data by Teletransmission
UNSM ..... United Nations Standard Message

7. References

[1] SITPRO and PFA
The UN/EDIFACT Workshop
SITPRO and PFA, UK, August 1994

[2] UN/ECE
Design of UN/EDIFACT Messages: Guidelines and Rules
TRADE/WP.4/R.840/Rev.2
UN/ECE, Geneva (Switzerland), 11 January 1994

[3] UN/ECE
Working Draft - EDIFACT - Application level syntax rules - Part 3: Syntax rules specific to interactive EDI, plus interactive EDI service directories
UN/EDIFACT WD 9735-3:1994
UN/ECE, Geneva (Switzerland), 20 September 1994

2

Analysis of UN/EDIFACT Structures

1. Introduction

Before investigating the use of the object-oriented approach to EDIFACT, a study of EDIFACT structures must be undertaken. This chapter enumerating observations only, consists of three sections:

1. analysis of the composite structure,
2. analysis of the segment structure, and
3. analysis of the message structure.

The comments on EDIFACT structures that follow, should be read in conjunction with the appropriate 92.1 UN/EDIFACT directories [1], [2], [3].

2. Analysis of the Composite Structure

2.1. Introduction

This section analyses the 92.1 UN/EDIFACT Composite Data Element Directory [1].

It details five different reasons for using the composite structure existing in this directory and observations on the use of the composite structure are made.

2.2. The Use of the Composite Structure

A CDE can be qualified by introducing a qualifier as the first data element, which qualifies the CDE in its entirety. The qualifier is called a composite qualifier. The classification that follows was made **disregarding composite**

qualifiers, because the composite qualifier works at a higher level and it has the same effect on all composites.

In the 92.1 UN/EDIFACT Directory, the composite structure is used in the following cases:

1. to allow data to be expressed in two different ways: coded and/or in clear;
2. to repeat the same specific data element, in order to list data elements or to break down a large data element;
3. to associate a qualifier with a generic data element;
4. to associate the 1131 and 3055 data elements[1] with a data element (DE); and
5. to group together related data elements with or without qualifiers.

Each of the five reasons above will now be illustrated with an example of a CDE and some observations on the CDEs concerned. The CDEs concerned are listed in Appendix A.

2.2.1. To Allow Data to Be Expressed in Two Different Ways: Coded and/or in Clear

An example of the composite structure used to permit data to be expressed in coded and/or in clear, is C517 LOCATION IDENTIFICATION.

C517 LOCATION IDENTIFICATION
Desc: Identification of a location by code or name

3225 Place/Location identification	**C**	**an..25**
1131 Code list qualifier	C	an..3
3055 Code list responsible agency, coded	C	an..3
3224 Place/Location	**C**	**an..17**

Below are observations that apply to the CDEs concerned:

- for the CDEs C100, C110, C240, C242, and C273, the component in clear form appears twice; but C100 violates RULE 15b [4] by repeating the 4052 Terms of Delivery DE where the format is defined as an..70;
- 80 % of the coded DEs are supported by the use of 1131/3055 DEs;
- for nine of the CDEs, the coded DE is mandatory while the remaining components are conditional;
- the CDEs C522, C543, and C945 are qualified;
- while the tags of clear and coded components differ from each other by one unit, the tags of the components of C552 ALLOWANCE/CHARGE INFORMATION CDE are totally different: 1230 and 5189, nevertheless the

[1] 1131 and 3055 data elements are associated with a coded data element, which is not a qualifier, to identify the code list and the agency responsible for the code list.

CDE function is "Identification of allowance/charge information by number and/or by code". [1]

2.2.2. To Repeat the Same Specific Data Element

C058 is an example of CDEs that contain only a specific DE repeated several times.

C058 NAME AND ADDRESS
Desc: Unstructured name and address: one to five lines

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**

Below are some observations about the CDEs that contain only a specific DE repeated several times:

- the number of times a component DE is repeated varies between one and four (as allowed by RULE 14 [4]), except from the C210 CDE, where the component DE may appear up to ten times;
- C233 SERVICE identifies the requested services, priority and the cargo type concerned. This CDE contains a triplet of associated components that appears twice, which is in contradiction to RULE 6 [4]. However, this is acceptable since the triplet consists of one coded component associated with the 1131/3055 DEs;
- apart from C236, the first component of each CDE is mandatory.

2.2.3. To Associate a Qualifier with a Generic Data Element

C206 IDENTIFICATION NUMBER is an example of a CDE associating a qualifier with a generic data element.

C206 IDENTIFICATION NUMBER
Desc: Goods item identification number

7402 Identity number	**M**	**an..35**
7405 Identity number qualifier	**C**	**an..3**

The CDEs C186, C270, C507 and C526 are qualified CDEs.

C512 SIZE DETAILS is an unusual composite: it is composed of a composite qualifier followed by a single specific data element. We think that the structure of it is an anomaly because there is no clear reason why the qualifier is at the composite level and not at the data element level. This composite should be part of the list above.

2.2.4. To Associate the 1131 and 3055 DEs with a SDE

1131 Code list qualifier and 3055 Code list responsible agency, coded are associated with a coded DE which is not a qualifier to identify the code list and the agency responsible for the code list.

An example of a CDE associating the 1131 and 3055 DEs with a DE is C082 PARTY IDENTIFICATION DETAILS.

C082 PARTY IDENTIFICATION DETAILS
Desc: Identification of a transaction party by code

3039 Party id identification	**M**	**an..17**
1131 Code list qualifier	**C**	**an..3**
3055 Code list responsible agency, coded	**C**	**an..3**

The following observations apply to CDEs, which use 1131/3055 to define further the DE of the composite in question:

- for 78 % of the CDEs, the first component is mandatory, while for the rest (C218, C237, C528, C545, and C554), it is conditional;
- C545 is qualified in its entirety and the qualifier is mandatory.

2.2.5. To Group Together Related DEs in Order to Fulfil the Function Required

C509 PRICE INFORMATION is an example of CDEs grouping together DEs in order to fulfil the function required.

C509 PRICE INFORMATION
Desc: Identification of a price type, price and related details

5125 Price qualifier	M	an..3
5118 Price	**C**	**n..15**
5375 Price type, coded	**C**	**an..3**
5387 Price type qualifier	**C**	**an..3**
5284 Unit price basis	**C**	**n..9**
6411 Measure unit qualifier	**C**	**an..3**

It should be noted that:

- the structure of C080 PARTY NAME is of particular interest because the same DE is repeated four times and a further DE is added to define the format of this repeated DE;
- ten of these CDEs are qualified (C128, C174, C211, C280, C501, C504, C506, C509, C516, and C521);
- for C203, a triplet of associated components appears twice, which is in contradiction to RULE 6 [4]. This may be acceptable since the triplet consists of one coded component followed by the 1131/3055 DEs;
- in C402, a pair of components (7064/7143) is repeated. This violates RULE 6 [4].

2.3. Observations on the Use of the Composite Structure

- The 92.1 UN/EDIFACT Directory contains one hundred and twenty-one CDEs. Nineteen of them are qualified, inferring that the real function of one out of six CDEs can be determined by the use of a qualifier.

  "The use of qualified composites significantly reduces the number of entries in the Composite Data Elements Directory, and provides flexibility." [4]

- A CDE should have a single function. However there are examples of CDEs which can be considered as fulfilling two functions: e.g. C213 NUMBER *AND* TYPE OF PACKAGES.
- Another example of double functioned CDE is C078, which identifies an account holder and/or account holder name in one or two lines. What is then the reason for 6345 Currency, coded presence? C200 CHARGE is affected in the same way.
- A CDE should have a single function. Can this mean that a CDE should consist of components from the same class, or in other words, components whose tag begins with the same digit (without taking into account the 1131/3055 DEs and some DEs of class six which are measure identifiers)?
- The method used to give a format to a party name in C080 PARTY NAME is interesting. Can't it be applied to other composites, e.g. C058 NAME AND ADDRESS and then shorten the NAD segment?
- There might be overlapping between C878 CHARGE/ALLOWANCE ACCOUNT, C078 ACCOUNT IDENTIFICATION and C088 INSTITUTION IDENTIFICATION as the DEs of the two last CDEs appear in the first CDE mentioned.

3. Analysis of the Segment Structure

3.1. Introduction

The following section is concerned with the use of the segment structure in the 92.1 UN/EDIFACT Segment Directory.

In this paper, we class segments into four groups according to the use of the segment structure and some remarks are made. The lists if the corresponding segments can be found in Appendix A.

It should be noted that segments marked for deletion are not taken into account in our study.

3.2. The Use of the Segment Structure

The segment structure groups together composite data elements (CDEs) and/or simple data elements (SDEs) in order to fulfil the function of the segment.

"Segments can be made up of logically related simple data elements, composite data elements or a mixture of both. A simple segment has a single function." [5]

Thus, segments are classified as follows:

1. segments that contain a single CDE;
2. segments that contain SDEs only;
3. segments that contain one CDE and one SDE only; and
4. other segments.

As with composites, a segment can be qualified by a segment qualifier. For the same reason mentioned in the analysis of the composite structure, **segments are classified leaving aside the segment qualifier**. That means that the third class mentioned above contains segments consisting of one CDE and one SDE which is not a segment qualifier. If the only SDE was a segment qualifier, the segment would appear in the first class.

3.2.1. Segments that Contain a Single CDE

DTM is an example of a segment which is composed of one CDE only.

DTM DATE/TIME/PERIOD
Function: To specify date, and/or time, or period
C507 DATE/TIME/PERIOD M

85% of the segments concerned contain a CDE which is used only in that segment.

The following segments are qualified segments: DIM, EQA, REL, RNG and TMP.

3.2.2. Segments that Contain SDEs Only

DLI is an example of a segment that contains SDEs only.

DLI DOCUMENT LINE IDENTIFICATION
Function: To specify the processing mode of a specific line within a referenced document

1073 DOCUMENT LINE INDICATOR, CODED	**M**	**an..3**
1082 LINE ITEM NUMBER	**M**	**n..6**

It can be noted that ALI is composed of two SDEs followed by another one repeated five times.

It seems from analysis that whilst some message designers have decided to group SDEs into a composite and then place it into a segment (see above 3.2.1), others have opted to group SDEs without using a composite. Clearly, there is a difference in approach.

3.2.3. Segments that Contain one CDE and one SDE Only

The PGI PRODUCT GROUP INFORMATION segment is an example of a segment that contains one CDE and one SDE only.

PGI PRODUCT GROUP INFORMATION
Function: To indicate the type of price calculation or tariff used for products

5379 PRICE/TARIFF TYPE, CODED	**M**	**an..3**
C288 PRICING GROUP	**C**	

It should be noted again that the segments that contain a segment qualifier and one CDE are listed in 3.2.1.

3.2.4. Other Segments

The CUX CURRENCIES segment is an example of a segment that is composed of CDEs and SDEs.

CUX CURRENCIES
Function: To specify currencies used in the transaction and relevant details for the rate of exchange

C504 CURRENCY DETAILS	**C**	
C504 CURRENCY DETAILS	**C**	
5402 RATE OF EXCHANGE	**C**	**n..12**
6341 CURRENCY MARKET EXCHANGE, CODED	**C**	**an..3**

The following is observed:

- the segments ALC, ATT, BII, COT, EQD, FII, FTX, GIN, GIR, LOC, MEA, MEM, NAD, PAT, PIA and TDT are qualified;
- some of the segments above are complicated and contain several concepts to fulfil the function required.

  Examples: BUS (Business Function, Geographic Environment, Bank Operation,...), DGS (Hazard Code, Dangerous Goods Shipment Flashpoint, Packing Instructions,...), NAD (name, structured and unstructured address),...

3.3. Overall Observations on the Use of the Segment Structure

The overall observations are:

- twenty-two segments are functionally equivalent to CDEs because they are composed of that CDE only;
- six segments contain only SDEs where the structure is identical to the composite;
- from eighty-seven segments contained in the 92.1 directory, twenty-one of them are qualified segments;
- the same data can sometimes be mapped into different segments, e.g. "number of units" can be mapped into any one of the MEA (when the value of 6311

Measurement Application Qualifier is CT for Counts), QTY, or EQN segments. It happens, as well, with the data "temperature": MEA (TE) or TMP can be used.

4. Analysis of the Message Structure

4.1. Introduction

This section analyses the 92.1 UN/EDIFACT Message Directory, and more precisely, the EDIFACT message structure.

The 92.1 UN/EDIFACT Directory has fifty-two approved drafts or recommended messages, designed by several different groups. Although message design guidelines and rules exist, some groups have their own approaches to designing messages. Each may have its own interpretation of the rules and methods are influenced by the sector they work in, resulting in variances in message structures. The present section studies these variances.

To accomplish this task, messages are classified according to their source, which is observed as part of our analysis.

To summarise our study, first we look at the messages and their structure within each group; next, we analyse the variances in the usage of the most common segments and segment groups; finally, a conclusion about message structures is drawn. An evaluation of the study is the subject of the next chapter.

4.2. Message Classification

Initially, EDIFACT message classification, according to the message sources, is required.

Messages have been classified mainly by Message Development Group (MDG) of the Western European EDIFACT Board (WEEB) which was involved in their development, plus some messages developed by the Australia/New Zealand EDIFACT Board:

1. MD1 Trade,
2. MD2 Transport,
3. MD3 Customs,
4. MD4 Finance,
5. MD5 Construction, and
6. Australia/New Zealand EDIFACT Board.

The lists of the corresponding messages can be found in Appendix A.

4.3. Message Structures: Overview

In this section, we have a look at the messages within each class to find similarities and differences, taking into account their function and their structure.

4.3.1. MD1 Trade

In the data segment clarification of trade messages, it is specified, among other things, that the coded element should be used whenever it is possible and that free text information should be avoided as it inhibits automatic processing.

Trade messages are divided in two or three sections: the heading, detail and optionally, summary sections. Everything related to the whole message can be found in the first section. The detail section provides details of individual items. On the other hand, the summary section specifies the message control and hash totals.

According to their structures, trade messages can be classified in the following way:

- the ORDCHG, ORDERS, ORDRSP and INVOIC messages;
- the PRICAT and DESADV messages;
- the DELFOR and INVRPT messages;
- the QUOTES and REQOTE messages; and
- other trade messages (DELJIT, PARTIN, QALITY, REMADV, SLSRPT, STATAC).

♦ The ORDCHG, ORDERS, ORDRSP and INVOIC messages

The Purchase Order Change, Purchase Order and Purchase Order Response messages are structured in the same way. Since an invoice usually requires the same information, the INVOIC structure is very similar to the former three. These are large messages because much information about the items concerned is required and they are internationally defined to cater to the needs of many countries.

Many segment groups that appear in the heading section, appear also in the detail section because the default information, set in the heading section, can be different for a particular item, which then overrides the default information. Consequently, correspondence can be made between the detail and heading sections.

In INVOIC, for example, segment group 24 (SG24) and SG30 correspond respectively to SG8 (specifying the terms of payments) and SG6 (specifying tax related information, and when necessary, the location(s) to which that tax information relates). SG31 (identifying the parties) is very similar to SG2, to within a data element. SG14 and SG35 are similar since the function of SG35 is to specify allowances and charges for the line item where it is different from or not specified within the heading section. Nevertheless, it should be noted that SG44

(specifying any rules, laws or regulations with which the supplier must comply to meet requirements) belongs to SG35 (specifying charges and allowances) whilst SG21, which is the same as SG44, is, surprisingly, not part of SG14. The correspondence between the segments above is shown in Table 3.

Detail Section	Heading Section	Function
SG24	SG8	Specifying the terms of payments
SG30	SG6	Specifying tax related information, and when necessary, the location(s) to which that tax information relates
SG31	SG2	Identifying the parties
SG35	SG14	Specifying allowances and charges
SG44	SG21	Specifying any rules, laws or regulations with which the supplier must comply to meet requirements

Table 3: Correspondence Between Detail and Heading Sections in INVOIC

♦The PRICAT and DESADV messages

The Price/Sales Catalogue Message can be considered to be structured in three levels: information related to the whole catalogue, followed by information about a group of items and, within each group, line item information. This explains why some segment groups used in this message appear three times.

The Despatch Advice Message is designed in a similar way. Its structure is also top-down hierarchical.

♦The DELFOR and INVRPT messages

The messages, Delivery Schedule and Inventory Report messages, have the following in common: their detail section is long because it provides two different ways of transmitting the required information. In DELFOR, the information can either be location driven or product driven. The two solutions cannot be used simultaneously in the same message.

♦The QUOTES and REQOTE messages

The Quotation and Request for Quotation messages are similar to each other in the sense that they often use the same segment groups but the order of the groups differs a lot, even if they have the same function.

♦ Other trade messages (DELJIT, PARTIN, QALITY, REMADV, SLSRPT and STATAC)

The Statement of Account, Sales Data Report and Party Information Messages are three very simple and basic messages since not much information is required.

In SLSRPT, the two segment groups that specify reference and time, and the parties involved, are inverted when compared with the others.

4.3.2. MD2 Transport

One of transport messages is IFTMFR, the International Forwarding and Transport Message Framework. Its intention is not to be used like a message: it serves as a structured master for aligning a set of related message types within the transport application area. As it is a framework, the specification does not include a data segment clarification. The message framework structure only can be found.

Taking account of their structure, messages can be classed as follows:

- the messages whose structure is based on IFTMFR, the message framework; and
- messages whose structure is not based on IFTMFR.

Table 4 groups together the messages from MD2 in the two categories defined above.

Messages Derived from IFTMFR	Other Messages
IFTMAN	IFCSUM
IFTMBC	BAPLIE
IFTMBF	BAPLTE
IFTMBP	IFTSTA
IFTMCS	
IFTMIN	

Table 4: Categories of Transport Messages

♦Transport messages derived from IFTMFR

As a message framework, it allows many messages to conform to a generic structure. Most of these messages are a subset of IFTMFR, sometimes with added segments or segment groups which do not appear in the framework.

It should be noted that the framework has now been deleted from the current UN/EDIFACT directory. The reason for withdrawing the framework was due to its maintenance overhead. A change to one message would always result in a change to IFTMFR.

♦Transport messages whose structure is not based on IFTMFR

IFCSUM, International Forwarding and Consolidation Summary Message, is structured in exactly the same way as IFTMFR: there is a total correspondence between their segment groups. However, its functionality differs: IFCSUM provides a statement for a means of transport or equipment and their summary type information regarding the consignment carried.

The three other messages, Bayplan/Stowage Plan - Occupied and Empty Locations, - Total Numbers, and International Multimodal Status Report messages, are totally different from the previous ones: they are shorter and they are not derived from the message framework since the framework is only for the transport of goods by a carrier (booking, confirming).

4.3.3. MD3 Customs

The Customs Conveyance Report, Customs Cargo Report, and Customs Response Messages have only a detail section. At the end of these messages, the AUT segment specifies results of the application of an authentication procedure.

The Customs Declaration Message is structured in three sections:

- the heading section, which sets the message context;
- the detail section, which gives information pertaining to commercial documentation and customs requirements for the clearance of goods. (This consists of a commercial part first and then detailed information for a single customs item);
- finally, the summary section.

4.3.4. MD4 Finance

Financial messages can be categorised as follows:

- the BANSTA message;
- the PAYMUL message;
- the DOCADV, DOCAPP and DOCINF messages; and
- the CREADV, CREEXT, PAYORD, PAYEXT, DEBADV, REMADV and DIRDEB messages.

♦ The BANSTA message

BANSTA, the Banking Status Message, is used to report on the status of processing of a previous message. Consequently, it is not a true financial message as such, it is more a management tool. BUS, the Business Function Segment that provides information to the processing and purpose of a financial message, is not a part of it.

♦ The PAYMUL message

The PAYMUL message, the Multiple Payment Order Message, is a particular type of financial message. It is structured in three levels:

- A level, which contains data related to the whole message;
- B level, which contains data related to debit side and data which applies to all further details of C level; and
- C level, which contains mainly data related to the credit side, and this data is considered as unique for each payment transaction identified in level B.

♦ The DOCADV, DOCAPP and DOCINF messages

The Document Credit Advice, Application and Issuance Information messages are very similar. As with trade messages, it is recommended to use coded elements instead of clear elements, and to avoid free text information. As financial messages, BUS appears in the heading section to qualify the financial function of the message.

♦ The CREADV, CREEXT, PAYORD, PAYEXT, DEBADV, REMADV and DIRDEB messages

There is a close relationship between the structures of the CREADV, CREEXT, PAYORD, PAYEXT, DEBADV and DIRDEB messages. DIRDEB and PAYEXT have the same structure. There is a small difference between CREADV and DEBADV structures, where a SG appearing in DEBADV, indicates the method of payment used by the payer's bank to carry out the financial transaction.

CREEXT and PAYEXT are extensions respectively of CREADV and PAYORD. Besides their basic function, CREEXT and PAYEXT provide full details of the transactions to which the messages relate. This explains why the CREADV and PAYORD messages are subsets respectively of CREEXT and PAYEXT. This is also the reason why the REMADV structure is included in the two extended messages.

CREADV, DEBADV and PAYORD are the only financial messages that consist solely of a detail section only.

4.3.5. MD5 Construction

Construction messages can be classified as follows:

- messages exchanged during the contract establishing process, and that transmit bills of quantities; and
- those relating to the work progress or payment.

♦ The CONITT, CONTEN, and CONEST Messages

The three following messages, the Invitation to Tender, Tender and Establishment of Contract Messages, all replace the bill of quantities, but at a different stage of the whole process of establishing a contract. That is why they share the same message structure.

The structuring of the bill of quantities in these messages is based on a data approach:

> *"The actual appearance of the message bears no relation to the appearance of the current paper documen. ... The CONITT appearance is similar to the results of data analysis exercise. Namely that there are four blocks or groupings of data,*

each of which can appear many times and that all of the mechanism for linking and relating the components together are data elements within these blocks." [3]

The advantages of this approach are:

- it is less rigid than the existing document;
- it delivers a variety of new opportunities to its users; and
- it offers a solution to the volume problem. [3]

These are quite long and complicated messages since they must provide much information about each work item. The bill of quantities is spread into the heading and detail sections: while the heading section consists of information about groups of items of a project, the detail section provides information about individual work items.

♦The CONPVA, CONDPV, and CONQVA Messages

The Payment Valuation Message provides detailed payment information. It is the largest message of these three. The heading section refers to groups of items and the detail section relates to the items, just as for the messages above. The Direct Payment Valuation Message, which communicates the value of progress against groups of work items, is a subset of CONPVA heading section. The CONQVA detail section, which reports the progress quantities against the work items, is a subset of CONPVA detail section.

4.3.6. Australia/New Zealand EDIFACT Board

The messages developed and maintained by the Australia/New Zealand EDIFACT Board are short, simple and easy to understand.

The three messages, Payroll Deduction Advice, Superannuation Maintenance and Superannuation Contributions Advice messages, are structured in three sections (heading, detail and summary sections), while the Passenger List Message has only one section (detail section).

4.4. Message Structuring into Sections

As specified in Message Design Guidelines and Rules [4], each message begins and ends respectively with the UNH and UNT service segments. The type and sequence of segments appearing in between depend on the message type. Nevertheless, the initial sequence of segments set the message context, in relation to the entire message (e.g. DTM to give dates related to the whole message, a SG including NAD to identify parties involved, etc.). What follows in detail is the body of the message. The summary details often include segments of control or some that summarise the information transmitted (such as hash totals).

In their definitions, messages are divided in heading, detail and summary sections. The three structures of the messages of the 92.1 UN/EDIFACT Directory are either:

- a detail section only containing the whole message;
- a heading and a detail section; or
- a heading, a detail and a summary section.

As for the latter two cases, the limits of the heading and the detail sections vary. On one hand, the heading section can set the context of the message, while the detail section consists of the real message body. On the other hand, in addition to the context setting information, the heading section can provide information about groups of items, whilst the detail section gives details about individual items.

The summary section is usually used for controls and totals for line item details. When this section does not exist, it is included in the detail section.

Below is a table of all messages in the 92.1 directory. They are still grouped according to their source. It shows information for each of them, about the number of sections the message is divided into and when the UNS segment is used between the sections. The number of sections appearing in Table 5 is:

- 1 for a detail section only;
- 2 for a heading and a detail section; and
- 3 for a heading, a detail and a summary section.

	Number of sections	Use of UNS between the heading and detail sections	Use of UNS between the detail and summary sections
DELFOR	3	*	*
DELJIT	2		
DESADV	3		
INVOIC	3		*
INVRPT	2		
ORDCHG	3		*
ORDERS	3		*
ORDRSP	3		*
PARTIN	2	*	
PRICAT	2		
QALITY	2		
REMADV	3		*
SLSRPT	2		
STATAC	3		*
BAPLIE	1		
BAPLTE	1		
IFCSUM	1		
IFTMAN	1		
IFTMBC	1		
IFTMBF	1		
IFTMBP	1		
IFTMCS	1		
IFTMFR	1		
IFTMIN	1		
IFTSTA	1		

CUSCAR	1		
CUSREP	1		
BANSTA	3		
CREADV	1		
CREEXT	3		*
DEBADV	1		
DIRDEB	3		*
PAYEXT	3		*
PAYMUL	2		
PAYORD	1		
CONDPV	1		
CONEST	3		
CONITT	3		
CONPVA	3		
CONQVA	3		*
CONTEN	3	*	
CUSDEC	3	*	*
CUSRES	1		
DOCADV	3		
DOCAPP	3		
DOCINF	3		
PAXLIST	1		
PAYDUC	3		*
SUPCOT	3	*	*
SUPMAN	3	*	
QUOTES	3		*
REQOTE	3		*

Table 5: Structuring Messages into Sections

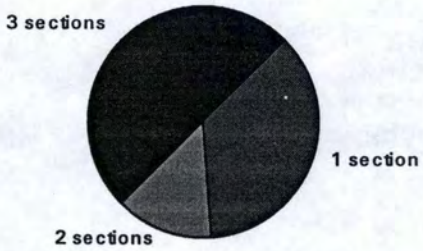


Figure 2: Proportion of Messages Structured into 1, 2 and 3 Sections

Half of the messages have three sections, while a minority of them have a single section. It is not clear why one or another structuring is used, except for MD2 messages that all consist of a single detail section.

The UNS service segment is used to separate the sections defined above and has been used to prevent the collision between segments [4], although this was not the original intention.

4.5. Segment and Segment Group Usage

A closer look at the segment and segment group usage is required for further analysis.

This is achieved in the following way, through an analysis of the usage of common segments and segment groups in messages:

- first, the segment frequency usage is investigated, and this is illustrated in a list of the most often used segments;
- then, the usage of these segments and the ways they are grouped together in SGs are analysed.

4.5.1. Most Common Segments

The first step is to find which segments are the most common.

Table 6 shows the number of messages (first column) that use a segment (second column). The third column indicates the total number of segments[1]. A missing row indicates that no segments correspond to the number missing. For example, the third row should be read like this: ATT appears in three messages, as does SEQ, but not necessarily in the same messages.

Number of Messages	Segments	Total Segment Number
1	CCI, CST, DMS, ERP, MEM, PGI, PIT, PSD, REL, SPS, STA, STS, TEM	13
2	CNI, COT, CPS, DLM, EMP, ERC, PRC	7
3	ATT, SEQ	2
4		
5	ARD, DLI, IND	3
6	AGR, AJT, BII, TCC	4
7	GOR	1
8	CPI, EQA, SCC, TMD	4
9	QVR, SEL, SGP	3
10	FCA, GDS, GIR, INP, TMP, TPL	6
11	BUS, DGS, EQN, GIN	4
12	APR, GID, RCS, RTE, TSR	5
14	DIM	1
15	PAT	1
17	HAN, PAC	2
18	ALC, ALI, TOD	3
19	PAI, UNS	1
21	EQD, RNG, TAX	3

[1]The total segment number does not take into account service segments.

22	GIR, GIS, IMD, LIN, PCI, PIA	6
24	PRI	1
25	AUT	1
26	PCD	1
27	FII	1
28	TDT	1
29	CNT, MEA	2
30	QTY	1
38	CUX	1
39	LOC	1
40	DOC, MOA	2
48	COM, CTA	2
50	FTX	1
52	BGM, DTM, NAD, RFF, UNH, UNT	4

Table 6: Segment Usage Frequency

An alternative representation of the table above is Figure 3. The X-axis is the number of different messages while the Y-axis indicates the total segment number.

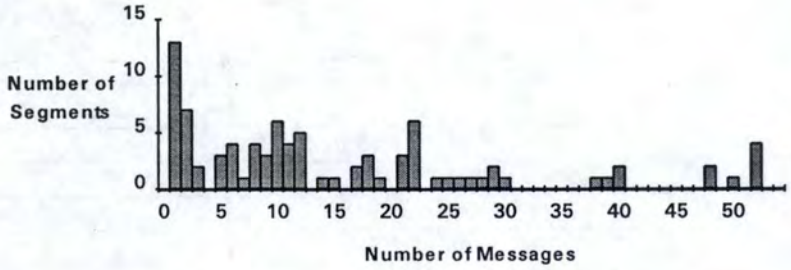


Figure 3: Segment Usage Frequency

We can see that more than half of the segments each appear in less than twelve different messages. This implies that many segments have particular functions to match particular message functions and are not generic to most messages.

With the exception of the UNH and UNT service segments, which are mandatory, four segments appear in all messages: they are BGM[1], DTM, NAD, and RFF.

[1]It can be noted that BGM BEGINNING OF MESSAGE is badly named; its name should be related to the function fulfilled by the segment, which is to indicate the type and function of a message and to transmit the identifying number. Therefore, it could be something like DOCUMENT/MESSAGE IDENTIFICATION.

The COM and CTA segments are frequently used and always appear together in a message.

The table above indicates the segments that are the most commonly used in all messages. These are, in increasing order:

1. CUX CURRENCIES,
2. LOC PLACE/LOCATION IDENTIFICATION,
3. DOC DOCUMENT/MESSAGE DETAILS,
4. MOA MONETARY AMOUNT,
5. COM COMMUNICATION CONTACT,
6. CTA CONTACT INFORMATION,
7. FTX FREE TEXT,
8. BGM BEGINNING OF MESSAGE,
9. DTM DATE/TIME/PERIOD,
10. NAD NAME AND ADDRESS, and
11. RFF REFERENCE.

We can have a look at the segment usage frequency within certain messages. Table 7 shows this for the most used messages, namely DESADV, INVOIC, ORDERS, ORDRSP, QUOTES, SLSRPT.

	CUX	LOC	DOC	MOA	COM	CTA	FTX	BGM	DTM	NAD	RFF
DESADV	0	4	0	0	1	1	7	1	10	2	3
INVOIC	2	8	2	12	2	2	4	1	21	2	10
ORDERS	2	9	3	11	2	2	8	1	26	2	11
ORDRSP	2	11	3	11	2	2	8	1	26	2	11
QUOTES	1	7	2	12	2	2	8	1	24	2	6
SLSRPT	1	1	1	1	1	1	0	1	4	2	2
TOTAL	*8*	*40*	*11*	*47*	*10*	*10*	*35*	*6*	*111*	*12*	*43*

Table 7: Frequency of Most Used Segments in Most Used Messages

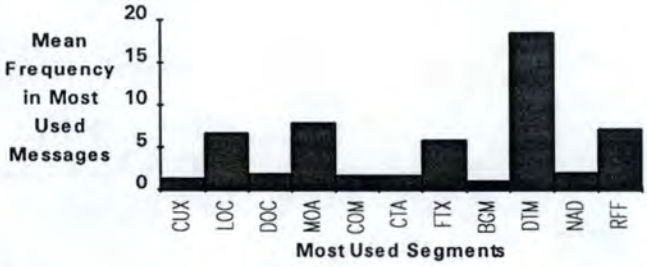


Figure 4: Mean Frequency of Most Used Segments in Most Used Messages

Table 7 and Figure 4 show that DTM is the most used segment in a message. The order message uses it up to twenty-six times. CUX, DOC, COM,

CTA, BGM, and NAD appear once to twice, while LOC, MOA, FTX, and RFF can appear six to eight times in a message. These are average figures.

What follows is an analysis of how these more common segments are used.

4.5.2. Analysis of the Usage of the Most Common Segments and Segment Groups

Within each class defined in point 4.2., we examine the use of segments and segment groups.

First, the following segment groups are used frequently in messages with the same purpose:

- [RFF-DTM]: SG giving references and date/time;
- [CTA-COM]: SG identifying the person, function or a department and appropriate numbers to whom communication should be directed;
- [CUX-DTM]: SG providing information about currencies;
- [DOC-DTM]: SG indicating documentary requirements;
- [TOD-LOC]: SG indicating the terms of delivery;
- [LOC-DTM]: SG giving location information;
- [TAX-MOA-LOC]: SG providing tax related information.

The DTM and LOC segments are very frequently used and can be associated to any other segment or SG that needs date or location information.

♦MD1 Trade

All trade messages begin with the UNH, BGM and DTM segments. BGM is used to identify uniquely the message narrower than UNH. DTM specifies date/time/period related to the whole message.

The first segment group is [RFF-DTM]. It specifies references (which cannot be defined in BGM, such as message reference number) and dates related to the entire message, except for DELJIT where it is related to the specified parties although no party is specified before it. This SG is not part of REMADV.

The second SG specifies the parties involved. Its form varies: it can be simple as [NAD-[CTA-COM]] in PARTIN, REMADV, SLSRPT and STATAC, or e.g. more complex in the form [NAD-LOC-[RFF-DTM]-[DOC-DTM]-[CTA-COM]]. In QUOTES and REQOTE, FII, which is optional, follows the LOC segment.

It can be noted that in SLSRPT, the order of the two segment groups [RFF-DTM] and [NAD-[CTA-COM]] is reversed.

In REMADV, SLSRPT, STATAC, INVOIC and order messages, DOC is used to identify and provide information relating to a document. But it is not often

used. In all other cases and other messages, RFF is used to give any reference, even document ones.

The [TAX-MOA] SG is common to specify tax rules. LOC can be included in this SG to indicate the location to which the tax applies. [MOA-CUX], [CUX-DTM] are often used.

The FTX segment often appears but as has already been mentioned, in the segment clarification, it is recommended to use it when no other approach is feasible.

In INVOIC and similar messages, there is a SG consisting of [RCS-RFF-DTM-FTX]. The DTM segment, by implication, should pertain to RCS, but it does not. It would be then appropriate to group together RFF and DTM in a SG.

♦MD2 Transport

In the beginning of transport messages the following segments can be found: UNH-BGM-DTM, but their functions vary. BGM identifies message number and function of message. In IFTMBC, DTM indicates date/time/period relevant to the whole message, whilst in BAPLIE and in BAPLTE, it specifies the date of the preparation of the message.

To specify parties, [NAD-LOC-...-[CTA-COM]..] is used. However, [CTA-COM] can be used on its own to indicate relevant contacts. As NAD, it can be used alone, or associated to DTM, to specify an address of a place of collection or delivery.

The [DOC-DTM] SG indicates information about documents. Otherwise, RFF or [RFF-DTM] is used.

It seems that in these messages, the FTX segment is an integral part of them. It is not like in trade messages where it is asked to avoid free text information. Here, FTX has a real function.

♦MD3 Customs

The CUSCAR, CUSDEC, CUSREP, and CUSRES messages begin with UNH-BGM-DTM. DTM may contain the date of message creation.

The SG that identifies parties varies: in CUSCAR and CUSREP, it is [NAD-[CTA-COM]], in CUSDEC, it is [NAD-RFF-CTA-COM], and in CUSRES, NAD alone specifies the parties.

LOC is frequently used to indicate locations/places relevant to the message.

The DOC segment appears to specify documentary requirements.

The use of FTX seems necessary as in transport messages.

♦MD4 Finance

The first segments of the majority of financial messages are as follows: UNH-BGM-DTM-BUS. The DTM segment aims to give the time when the message was created as well as other times relevant to the message. The BUS segment identifies the message function. It can also provide information either to route the message, or for tariffing or for the provision of some statistical information. BANSTA does not contain the BUS segment.

In the BANSTA message, the BGM and DTM segments related to the whole message, are grouped together in a mandatory SG, repeating only once, for unique identification of the message, the date and time, the type of BANSTA message, and its function.

Where financial institutions are concerned, there are two different manners to specify the parties involved: [FII-CTA-COM] for financial institutions and [NAD-CTA-COM] in other cases. Organised in this way, we would say that COM refers to the trigger segment. However, for PAYMUL, as written in the segment clarification, it refers to CTA. The use of a SG like [CTA-COM] would be more accurate.

We can also note that NAD sometimes is used individually to identify a party name and address.

Other frequent SGs are [MOA-CUX-DTM-RFF] to specify an amount, [TAX-MOA-CUX-DTM] to indicate tax rules and [DOC-DTM] to identify a document.

In these messages, FTX sometimes has a real function and free text information cannot be avoided.

For DOCADV, DOCAPP and DOCINF, it is slightly different. Their first segments are UNH-BGM-BUS-INP-FTX. BUS provides information related to the type of the message and it may also identify its geographic environment. The SG that identifies the parties is [NAD-RFF-CTA-COM], and for financial institutions, it is [FII-RFF-CTA-COM]. COM is related to CTA. Another one used to identify parties and other related documentary requirements associated with the whole message is [NAD-[DOC-DTM-LOC]]. In these latter messages, it is mentioned that FTX should be for free text or uncoded form only.

♦MD5 Construction

Construction messages begin with the usual UNH-BGM-DTM segments. Then follows AUT which transmits a password or some other form of identification agreed between the trading partners.

The first SG is [RFF-DTM]. Although it should give references applicable to the whole message, in CONPVA, it refers to the specified party.

The segment that identifies parties is [NAD-LOC-FII-[RFF-DTM]-[DOC-DTM]-[CTA-COM]]: [RFF-DTM] gives references relevant to the specified party and [DOC-DTM] identifies the documentary requirements of the specified party. The DOC segment is only used in the SG triggered by NAD. That means that whenever a reference is needed RFF is used.

The SG to give information about tax rules is [TAX-MOA]. Sometimes LOC is included in this SG.

In CONITT and CONTEN, the SG giving information about currencies can be [CUX-DTM-FTX] where FTX contains any narrative necessary to explain or qualify the currency information.

As for FTX, it is mentioned that any free text information will require this segment to be processed manually though it should be used when necessary.

♦Australia/New Zealand EDIFACT Board

In the beginning of the messages elaborated by the Australia/New Zealand EDIFACT Board, the common segments found are: UNH-BGM-DTM-RFF...

In PAXLST, [NAD-CTA-COM] is the SG that identifies the parties involved while in the three other messages, it is [NAD-[CTA-COM]].

DOC is used to identify documents.

FTX can contain clear or coded information.

4.6. Conclusion

We can conclude that message structures vary a lot, not only between groups but also within groups.

Although very different structures exist, some message types can be used to fulfil several functions. The majority of transport messages are based on a message framework. Some financial messages are closely related and are subsets of others. Some construction messages share the same structure.

In addition, there is no real common rule for message structuring into sections, nor for the usage of segments or the grouping of segments.

The next chapter evaluates these structures and defines the expectations of a new approach to designing messages.

5. Glossary of Terms

CDE.....Composite Data Element
DE .....Data Element
EDIFACT.....Electronic Data Interchange For Administration, Commerce and Transport
MDG .....Message Development Group
PAEB .....Pan American EDIFACT Board
SDE.....Simple Data Element
SG.....Segment Group
UN/ECE .....United Nations Economic Commission for Europe
UN/EDIFACT.....United Nations Rules for Electronic Data Interchange For Administration, Commerce and Transport
WEEB .....Western European EDIFACT Board

6. References

[1] UN/ECE
The 92.1 UN/EDIFACT Composite Data Element Directory
UN/ECE, Geneva (Switzerland), 1992

[2] UN/ECE
The 92.1 UN/EDIFACT Segment Directory
UN/ECE, Geneva (Switzerland), 1992

[3] UN/ECE
The 92.1 UN/EDIFACT Message Directory
UN/ECE, Geneva (Switzerland), 1992

[4] UN/ECE
Design of UN/EDIFACT Messages: Guidelines and Rules
TRADE/WP.4/R.840/Rev.2
UN/ECE, Geneva (Switzerland), 11 January 1994

[5] SITPRO and PFA
The UN/EDIFACT Workshop
SITPRO and PFA, UK, August 1994

3

Evaluation of UN/EDIFACT Structures

1. Introduction

This chapter presents an evaluation of EDIFACT structures, referring to the previous chapter analysing 92.1 UN/EDIFACT directories [1], [2] and [3]. It highlights problems that are brought about by current structures. These problems should be taken into account when investigating the use of the object-oriented approach to designing messages, which constitutes the next step to be undertaken.

The composite structure is first evaluated, followed by the segment structure, and finally, the message structure. Although this evaluation is threefold, each structure is closely related to the others. At the end of the chapter, a short conclusion is drawn.

2. Evaluation of the Composite Structure

This section evaluates the composite structure. The following subjects are discussed:

1. the fact that the composite structure is used for a variety of reasons and covers different levels of information;
2. the definition of the composite as a single function element; and
3. other inconsistencies related to the component status, name and tag numbering.

2.1. Multiple Uses of the Composite Structure

The composite structure is currently used for many different reasons and covers different levels of information.[1]

In order to show this, EDIFACT elements can be redefined according to their function.

Three basic elements can be defined as:

- simple data elements, as defined in EDIFACT;
- [1131/3055] where 1131 and 3055 are data elements; and
- qualifiers, as defined in EDIFACT.

There are four different levels of information:

1. level 1
   Level 1 consists of the smallest units of information.
   Elements of this first level are either:
   - a simple data element;
   - a simple data element combined with [1131/3055], where the latter component [1131/3055] does not add anything new, but merely helps to decode the simple data element; or
   - a repeating simple data element, which as a whole, provides a unit of information.

2. level 2
   Level 2 data elements are composed of data units to which precision is given. Essentially, it is a level 1 data element which is qualified.

3. level 3
   The purpose of level 3 is to group together related data elements to fulfil a function. An element of this level consists of a group of a minimum of two elements either levels 1 and/or 2.

4. level 4
   A level 4 data element is a level 3 data element qualified in its entirety.

Table 8 summarises the four levels of information.

Basic Elements:

- **simple data element,**
- **[1131/3055],**
- **qualifier.**

[1] Here we do not take into account C512 SIZE DETAILS, composed of a composite qualifier and a simple data element, as we consider it to be an anomaly in the EDIFACT composite data element directory.

Levels	Function	Component Elements
level 1	1) to provide a unit of information	• a simple data element • a simple data element and [1131/3055] • a repeating simple data element
level 2	2) to give precision to a unit of information	{1} and qualifier
level 3	3) to put together related data elements (qualified or not) in order to fulfil a function	{1}(s) and/or {2}(s) (minimum two)
level 4	4) to give precision to the function of a group of related data elements	qualifier and {3}

Table 8: Four Levels of Structuring Information

If we look again at the different reasons for using the composite structure, mentioned in the analysis of the composite directory, we can see that the composite structure covers all four of the levels defined above:

- the composite data elements (CDEs) which associate only the 1131 and 3055 data elements (DEs) with a DE and the CDEs that contain a repeated data element correspond to level 1;
- the CDEs that associate a qualifier with a generic DE correspond to level 2;
- the CDEs that allow data to be expressed in clear and in coded and the CDEs that group together related DEs in order to fulfil the function required correspond to level 3;
- qualified CDEs that group together related DEs in order to fulfil a function, correspond to level 4.

Moreover, level 4 is a logical qualification of structured level 3. However, in the composite data element directory, there are examples of qualified composites that can be defined as level 1 or 2.

We see that a single structure, namely the composite structure, covers four different information levels. It can lead to confusion and ambiguity.

2.2. Multifunctional CDEs

It is ruled that a CDE should have a single function. This may lead to ambiguity in the use of the composite structure.

> *"RULE 5: Composite data element shall have a single function, with each component data element relating directly to the function of the composite." [4]*

If a composite must have a single function, can it consist of several different concepts? This question is closely related to the point raised previously regarding the different information levels.

The problem lies in defining *function*.

Does this mean that components must be of the same class, except components of class six that are measure identifiers and do not introduce a new concept. It also implies that composite qualifiers must not introduce a new concept too.

This problem will be discussed further in the evaluation of segment structures, since the border between composites and segments is not always clear.

2.3. Other problems

During the analysis, some inconsistencies were highlighted. However, these are not really related to the structure of the composite. They are concerned with the component status, name and tag numbering. The same kinds of inconsistencies can also be found in segments.

♦ **Inconsistency in the Component Status**

The way in which the status of component data elements in a CDE is attributed is not always logical.

In CDEs which are composed of a data element repeated several times, the first component is either mandatory or optional. Logically, the first component should always be mandatory since the first DE automatically has a value. (This is on the assumption that a specific data function has not been allocated a specific position.)

♦ **Inconsistency in the Component Name**

The name of a DE does not always appear to be appropriate, e.g. in C512 SIZE DETAILS, the name of the qualifier of this CDE should be "Size details qualifier" instead of "Size qualifier" because it implies that it qualifies the component that follows. Moreover, that qualifier only appears in C512.

♦ **Inconsistency in the Component Tag Numbering**

The tags of clear and coded components usually differ by one unit. However, the tags of the components of C552 are totally different, namely 1230 and 5189, although the function of this segment is to identify the allowance/charge information by number and/or by code.

2.4. Conclusion

The problem of the composite structure is the inconsistency in its use, which is related to:

- the different levels of information the structure covers; and
- the definition of a composite as having a single *function*.

When the composite structure is used will have to be defined clearly, without ambiguity.

3. Evaluation of the Segment Structure

Below, the segment structure is evaluated. The following points are discussed:

1. the fact that it is not always clear when the composite or segment structure should be used;
2. the multiple uses of the structure in question; and
3. the complexity of some segments.

3.1. Overlapping in Usage of Composite and Segment Structures

Firstly, the difference between the composite and segment structures is not always clear since segments can contain a single CDE and also simple data elements (SDEs) only.

A priori, a segment that contains a single CDE, could be made up of a CDE from any of the five classes defined in the brief analysis of the composite data element directory. However, CDEs that are composed of a repeated DE are in fact not used in this way.

Segments that contain SDEs only are made up of specific DEs only. Consequently, no DE qualifier or 1131/3055 DEs appear in these segments as these constitute a reason for using the composite structure. These segments correspond to a subset of the composites from the fifth class that are composed of several related DEs to fulfil a function.

Below is the AUT AUTHENTICATION RESULT segment. It comprises two DEs from the same class. According to the five reasons for using the composite structure, these two related DEs constitute the fifth one. This begs the question: why are these DEs grouped in a segment and not in a composite?

AUT AUTHENTICATION RESULT
Function: To specify results of the application of an authentication procedure.

9280	VALIDATION RESULT	M	an..35
9282	VALIDATION KEY IDENTIFICATION	C	an..35

Strict rules should define when the composite or the segment structure should be used. There should be no ambiguity.

3.2. Multiple Uses of the Segment Structure

The segment structure is affected by a similar problem of different information levels as the composite structure.

The two component elements of a segment are the specific data element and the composite data element. A segment can contain either specific DEs, or

CDE(s), or a mixture of both, in order to fulfil a function. In addition, the whole can be qualified to give precision to the function.

Figure 5 shows how the component elements of a segment are put together. Here, a DE is an EDIFACT specific data element. It belongs to the first level defined in the evaluation of the composite structure. A CDE can be any of CDEs, except those that are composed of a repeating DE. It can be from any level defined in the previous point. A segment is represented by any of the shaded rectangles.

There are three levels:

1. the first one, where the component elements of a segment can be found;
2. the second one, where these elements are grouped together to fulfil a function; and
3. the third one, where precision is given to the function of a group by using a qualifier.

It should be noted that the first level here covers all four levels detailed in the evaluation of the composite structure.

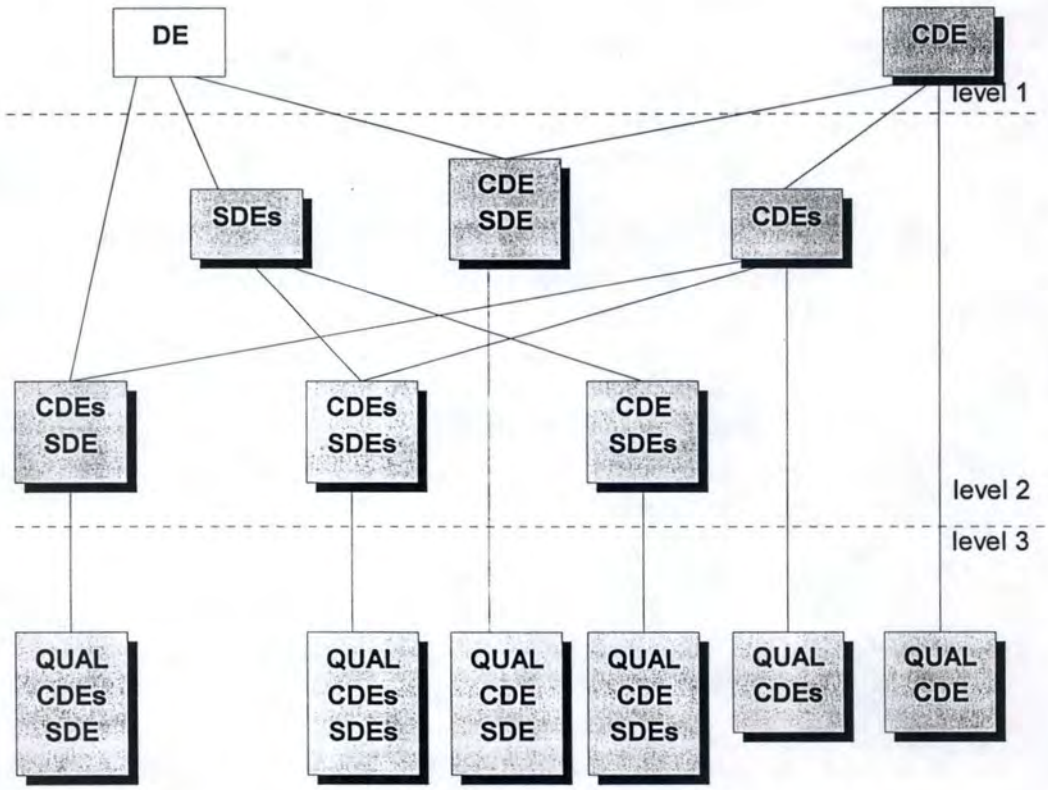


Figure 5: Segment Structure

The chart above shows that the segment structure covers all three of these levels. Moreover, a segment can contain a single CDE, implying that the segment structure also covers all four levels defined for the composite structure.

3.3. Complexity of Segments

As mentioned in the analysis of the segment directory, some segments are complex and may contain several concepts to fulfil the function required. TDT is an example.

TDT DETAILS OF TRANSPORT

Function: To specify the transport details such as mode of transport, means of transport, its conveyance reference number and the identification of the means of transport. The segment may be pointed to by the TPL segment.

8051	TRANSPORT STAGE QUALIFIER	M	an..3
8028	CONVEYANCE REFERENCE NUMBER	C	an..17
C220	MODE OF TRANSPORT	C	
C228	TRANSPORT MEANS	C	
C040	CARRIER	C	
8101	TRANSIT DIRECTION	C	an..3
C401	EXCESS TRANSPORTATION INFORMATION	C	
C222	TRANSPORT IDENTIFICATION	C	

Compared to AUT that is composed of two specific DEs, this segment is complex: it contains five CDEs and four specific DEs. All of them provide details of transport, but they are different concepts (carrier, mode of transport, means of transport,..). Moreover, the conveyance reference number is a duplication of RFF.

Here, again, rules are required to avoid too complex segments and to define clearly how segments should be constructed.

3.4. Conclusion

The reasons for using a segment are many and when a segment should be used instead of a composite one is not clear:

- some segments are structured in the same way as composites;
- segments cover many different levels of information; and
- some segments are simple whilst others are fairly complex containing several concepts.

The rationale of these inconsistencies and ambiguities is that both composites and segments are defined as having *a single function.*

> *"RULE 5: Composite data element shall have a single function, with each component data element relating directly to the function of the composite." [4]*

> *"RULE 19: A new segment shall have a single function (which can be qualified if necessary, to identify its usage). A segment shall contain sufficient simple and/or*

composite data elements to fulfil its functional definition and the contents shall relate directly to the function of the segment." [4]

The problem lies in the definition of the word *function*. To date, this has been resolved in a very subjective manner which perhaps explains the inconsistencies observed so far.

It is obvious that if *function* has the same meaning in both cases, there is no point in using a composite structure. A reasoning behind the composite structure is best explained by the compression and omission techniques used in EDIFACT. Data elements are positional, therefore, if a set of stand alone simple data elements in a segment are not transmitted, the data element separator (+) must be sent as many times as the number of data elements contained in the omitted set. On the other hand, if the set of simple data elements corresponds to a composite, only one separator per composite must be transmitted, whatever the number of components is.

It is clear that composites are of a simpler level than segments. Furthermore, a group of elements that can be used on its own in a message should be defined in segments. This is still too subjective. Stricter rules and rigour are required to define what should go into a composite and what should go into a segment.

4. Evaluation of the Message Structure

This section evaluates the message structure. Below, the following issues are addressed:

1. the use of the FTX segment;
2. specific segments;
3. the use of segment groups;
4. the order of segments and segment groups;
5. the structuring of messages into sections;
6. the problem of multifunctional and extended messages; and
7. the variety of ways of establishing the relationship between data.

4.1. Use of FTX

The FTX segment, which allows free text information, goes against the concept of EDI since FTX does not always provide machine processable information. Therefore, it should be recommended to avoid its use whenever it is possible.

4.2. Specific Segments

More than half of the segments each appear in less than twelve different messages, implying that many segments have particular functions to match particular message functions. This is in contradiction with the guideline below.

> *"If a new segment has to be designed, it should be generic, allowing for use across the widest possible number of applications." [4]*

However, it is also specified that:

> *"the creation of a specific segment needed for certain defined business areas is permissible whenever the result is approved under the maintenance procedures." [4]*

Nevertheless, before drawing any conclusion, the time that the segments concerned were introduced in the directory, should be taken into account. It is clear that a segment developed in 1992 cannot appear in many messages from the 92.1 directory. A further analysis of the use of segments in messages from the D.94B directory would be interesting.

4.3. Use of Segment Groups

Some existing segment groups are used very frequently. Consistency could be improved by creating a small directory of common segment groups or by providing some guidance in grouping segments, so that they will be used in a consistent fashion.

A typical example is the use of NAD, CTM and COM. In a majority of cases, the COM relates to contacts (CTA) of a party identified in NAD.

However, this implies extra maintenance problems.

4.4. Segment and Segment Group Order

Some messages, which are similar to each other in their function, use the same segments and groups of segments in a different order. This inconsistency can sometimes be explained by the fact that the order has to be modified to avoid segment collision.

However, the revision of the syntax proposes a new segment collision avoidance technique which consists of the allocation of a unique segment tag extension. [5] If it is accepted, the messages that are very similar in the segments they are composed of, could be structured in the same order. However, it may be that no new technique is recommended and that the UNS can be applied in a more flexible way.

4.5. Message Structuring Into Sections

Current messages are divided into one, two or three sections. Although an application need not take account of this, mandatory structuring of messages into

sections would make messages clearer and easier to understand by human beings and each section could have a defined function.

As it is already done for some messages, each message could be divided in three sections, as follows:

1. a heading section, that would contain information related to the whole message and default information;
2. a detail section, that would constitute the body of a message; and
3. a summary section, that would be used for controls and totals for line item details.

4.6. Multifunctional and Extended Messages

The construction of the transport message framework (IFTMFR) promotes consistency between messages because each structure is based on a common framework. However, there is a drawback with this approach since there is an interdependency between the messages of the framework and amendments to a message may have an impact on related messages of the framework.

Construction message development group did not create a message framework. However, the same message structure is used to fulfil three diverse business functions. As with the transport messages above, the use of a generic structure promotes consistency but the disadvantage of it is the impact of changes on a message set rather than a message.

The extended financial messages, like CREEXT and PAYEXT, have been developed so that they can provide full details of the transactions to which the message relates, namely REMADV information. There are now user requirements to amend the Remittance Advice, which, if amended, will impact many related messages. [6]

This shows that the number of changes liable to affect these kinds of messages will be high since they are large and include the same information given by another message type. It will result in changes in implementation guidelines and in EDI applications.

4.7. Grouping Related Data

Currently, there are three different ways of establishing the relationship between data:

- through grouping data elements in a composite;
- through grouping elements in a segment; and
- through grouping segments in a message.

The relationship is always positional and any data item is defined by its position in relation to data around it.

It is obvious that these element groupings are of different levels, since they use different structures but they are not clearly defined in terms of data levels. This is closely related to the use of EDIFACT structures.

4.8. Conclusion

There is a great diversity of messages. However, from the problems highlighted earlier, we can deduce some criteria that should be taken into account when investigating a new method to design messages.

The criteria are:

- **simplicity** and **flexibility**: messages should be as simple and as flexible as possible both in structure and in concepts;
- **stability**: messages should be stable since changes in messages infer updating of directories, implementation guidelines and EDI applications and thus increasing costs for maintenance. However, changes in the real world (new functionalities) may have to be reflected in new message structures.
- **consistency**: messages should be consistent, which is more important in cross-industry;
- **efficiency**: messages should be built taking into account the efficiency in processing data. This criteria is more important in interactive EDI (I-EDI) where time is crucial.

5. Conclusion

In conclusion, the lack of strict definitions for the different structures as well as the lack of guidance in the use of these, can lead to inconsistency and ambiguity.

The five criteria presented in the evaluation of the message structure can be applied to all EDIFACT directories. These are: simplicity, flexibility, stability, consistency and efficiency. They constitute design principles that will absolutely have to be considered when investigating the object-oriented approach to design messages.

An idea for a new way to design messages that has already been suggested is as follows: current messages could be divided in smaller and simpler common blocks of information. There would be no message structure any more but a new "message" would then be a non-predefined set of blocks. These blocks would be designed in order to be reusable. This would increase:

- simplicity and stability, as these blocks would be smaller;
- consistency, since blocks would be common and reusable;

- efficiency, as an application must forward some information, it would no longer be necessary to manipulate the received data to put it in another message. The received block could be forwarded just as it was. [6]

This idea, suggested by a Swedish group, is still very hazy in its description. In the next chapters, after introducing the object-oriented approach, we first study how two message development groups try to overcome EDIFACT deficiencies. We also present two proposed methods to design new EDIFACT structures.

6. Glossary of Terms

CDE.....Composite Data Element
DE .....Data Element
EDIFACT.....Electronic Data Interchange For Administration, Commerce and Transport
I-EDI .....Interactive Electronic Data Interchange
SDE .....Simple Data Element
UN/EDIFACT.....United Nations Rules for Electronic Data Interchange For Administration, Commerce and Transport
UNSM .....United Nations Standard Message

7. References

[1] UN/ECE
The 92.1 UN/EDIFACT Composite Data Element Directory
UN/ECE, Geneva (Switzerland), 1992

[2] UN/ECE
The 92.1 UN/EDIFACT Segment Directory
UN/ECE, Geneva (Switzerland), 1992

[3] UN/ECE
The 92.1 UN/EDIFACT Message Directory
UN/ECE, Geneva (Switzerland), 1992

[4] UN/ECE
Design of UN/EDIFACT Messages: Guidelines and Rules
TRADE/WP.4/R.840/Rev. 2
UN/ECE, Geneva (Switzerland), 11 January 1994

[5] UN/ECE, Syntax Development Group
EDIFACT Syntax : Corrigendum
TRADE/WP.4/R.1082/Corr.1
UN/ECE, France, 1 September 1994

[6] Ebba Lonaeus and Henning Klawitter
An Alternative Approach to Message Design
SWEDIFACT Finans, Sweden, 26 October 1993

4

Introduction to the Object-Oriented Approach

1. Introduction

This chapter is devoted to the object-oriented approach. It details mainly the object-oriented analysis by introducing:

1. the natural aspects of the object-oriented analysis,
2. object-oriented concepts, and
3. the advantages of the object-oriented analysis.

We close this chapter by explaining how the idea to apply the object-oriented approach to EDIFACT message design occurred and what it would mean.

2. Introduction to the Object-Oriented Analysis

2.1. The Natural Aspects of the Object-Oriented Analysis

The object-oriented analysis is a relatively new method and has success because it has natural ways:

1. to reflect nature's own techniques for managing complexity; [1] and
2. to organise information in ways that are familiar to man. [2]

2.1.1. Reflection of Nature's Own Techniques for Managing Complexity

All living things are composed of cells. Cells are organised packages that combine related information and behaviour - just as objects. Information is contained in the DNA and molecules within the nucleus of the cell. Structures outside the nucleus determine the behaviour of the cell.

The cell is surrounded by a membrane that protects and hides the internal working of the cell from outside intrusion. Cells can only read and control their own information. To communicate with each other, cells send chemical requests to one another.

> *"This message-based communication greatly simplifies the way cells function.... The membrane hides the complexity of the cell and present a relatively simple interface to the rest of the organism.... As you can see from the structure of the cell, encapsulation is an idea that's been around for a very long time." [1]*

2.1.2. Organisation of the Information in Ways that are Familiar to Man

The object-oriented approach is based on concepts that we learned when we were very young: objects and attributes, wholes and parts, classes and members. According to Encyclopaedia Britannica, man constantly uses three different methods to organise the world:

1. the experience differentiation between particular objects and their attributes, i.e. when man differentiates a tree from its height;
2. the distinction between objects as wholes and their parts, i.e. when man makes the difference between a tree and its branches;
3. the formation of classes and objects and the distinction between different classes and objects, i.e. when man forms a class with all trees and a class with all stones and when he differentiates.

The approach of the object-oriented analysis is based on these three organisational methods.

2.2. Object-Oriented Concepts

The fundamental ideas that underlie the object-oriented analysis include:

1. objects,
2. classes,
3. attributes,
4. methods,
5. encapsulation,
6. messages,
7. generalisation/specialisation structure and inheritance,
8. polymorphism,
9. whole/part structure, and
10. instance connections.

2.2.1. Objects

An *object* is an abstraction of something in a problem domain. It is something to which we apply a concept, a particular idea or understanding we have of our world. An object may be, for example, an organisation, an invoice, a book, a vehicle, a person, etc.

An object has got characteristics and behaviours. The characteristics of an object are called variables [1] or attributes [2]; they are everything an object "knows". Behaviours are everything an object can do in relation with its attributes and they are called methods [1] or services [2].

An object may be composed of other objects. These objects, in turn, may be composed of objects. This intricate structure allows very complex objects to be defined.

Figure 6 shows the graphical representation of an object.[1]

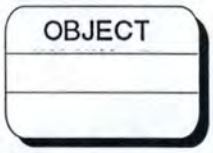


Figure 6: Object

2.2.2. Classes

A *class* describes objects with a uniform set of attributes and methods. A class is a category of objects. An object in a class is called an instance of this class and differs from the others by the values of its attributes.

We draw the attention to the fact that an object is concerned with both attributes and methods with which attributes are manipulated. In the object-oriented world, the attributes and the methods for each class are packaged together. Attributes cannot be accessed or manipulated except with the methods that are part of the class.

A class is graphically represented in Figure 7.

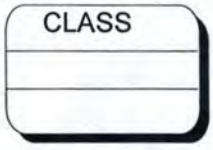


Figure 7: Class

2.2.3. Attributes

As it is said above, the *attributes* of an object are its characteristics. All together, they represent the state of the object at a certain time. They can be accessed only through the methods of the object.

Figure 8 is the graphical representation for a class with three attributes.

[1]The graphical representations used in this work are based on [2].

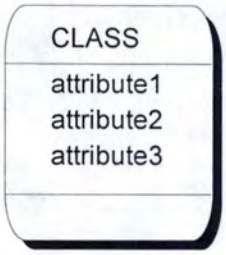


Figure 8: Class with Attributes

Figure 9 illustrates a STUDENT class with the attributes name, address, and section.

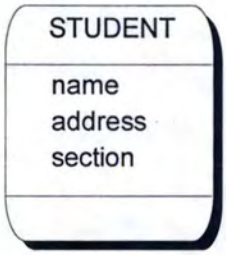


Figure 9: Example of a Class with Attributes

An example of an instance of the STUDENT class is Student1 whose name is J. Smith, who lives at number 12, Market Street, Beckhenham BR3 1LS and who is in the math section.

2.2.4. Methods

As mentioned earlier, the behavioural aspect of an object is represented by its *methods*. Methods specify the ways in which the attributes of an object are manipulated. The values of the attributes of an object evolve with the methods carried out. The methods in a class reference only the attributes of that class. They should not directly access the attributes of another object. To use the attributes of another object, they must send a message to that object.

Figure 10 is the graphical representation for a class with two methods.

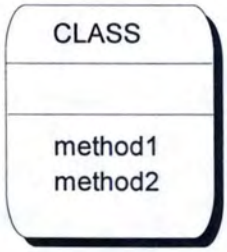


Figure 10: Class with Methods

In Figure 11, the STUDENT class is represented with two methods: update_address and update_section.

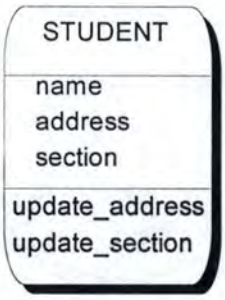


Figure 11: Example of a Class with Methods

2.2.5. Encapsulation

The mechanism of packaging characteristics and the behaviour of an object together is called *encapsulation*.

The object hides its attributes from the other objects and allows attributes to be accessed via its own methods. This is called information hiding. Encapsulation hides the details of its internal implementation from the users of an object. Users understand what operations may be requested of the object but do not know the details of how the operation is performed.

That mechanism of encapsulation permits protection in two ways:

1. it protects an object's attributes from being corrupted by other objects; and
2. an object protects other objects from the complications of depending on its internal structure.

It must be noted that encapsulation allows flexibility: object implementations can be modified without requiring the applications that use them to be modified also, this because of the fact that encapsulation separates how an object behaves from how it is implemented.

2.2.6. Messages

Objects interact with each other through *messages*, where messages are used by objects to carry out their methods. When an object "sends" a message to another object, it specifies the name of the object concerned and the method to be carried out, possibly with some parameters.

An example of a message, using the class defined above, STUDENT, can be [Student1, update_section (science)]. This message means that the object Student1 is asked to carry its method update_section, updating its attribute section with the value "science".

2.2.7. Generalisation/Specialisation Structure and Inheritance

The *generalisation/specialisation structure* permits a class of objects, a subclass, to be defined as a special case of a more general class, a superclass.

Thanks to the mechanism of *inheritance*, instances inherit all and only the features of the classes which they belong to.

Figure 12 is the graphical representation for the generalisation/specialisation structure. An oval is used to symbolise the generalisation/specialisation structure.

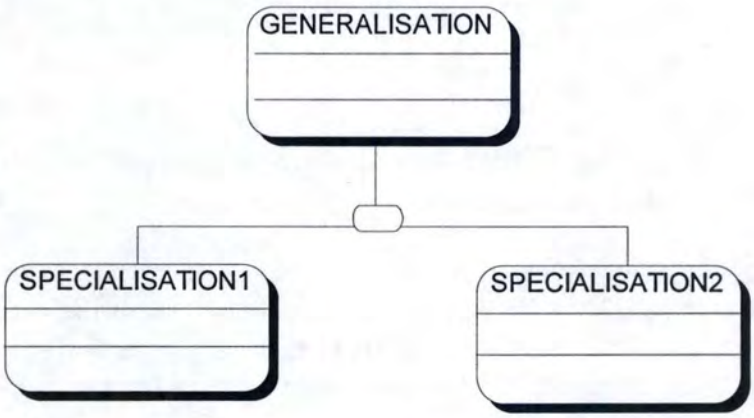


Figure 12: Generalisation/Specialisation Structure

Figure 13 illustrates a PERSON superclass and two subclasses MAN and WOMAN. The two subclasses inherit the attributes name, address and date_of_birth and the method modify_address of PERSON. MAN has as specific attribute military_service (that indicates if the man concerned has done his military service or not) and as specific method update_military_service (that consists in updating the attribute military_service). The class WOMAN has only a specific attribute which is maiden_name.

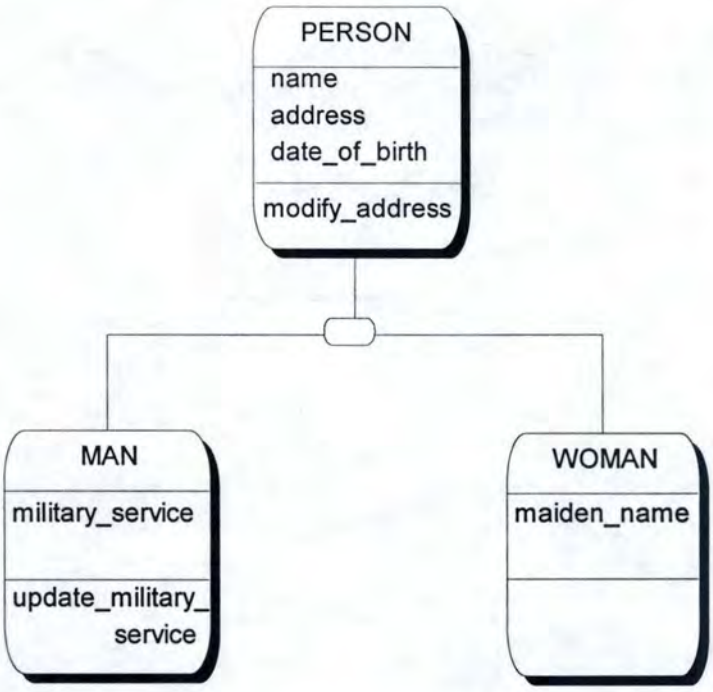


Figure 13: Example of a Generalisation/Specialisation Structure

It must be noted that a class may inherit attributes and methods from more than one superclass. This is called *multiple inheritance*.

2.2.8. Polymorphism

Some inherited methods may require customisation to meet a particular need. Thus, subclasses may redefine the implementation of the methods which they inherited from their superclasses, and in this way, override the latest. The ability to use the same expression to denote different method implementations is referred to as *polymorphism*.

One strength of polymorphism is that a request for a method can be made without knowing which method should be invoked. These implementation details are hidden from the user.

2.2.9. Whole/Part Structure

The *whole/part structure* allows to specify a relation of "composition" between a whole object and a part object. That structure indicates that an object (a whole object) is composed of other objects (part objects), which can possibly be made of other objects,... Cardinality constraints which indicate a constraint on the number of objects that must participate in a relation are taken into account.

Figure 14 is the graphical representation for the whole/part structure. A triangle is used to symbolise the whole/part structure.

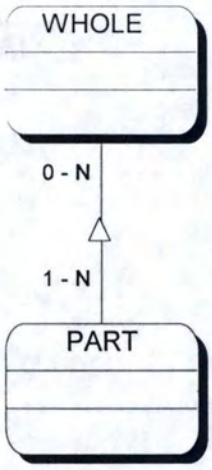


Figure 14: Whole /Part Structure

If an organisation is considered as a group of people, the fact that an organisation has at least one employee and that an employee works in one and only one organisation is represented by Figure 15.

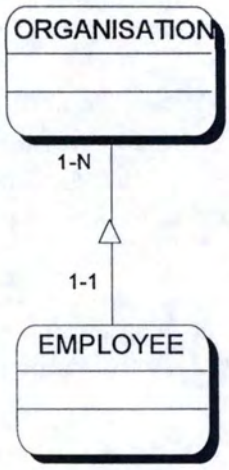


Figure 15: Example of a Whole/Part Structure

2.2.10. Instance Connections

The *instance connection* is used to represent the relationships that an object has with others in order to fulfil its responsibilities.

Figure 16 is the graphical representation for instance connection.

Figure 16: Instance Connection

Figure 17 illustrates that a patient is associated with a medical file.

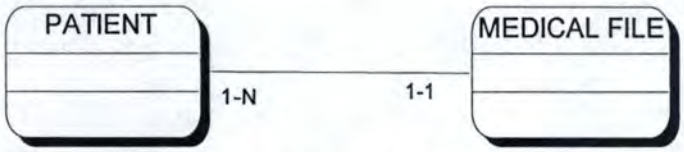


Figure 17: Example of an Instance Connection

2.3. Main Advantages of the Object-Oriented Analysis

The first advantages, already sufficiently underlined, are the natural way to reflect nature's own techniques for managing complexity and the way to organise information in ways that are familiar to man.

Other advantages of the object-oriented analysis can be deduced from the previous section detailing object-oriented concepts. We see two main advantages: reusability and flexibility.

2.3.1. Reusability

Reusability means that the results of previous analyses can be integrated and - if needed - customised in a current analysis. This is possible by looking for commonality of classes and defining general classes in order to reuse them. General classes can afterwards be specialised to meet the needs of the system to model. The modelling of systems from components that already exist provides faster and cheaper development.

2.3.2. Flexibility

The advantage of flexibility is directly deducted from the concepts of encapsulation, inheritance and polymorphism. In the previous section, it is explained that, thanks to encapsulation, objects are protected against each other. Moreover, the implementation of a method of a class can be amended without requiring any other modifications of the applications that use the method in question. Polymorphism allows the use of the same expression to denote different method implementation; however, a request for a method can be made without knowing which method should be invoked.

3. The Object-Oriented Approach to Overcome EDIFACT Deficiencies?

EDIFACT representatives are aware of the great diversity among the EDIFACT message structures and of the ever increasing number of messages. After several years devoted to the development of EDIFACT, views have changed. It is now believed that current methods for designing messages may be too pragmatic.

This is actually what we developed in the previous chapters: UN/EDIFACT is not perfect and suffers from imprecisions and ambiguities due to the lack of rigour in defining composite, segment and message structures. A better structured approach for designing EDIFACT messages is required.

In their concern to homogenise all messages, EDIFACT representatives are considering the adoption of the object-oriented approach applied to EDIFACT message design, not only because of its popularity, but particularly since object-oriented concepts seem to fit with EDIFACT data element principles.

To date, not a lot has been done in this direction, since the EDIFACT community's first concern is to develop new messages to meet user requirements. This alone explains the lack of references.

Though, among some message development groups, the current trend is to use the data modelling principles during the business analysis phase. This is already a good step towards better structured messages but it does not solve all the problems.

Applying the object-oriented approach to EDIFACT structures could also mean defining classes of objects, attributes, relationships between classes, and possibly methods. This leads to redefining EDIFACT concepts in terms of attribute, object, class, relationship...

In the next two chapters, we detail two current methods to design EDIFACT messages using object-oriented data modelling and two attempts to redefine EDIFACT structures in terms of object-oriented structures.

4. Glossary of Terms

EDIFACT ..... Electronic Data Interchange For Administration, Commerce and Transport

5. References

[1] David A. Taylor, Ph. D.
Object-Oriented Technology - A Manager's Guide
Addison-Wesley, USA, 1994

[2] Peter Coad and Edward Yourdon
Object-Oriented Analysis, 2nd edition,
Prentice Hall, USA, 1991

5

Data Modelling to Design EDIFACT Messages

1. Introduction

Object-oriented data modelling is used by two message development groups (MDGs) to model data which they want to map into a new message. These groups try to reflect their data models in messages using current EDIFACT structures. The methods adopted by MD9 (Health Care Message Development Group) and MD8 (Tourism, Travel and Leisure Message Development Group) are detailed in this chapter.

2. Design of Health Care Messages

As automated interchange information in health care increases, it is essential to provide appropriate information interchange standards. Standards are required to facilitate electronic transfer of requests for and results of investigations between the many systems currently used.

Project Team 008 of CEN/TC 251 prepared the European Prestandard. Its implementation will:

- facilitate the electronic transfer of orders for laboratory investigations from requesting health care parties, to clinical laboratories;
- facilitate the electronic transfer of reports from clinical laboratories to requesters and other health care parties;
- minimise the time and effort required for the introduction of interchange format agreements; ...

The method by which this European Prestandard has been developed is based on the recommendations of the CEN technical Report "Investigation of syntaxes for existing interchange formats to be used in healthcare" [1].

This section covers:

1. an overview of the European Prestandard [2], and
2. a detailed description of the recommendations of the CEN Technical Report [1], which is the most relevant for us.

2.1. European Prestandard

> *"The European Prestandard specifies general messages for electronic information exchange between clinical laboratories' computer systems and computer systems used by health care parties requesting the services of, or receiving results from, clinical laboratories." [2]*

The prestandard defines the communication roles which shall comply with the specifications of the prestandard when exchanging clinical laboratory information. The services that shall be supported by a communication party are also given.

The main part of the prestandard consists of the domain information model. The approach detailed in the previous section and the technique described by P. Coad and E. Yourdon [3] are at the root of the domain information model. Domain information model diagrams can be found as well as the textual descriptions of the objects appearing in the diagrams, their attributes and their relationships.

Finally general message descriptions are given. A message exchanged between communication parties incorporates only a limited amount of the information covered by the domain information model. By message type, a number of objects and attributes that are part of the domain information model are not required to satisfy the purpose of the message. This explains the differences of cardinality constraints between the domain information model and the message description.

Examples of general message descriptions are:

- new laboratory service order,
- laboratory service order modification,
- new laboratory service report, ...

2.2. Method

2.2.1. Description of the Method Used

To develop a new EDIFACT health care message, the object-oriented analysis is used to analyse the problem domain and the responsibilities of the system. An object-oriented model with classes, attributes and relations between these classes defining the scope of the required message is created. In other words, all the information which needs to be transmitted in a new message is structured in an object-oriented model.

An object-oriented health care model can contain:

- objects and classes,
- attributes,
- generalisation/specialisation structures,
- whole/part structures, and
- instance connections.

Faced with these object-oriented concepts, the different elements of an EDIFACT message are:

- simple data elements,
- composite data elements,
- segments,
- qualifiers,
- sequence, in other words, ordered juxtaposition of segments or segment groups,
- grouping of segments or segment groups, which MD9 calls nesting.

The problem is then to map object-oriented structures into EDIFACT structures by converting the different constructs of an object-oriented model into their suitable counterparts in the EDIFACT syntax.

Classes describe categories of "things" in the real world. An object can be mapped into a segment group or in one or more segments, which are the essential EDIFACT grouping level. Segment groups are composed of related segments, or even other segment groups. Therefore, these should be suitable for describing a complex object, since, as already mentioned, complex objects may be composed of other objects.

Attributes represent the state of an object. They carry the information about an object. Attributes can be mapped into elements of a stand-alone segment. They can also be mapped into segments belonging to a group or even in their data elements. Attributes are simple data elements, composite data elements, or segments.

The *generalisation structure* can be mapped into a generic simple data element, composite data element, segment or group of segments, while the *specialisation structure* can be mapped into the qualifier thus qualifying or giving precision to the function of the generic element in question.

The *whole/part structure* can be mapped into nesting or sequence.

Instance connections are not as easy to map because they may imply a many-to-many relationship. This can be solved by using references. Implicitly, references internal to a message can only be mapped by nesting or sequence. The same can also be done explicitly on a semantic level.

Figure 18 summarises the mapping rules from the data model to EDIFACT proposed by MD9.

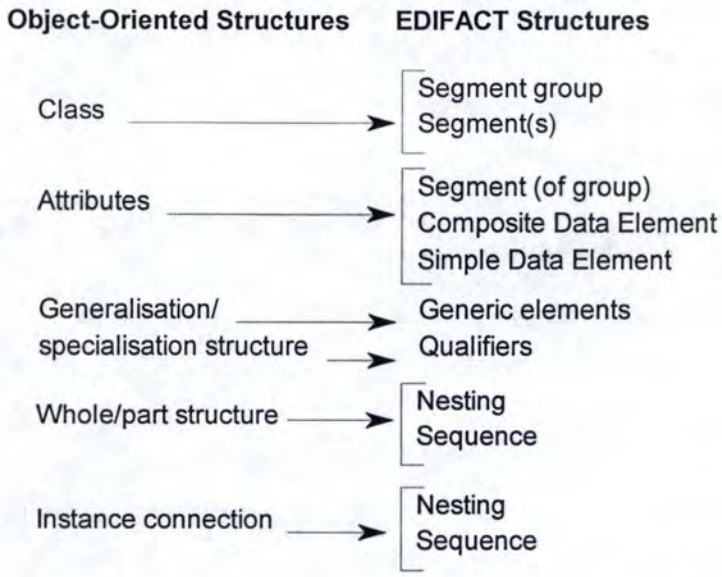


Figure 18: Proposed Mapping Rules from the Data Model to EDIFACT

A one-to-one correspondence of object-oriented and EDIFACT structures is not possible. This is partly because the contents of data elements and segments are already defined and must be used as they are, as well as partly because the structures are too different to match: i.e. the approach used to build EDIFACT structures is different from the object-oriented approach.

2.2.2. Example

To illustrate the method described above, here is a summarised example about laboratory communication presented in [1]. The example covers information exchanged from a requester to a laboratory.

In this example, there are two communication roles: a **requester** and a pathology **laboratory**.

> *"A Requester sends **Requests** to a laboratory. A Request consists of a set of service specifications ordered simultaneously. The lowest level of service specification available in a Request is called a **Request element**. A Request for laboratory services also includes information about a **Specimen**. The Specimen could either have been collected by the Requester and a **Collected Specimen** object included in the Request or the Requester could ask the Laboratory to collect the Specimen and add a **Requested Specimen** object to the Request. A **Requested Test** object specifies which tests are going to be performed on the Specimen. The Requested Test is a subtype of Requested Element.*
>
> *A Request is related to one **Patient**. In addition to demographic information about the Patient, an extract of the Patients **Clinical Information** may be included in the Request". [1]*

The first step is to elaborate the domain information model for the problem that we are dealing with. This is represented in Figure 19. Some attributes have been added.

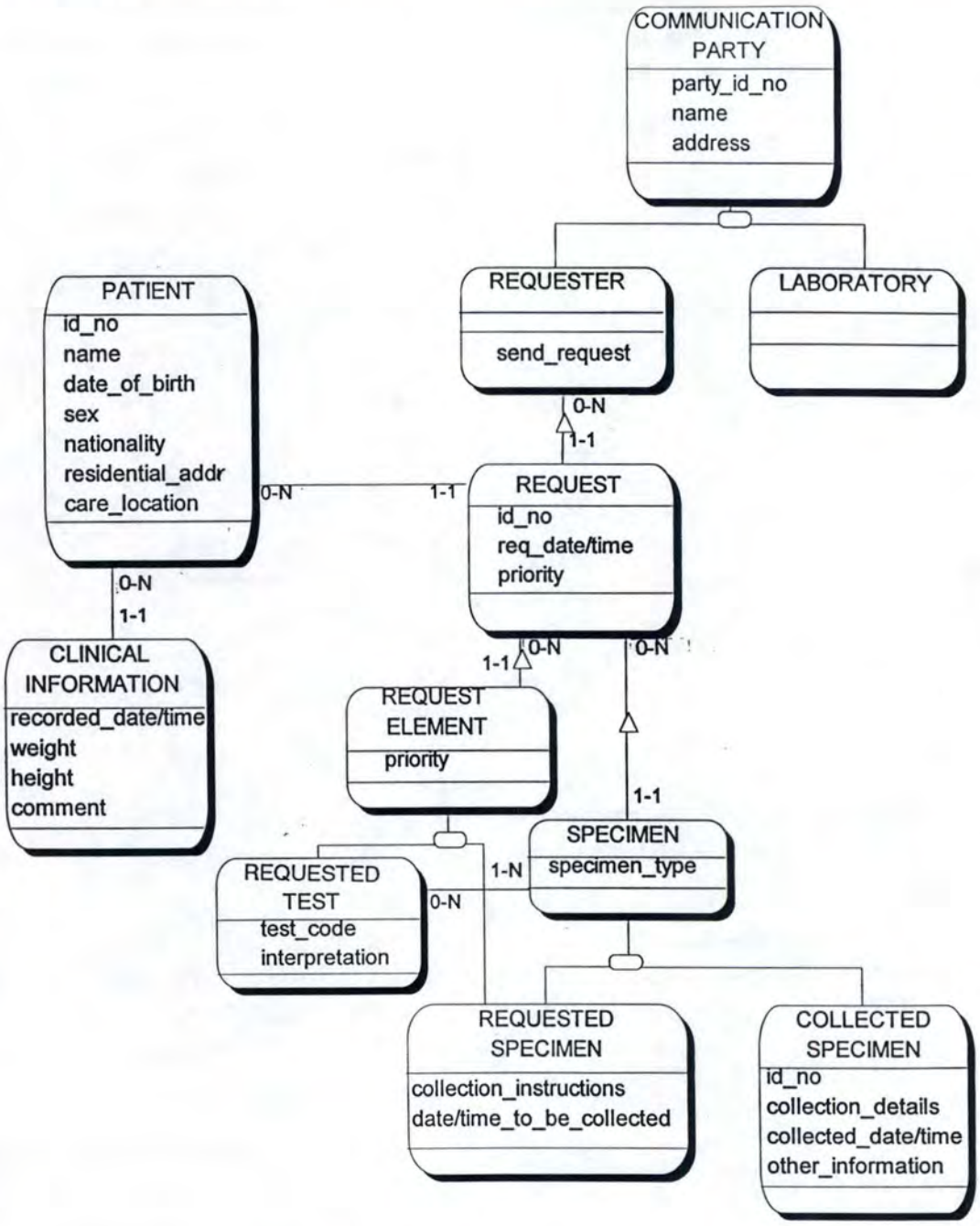


Figure 19: Domain Information Model[1]

[1]We would like to draw attention to the fact that the domain information model of Figure 19 comprises an example of multiple inheritance: the class REQUESTED SPECIMEN inherits both the attributes specimen_type and priority from respectively SPECIMEN and REQUEST ELEMENT.

It can be noted that REQUESTER and LABORATORY are considered as specialised classes of a general class COMMUNICATION PARTY.

Moreover, another class, ADDRESS, can be created to reuse the definition of an address for both PATIENT and COMMUNICATION PARTY. It is illustrated in Figure 20.

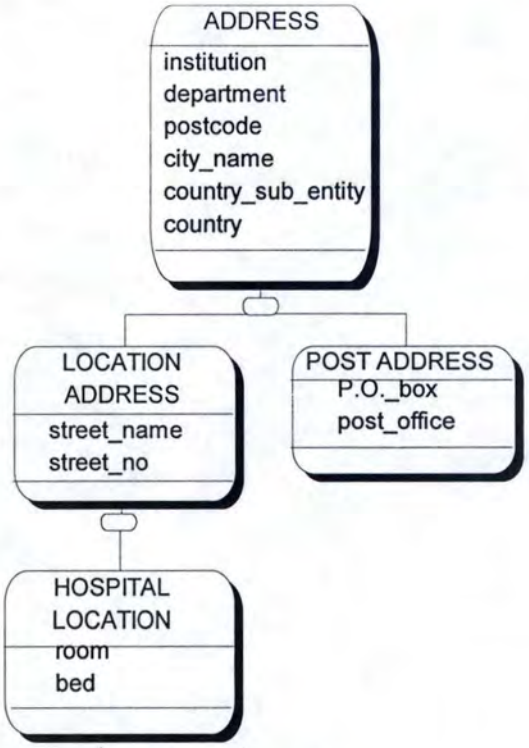


Figure 20: Address

The second step is to map into interchange format dependent message description. According to the domain information model, most classes are mapped into groups of segments. FTX is chosen to map comments about clinical information and other information about collected specimens since their contents are not detailed.

The segments marked with a plus sign are segments used in the MEDRPT message of status 0. Their description can be found in Appendix B. The two segments marked with an asterisk are new and result from the splitting of the NAD segment: they are NAM NAME and ADR ADDRESS.

Below is the mapping of objects and attributes into EDIFACT segment groups. We remind the reader that the purpose of the message is to transmit a request of a requester to a laboratory.

REQUEST

	BGM	id_no
	DTM	req_date/time
+	PTY	priority

COMMUNICATION PARTY

*	NAM	party_id_no, name
*	ADR	address

ADDRESS

*	ADR	post address and location address (postcode, street_no, etc.)
*	NAM	institution_name
	LOC	hospital location/department

PATIENT

+	PID	id_no, name, sex
	DTM	date_of_birth
+	ATT	nationality
	ADDRESS	
	* ADR	address, residential_addr/care_location
	* NAM	institution_name in address
	+ LOC	care_location

CLINICAL INFORMATION

	DTM	recorded_date/time
	MEA	height, weight
	FTX	comment

COLLECTED SPECIMEN

+	SPE	specimen_type
	RFF	id_no
+	PRO	collection_details
	DTM	collection_date/time
	FTX	other_information

REQUESTED SPECIMEN

+	SPE	specimen_type
+	PRO	collection_instructions
	DTM	date/time_to_be_collected
+	PTY	priority

REQUESTED TEST

	TEM	test_code
+	PTY	priority
	RFF	reference_to_specimen
+	PRO	interpretation

The last step is to design the message.

UNH	M	1			
—Segment group 1			M	1	REQUEST
BGM	M	1			id_no
DTM	M	1			req_date/time
PTY	M	1			priority
—Segment group 2			M	n	COMMUNICATION PARTY
NAM	M	1			party_id_no, name
—Segment group 3			C	n	ADDRESS
ADR	M	1			post/location address
NAM	C	1			institution_name in address
LOC	C	1			department
—Segment group 4			M	1	PATIENT
PID	M	1			id_no, name, sex
DTM	M	1			date_of_birth
ATT	M	1			nationality
—Segment group 5			C	n	ADDRESS
ADR	M	1			residential_addr, care_location
NAM	C	1			institution_name in address
LOC	C	2			location in hospital/department
—Segment group 6			C	n	CLINICAL INFORMATION
DTM	M	1			recorded_date/time
MEA	C	2			weight, height
FTX	C	n			comment
—Segment group 7			M	n	COLL./REQ. SPECIMEN
SPE	M	1			specimen_type
RFF	M	1			id_no
PRO	C	1			coll./req., collection_details
DTM	C	1			date/time collected/to_be_coll.
PTY	C	1			priority
FTX	C	n			other_information
—Segment group 8			C	n	REQUESTED TEST
TEM	M	1			test_code
PTY	M	1			priority
RFF	M	n			reference_to_specimen
PRO	M	1			interpretation
UNT	M	1			

2.3. Conclusion

In conclusion, in order to facilitate electronic transfer of requests for and results of investigations between health care parties, MD9 create new EDIFACT messages. In the process, they first model the data to be transmitted using the object-oriented approach. Next, they design the new message applying the results of the correspondence they established between object-oriented and EDIFACT structures. Even if it is not a one-to-one correspondence, we can see, through the example described in the section, that an object-oriented model of the information to be mapped is worthwhile: the new message is better structured.

3. Design of Tourism Messages

The process of developing tourism, travel and leisure messages is similar in broad outline to MD9's: the first step consists in an inventory of tourism data elements, which are afterwards classified and modelled. Eventually, the tourism, travel and leisure (TT&L) data model is mapped into an EDIFACT message. However, the approaches used in the two last steps are in themselves specific to MD8.

Although at first MD8 does not really use the object-oriented approach to model data (the information that has to be mapped into a new message is first modelled into a relational data model with objects and attributes), we think that the way they proceed in structuring tourism information is original and very interesting.

TINRSP, Tourism Information Response message (draft) was created according to this process. Below we describe how they created the message.

3.1. Principle of Structuring a Tourism Information Message

To design a TT&L information message, MD8 had to face two major problems:

1. there are a variety of tourism entities which can be considered as tourism information (i.e. a hotel, a restaurant, a leisure park, a museum, etc.); and
2. the same entity can be used for the full description of an object, or as a single element for complementary information. A hotel can be completely described as a tourism object or it can be described as a hotel in a leisure park where concise information is more adequate.

The solution MD8 found is the classification of data elements into three levels, depending on the function of the data elements in the message:

- level 1 - common data structures that can be used for any object description (date, tariff,...);

- level 2 - tourism object families; a tourism object family is a collection of tourism entities that can be considered as "similar", and for which a set of common characteristics can be found. A structure of this level is used to name the minimal mandatory data structure required to identify and characterize a tourism object within a family. The most important information is the status of an object taking either the value "main" or "related". It indicates which object is fully described. There can only be one "main" object per message, all the other objects must be "related";
- level 3 - specific data structures. These sets of data elements are optional and give additional information for a specific tourism entity within a family.

Figure 21 illustrates this solution.

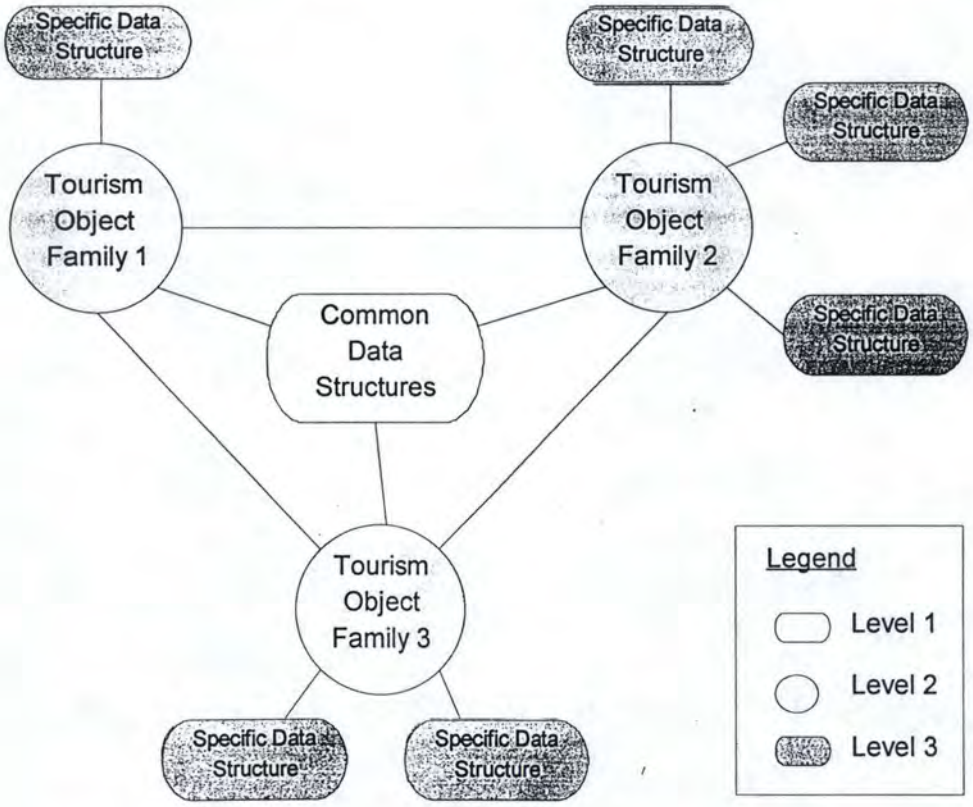


Figure 21: Structuring Levels of a TT&L Information Message [4]

This solution allows them to design one message structure for all tourism entities and to also use it for short as well as extended description of the entities.

The classification of data elements into three levels presents three advantages:

1. the identification of common data structures which can be used for any object description;
2. the modularity for the description of a tourism object within a family: an object can either be basically described for a standard use or described in detail if required; and

3. the flexibility in the evolution of the message: a new tourism object family can be added without amending the structure of the existing message.

With reference to the third chapter evaluating EDIFACT structures, consistency in a message is thus improved: the same message structure is used to transmit information about any tourism object. Stability is also better since a new object can be added without changing the existing message structure.

An example of a relational data model for the description of an *activity* as "main" tourism object (e.g. tennis) related with an *accommodation* (e.g. camp site) is given in Figure 22.

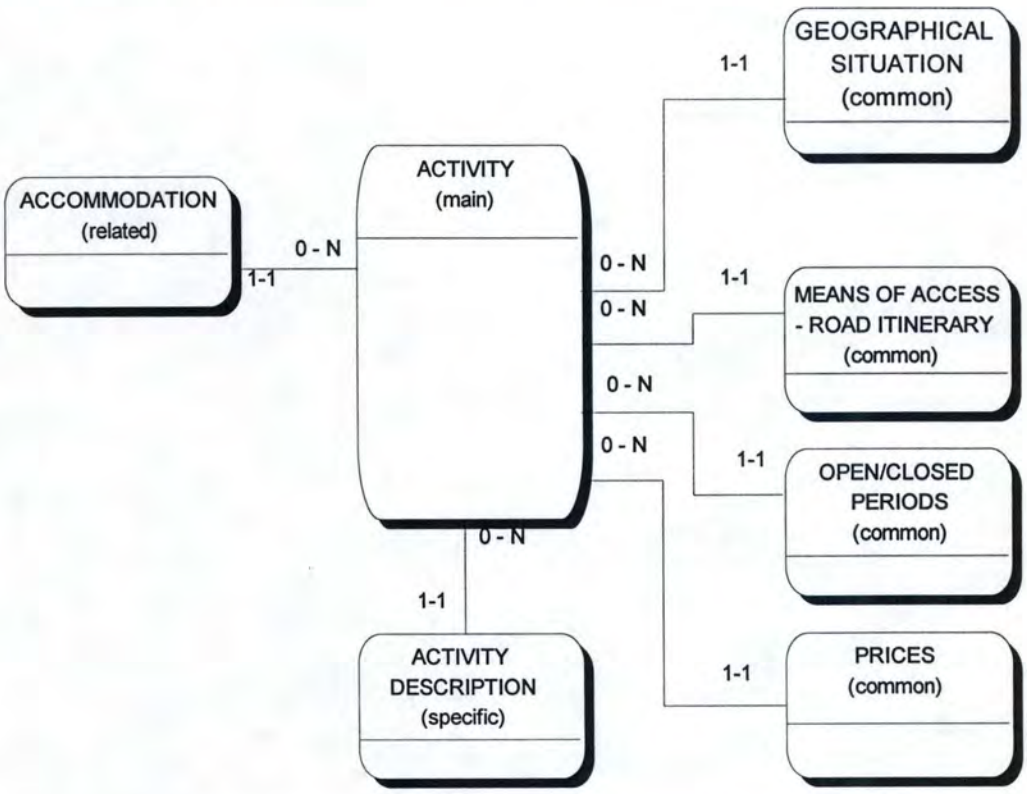


Figure 22: Example of a Tourism Data Model [4]

3.2. Principle of Mapping a Tourism Data Model into an EDIFACT Message

Below the segment table of TINRSP is given. [5] The corresponding branching diagram can be found in Appendix B.

UNH	Message header	M	1
BGM	Beginning of message	M	1
RFF	Reference	M	1
DTM	Date/time/period	C	5
LAN	Language	C	1
CUX	Currencies	C	1

— Segment group 1				M	999
TLO	Tourism and leisure object	M	1		
REL	Relationship	M	1		
RFF	Reference	C	1		
QTY	Quantity	C	99		
MEA	Measurements	C	99		
LOC	Place/Location identification	C	99		
PAI	Payment information	C	99		
ATT	Attribute	C	99		
FTX	Free text	C	99		
— Segment Group 2				C	99
TLF	Tourism and leisure feature	M	1		
QTY	Quantity	C	99		
MEA	Measurements	C	99		
— Segment Group 3				C	99
MEM	Membership details	M	1		
QTY	Quantity	C	5		
RNG	Range details	C	5		
MEA	Measurements	C	5		
— Segment Group 4				C	99
NAD	Name and address	M	1		
TLF	Tourism and leisure feature	C	5		
COM	Communication contact	C	5		
LAN	Language	C	20		
— Segment Group 5				C	99
CTA	Contact information	M	1		
COM	Communication contact	C	5		
LAN	Language	C	20		
— Segment Group 6				C	99
ITS	Itinerary section	M	1		
QTY	Quantity	C	10		
MEA	Measurements	C	10		
FTX	Free text	C	10		
— Segment Group 7				C	99
LOC	Place/location identification	M	1		
TLF	Tourism and leisure feature	C	10		
MEA	Measurements	C	10		
— Segment Group 8				C	99
DTX	Period	M	1		
DTM	Date/time/period	C	99		
FTX	Free text	C	10		
— Segment Group 9				C	99
ALC	Allowance or charge	M	1		
PCD	Percentage detail	C	1		
— Segment Group 10				C	99
MEM	Membership details	M	1		
QTY	Quantity	C	5		
RNG	Range details	C	5		
MEA	Measurements	C	5		

— Segment Group 11				C	99
DTX	Period	M	1		
DTM	Date/time/period	C	99		
— Segment Group 12				C	99
PRI	Price details	M	1		
RNG	Range details	C	1		
CUX	Currencies	C	1		
QTY	Quantity	C	99		
— Segment Group 13				C	99
TLP	Tourism and leisure product	M	1		
MOA	Monetary amount	C	1		
— Segment Group 14				C	99
MEM	Membership details	M	1		
QTY	Quantity	C	5		
RNG	Range details	C	5		
MEA	Measurements	C	5		
— Segment Group 15				C	99
DTX	Period	M	1		
DTM	Date/time/period	C	99		
— Segment Group 16				C	99
ALC	Allowance or charge	M	1		
PCD	Percentage detail	C	1		
— Segment Group 17				C	99
MEM	Membership details	M	1		
QTY	Quantity	C	5		
RNG	Range details	C	5		
MEA	Measurements	C	5		
— Segment Group 18				C	99
DTX	Period	M	1		
DTM	Date/time/period	C	99		
— Segment Group 19				C	99
DOC	Document/message details	M	1		
— Segment Group 20				C	99
MEM	Membership details	M	1		
MEA	Measurements	C	5		
UNT	Message trailer	M	1		

Tourism object references and parameters that give system oriented information on the tourism object transmitted by the message are mapped into usual segments (BGM, RFF, DTM, ...) at the beginning of the message.

All the segments where information about a tourism object will be given are arranged under a single TLO, TOURISM AND LEISURE OBJECT segment[1].

[1] The specifications of the new TLO segment can be found in Appendix B.

"The repetition of this structure allows to describe many tourism objects in a single message in a more or less detailed way. However, for a good use of this message, it is recommended to observe the following rules. ...

1. *There must be one and only one "main" tourism object in the message, all the other tourism objects must be declared as related.*
2. *Full description should be given only for the main tourism object. For a related tourism object, it is recommended to give only information summary.*
3. *Full information on a related tourism object should only be given by indication of an external reference." [5]*

The TT&L object family (accommodation, activity, food & drink or theme park) and the TT&L object type are defined in the TLO, TOURISM AND LEISURE OBJECT segment.

REL, RELATIONSHIP segment specifies if the tourism object described by the group is the main or a related tourism object.

In RFF, REFERENCE segment, it is possible to refer to a previously transmitted object so that all the properties defined for the previously transferred object are available for the current object. In this case, a data element of the REL segment takes as value "external relationship". In this way, the here above third rule is observed.

Segment group 2 (SG2), triggered by TLF, TOURISM AND LEISURE FEATURE segment[1], describes various features considered as a characteristic for the current tourism object. Within this group, specific information about a tourism object can be found as well as information like access conditions, equipment, facilities,... For example, food & drink specific features, which correspond to previously defined level 3, like the chef's name, the type of cooking and specialities are described in TLF. Every occurrence of TLF corresponds to the description of a feature of the related tourism object.

The other main segments and segment groups (SG4, SG6, SG8, SG9, SG12, SG19) specify common data like names and addresses, means of access, open/close periods, etc.

3.3. Example

To have a better understanding on how this message is used, below is a simplified example. The hypothetical tourism object to be coded is the following:

- the tourism object to be described is "Leinster Hotel", whose reference is "HT111". The hotel has 120 rooms and 4 suites;
- this hotel is related with another tourism object, fully described in the database of the receiver, a restaurant called "Leinster Restaurant", whose reference is "FD333";

[1] The specifications of the new TLF segment can be found in Appendix B.

- additional information must be given on a pub related to the hotel: "Leinster Pub", closed from the 24th of October until the 10th of November. The phone number is (44) 171 723 24 89.

The codification for this hotel would be:

UNH	
BGM	Message name (response to query)
RFF	Hotel reference (article number = HT111)
— Segment Group 1	
TLO	Object identification (accommodation, hotel) and name (Leinster Hotel)
REL	Relationship qualifier: this is the main object (main). All the other tourism objects in this message relate to this hotel
— Segment Group 2	
TLF	Identification of a hotel feature (room)
QTY	Indication of the number of rooms (120)
— Segment Group 2	
TLF	Identification of a hotel feature (suite)
QTY	Indication of the number of suites (4)
— Segment Group 1	
TLO	Object identification (food & drink, restaurant) and name (Leinster Restaurant)
REL	Relationship qualifier: this object is related to the hotel and is identified by an external reference (external relationship)
RFF	External reference of the restaurant (article number = FD333)
— Segment Group 1	
TLO	Object identification (food & drink, pub) and name (Leinster Pub)
REL	Relationship qualifier: this object is related to the hotel (related). Information follows.
— Segment Group 4	
NAD	Identification of a party for the pub (information desk)
COM	Information desk phone number = (44) 171 723 24 89
— Segment Group 8	
DTX	Identification of a closed period for the pub (closed)
DTM	Indication of the closed dates (from 24-10 to 10-11)
UNT	

3.4. Conclusion

We can see that MD8 have in fact adopted an object-oriented point of view to solve their problems. They defined a general class with common data structures (level 1) of all tourism objects that can be related to each other. That general class was specialised in tourism object families (level 2), which are accommodation, food & drink, activity and theme park, etc. The tourism object families have specific data structures (level 3) and are, in turn, specialised in object types. For example, accommodation is specialised in hotel, bed & breakfast, camp site, ... All subclasses inherit attributes of superclasses. The above approach is represented graphically in Figure 23.

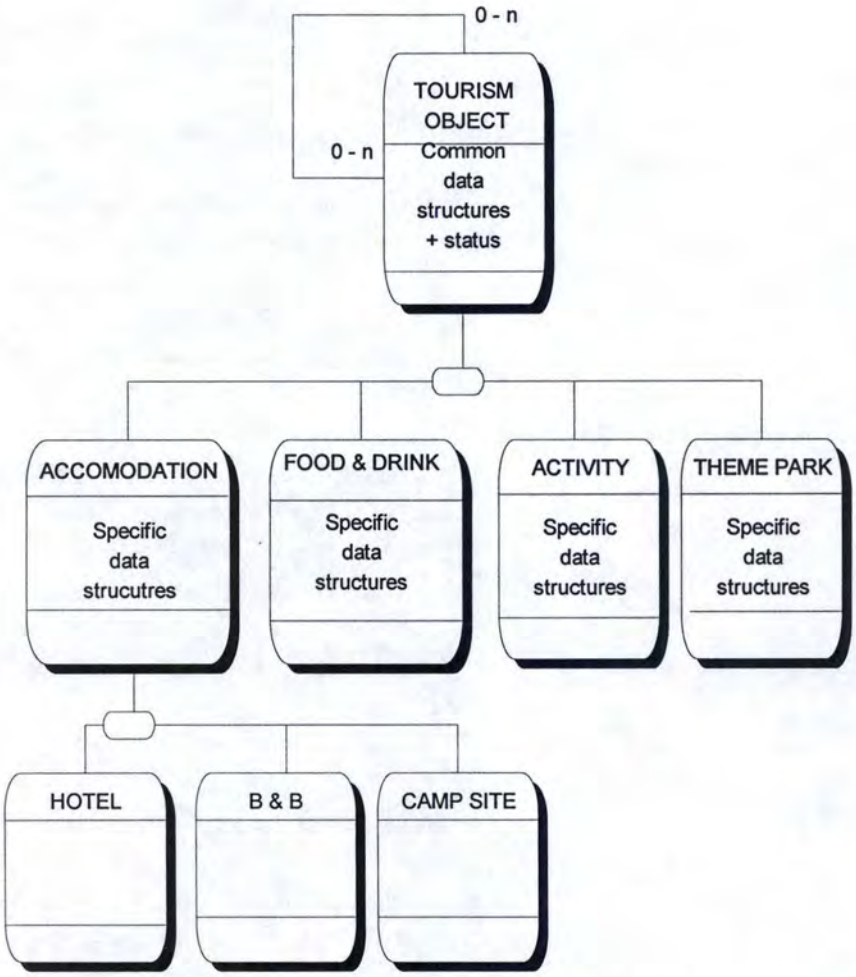


Figure 23: Object-Oriented Representation of Tourism Objects

If we examine the schema above and the resulting TINRSP message closely, we can see that MD8's mapping of object-oriented structures into EDIFACT structures is very similar to MD9's approach:

- classes are mapped into segment groups;
- attributes are data elements (of segments) or segments of groups;
- instance connections between tourism objects are made explicitly on a semantic level by introducing a data element defining the object status (main,

related or external relationship). Moreover, some common attributes are complex and could be considered as classes in themselves, related to the general class by instance connections. These instance connections are realised by nesting of segments;

- the mapping of the generalisation/specialisation structure differs slightly: the generalisation structure is mapped into a simple data element (9917 Tourism and leisure object family, coded). In order to specialise that simple data element, the object type is specified in another simple data element which follows it directly (9919 Tourism and leisure object type, coded). Although it has the function of a qualifier, it has not been called a qualifier.

4. Conclusion

We have seen in this chapter how two message development groups, MD9 and MD8, created their new messages. MD9 established a correspondence between object-oriented and EDIFACT structures resulting in more coherent and better structured messages. MD8 divided the tourism information into three levels to be able to model many tourism entities. This resulted in an original message, increasing the structuring and the flexibility as the message evolves.

In the next chapter a totally different view is exposed: to remedy EDIFACT problems evoked in the first chapters, new EDIFACT structures are investigated.

5. Glossary of Terms

CEN.....Comité Européen de Normalisation
EDIFACT.....Electronic Data Interchange for Administration, Commerce and Transport
MD8 .....Tourism, Travel and Leisure Message Development Group
MD9 .....Health Care Message Development Group
MDG .....Message Development Group
SG .....Segment Group
TC .....Technical Committee
TT&L .....Tourism, Travel and Leisure

6. References

[1] BSI Standards, CEN
Investigation of syntaxes for existing interchange formats to be used in healthcare
CR 1350:1993
BSI, Belgium, September 1993

[2] CEN/TC 251
Medical Informatics
N 94-068
CEN, Belgium, May 1994

[3] Peter Coad and Edward Yourdon
Object-Oriented Analysis, 2nd edition,
Prentice Hall, USA, 1991

[4] Joint Rapporteurs Team
EDITIN - EDIFACT Message TINRSP V. 1.0 - Temporary Version
Wien (Austria), 26-30 September 1994

[5] MD8
Minutes of MD8 Meeting
DOC 94.04 MIN.04
UN/ECE, Brussels (Belgium), 7-9 November, 1994

[6] MD8
EDITIN - EDIFACT Messages TINRSP & TINREQ, V 0 - Release 2.0 Additional Information
Brussels (Belgium), 7-9 November, 1994

6

New EDIFACT Structures

1. Introduction

In this chapter we present two different approaches to designing new EDIFACT structures. The first one, elaborated by N. Rasmussen, member of the Pan American EDIFACT Board, consists in defining new composite and segment structures in terms of entities and attributes and applies to batch EDI. [1] On the other hand, the second approach, developed within the framework of a distinct project in which R. Williams participates, concerns interactive EDI. [2], [3], [4]

2. Batch EDI

2.1. Introduction

Niels Rasmussen regrets that many alternative approaches to mapping a given set of data into a message are permitted. This results in problems that we raised in chapter 3, like inconsistencies and ambiguities in mapping data, instability in message and segments structures, ...

To improve this situation, N. Rasmussen's work takes as its starting point criteria for what constitutes quality standards and suggests design rules based on data modelling principles that meet the quality criteria.

The main problems in data mapping techniques that he sets are as follows:

a) some code values or data elements are synonymous or overlapping in meaning and usage;

b) some segments are also overlapping in function; and

c) data relationships in a message can be established through the grouping of elements in a segment and through the grouping of segments in a message.

Alternatively it is sometimes possible to use specific code values rather than linking two segments in a group and using the generic code value.

Moreover, some existing segments are designed so that they establish relationships between different entities. The NAD segment, which links PARTY and ADDRESS is an example.

According to N. Rasmussen, the quality criteria that are of importance while designing new EDIFACT messages are:

1. flexibility,
2. consistency,
3. stability,
4. transmission economy,
5. easiness of understanding,
6. easiness and economy of maintenance, and
7. easiness and economy of publishing.

2.2. Suggested Method to Design New EDIFACT Structures

In order to design new EDIFACT structures, N. Rasmussen suggests to make use of the concepts of data modelling of *entities* and *attributes*.

He defines entities and attributes as follows:

"An entity is a class of objects with common characteristics. Instances of an entity are uniquely identifiable and distinguishable from one another. Examples of entities are Parties, Places, Goods items, etc.

An attribute is a characteristic or property of an entity. Examples of attributes are Name, Weight, Street addresses, etc." [1]

He recommends to categorise data as either instances of entities or as instances of attribute types. An entity can be an attribute of another entity, in other words, instance connections may exist between entities.

"The result of this is that in messages, entities correspond to "trigger" segments, i.e. the first segment in a segment loop which must be transmitted once and only once if the segments loop is transmitted. An entity as an attribute to another entity corresponds to nested loops." [1]

In order to eliminate synonyms in codes, different code lists shall not exist for the same entity or the same attribute type and each code value in a code list shall have a meaning that is unique.

To limit the members of a code list, code values shall be as generic as possible. However, a code value that gives a more precise meaning to another code value is acceptable if the required precision cannot be achieved by linking data. But code values shall not be used as an alternative to linking data in a

segment group. The purpose of this is to achieve consistency in data mapping by only allowing linking of data through the grouping of segments.

He recommends two types of data:

1. *qualifiers*, that are codes which define or give precision to application values, and
2. *application values*, i.e. variable data that can be coded or "clear".

As for codes, specific data elements shall only be accepted when a generic data element is demonstrably inappropriate and the meaning associated with one data element shall be different from any other data element.

The objectives of these rules concerning data elements are

- to achieve consistency by avoiding synonyms, and
- to reduce the number of entries in the data element directory, to improve flexibility and facilitate maintenance.

A composite data element is recommended to be a collection of component data elements designed so as to permit the mapping of either one entity or one attribute type.

Consequently, he proposes two types of composites:

- an entity type composite whose structure would be as follows:

CXXX	ENTITY-TYPE COMPOSITE			
xxx1	Entity qualifier	M	an..3	
xxx2	Entity identification	C	an..35	
(xxx3	Code list identifier[1]	C	an..3	if required)

- an attribute type composite which could be constructed as follows:

CXXX	ATTRIBUTE-TYPE COMPOSITE			
xxx1	Attribute qualifier 1	M	an..3	
xxx2	Attribute value	C	an..350 (or 35) or C	n..18
(xxx3	Code list identifier[1]	C	an..3	if required)
(xxx4	Attribute qualifier 2	C	an..3	if required)
(xxx5	Attribute qualifier 3	C	an..3	if required)

What is important is that the function of a composite shall be only to map one entity or one attribute type.[2]

[1] This new data element is defined as a combination of the 1131 code list qualifier and 3055 code list responsible agency.

[2] An example of a composite is given in the following chapter.

Consequently, segments are also of two types:

- a "trigger" segment which contains an entity type composite. That composite data element cannot be repeated in the segment; and
- an attribute type segment which is composed of attribute type composites. These can be repeated as often as required by the application.

In this way relationships between data shall only be established through the grouping of segments in segment groups.

N. Rasmussen points out that this approach improves:

- flexibility since any new instance of an entity or of an attribute can be accommodated through additions in code lists;
- stability since segments are upward compatible by design;
- consistency in mapping because an instance of an entity or of an attribute type can only be mapped into one segment; and
- economy in transmission since attribute type composites can be repeated in a segment as often as required by the application.

Concerning messages, he recommends that each message should have a unique structure, i.e. it shall not be a subset of an existing UNSM. And, if it is the case when proposing a new message, the existing UNSM structure should be extended in order to fulfill both functions. If necessary, its name and scope should be redefined to indicate its extended functionality. In addition, the specific business function of the message transmitted should be specified by using an appropriate code in the BGM segment. In this way, he reduces the number of messages making it more economical to publish and maintain.

2.3. Conclusion

In conclusion the key elements of N. Rasmussen's proposal are as follows:

a) "The definition and design of code lists, data elements, composite data elements and segments shall be based on data modelling, in particular, the concepts of ENTITIES and ATTRIBUTES;

b) The relationships between different entities and between an entity and its attributes shall be established through the grouping of segments in segment groups only;

c) The design of messages shall be based on unique data structures rather than specific business functions;

d) The permissible data mapping techniques shall be consistent and unambiguous." [1]

3. Interactive EDI

3.1. Introduction

Richard Williams is a consultant in information systems. He participated in the development of tourism, travel and leisure messages. He is now involved in the Unicorn project. That project consists in proposing interactive messages for booking systems.

As we have already mentioned in the first chapter on the EDIFACT Standard, I-EDI is characterised by the following:

- *"a formalised association between two parties using a dialogue;*
- *the ability, dynamically, to direct the course of an EDI transaction depending upon the result of earlier exchanges within the dialogue;*
- *short response times;*
- *all the messages in one dialogue relate to the same business transaction;*
- *a transaction is a controlled set of dialogues which can take place between two or more parties" [2]*

Consequently, it is widely accepted that I-EDI messages:

- consist of small amounts of instances of data;
- involve conversation or negotiation between a requester and (one or more) responder;
- often have constraints (notably time of response) associated with them; and
- can include requirements to navigate through business options (the whole issue of linked data exchange within a scenario[1]).

According to R. Williams, the object-oriented approach can be very useful in I-EDI for several reasons:

---

[1] An important component in I-EDI is the scenario. It is the definition of the business framework that the data exchanges are required to operate in.

> *"SCENARIO. A formal description of a class of business activities which defines, among other things, the dialogue which may take place between parties to achieve a particular business objective." [3]*

A scenario includes:

- the business processing of data;
- the roles and responsibilities of different parties within the data exchange processes that are undertaken; and
- the constraints placed upon the business processing of data.

An example of a scenario can be found in Appendix C.

- an advantage of the object-oriented approach is the re-usability of objects. It is possible to define generic scenarios and objects that are agreed internationally and that can be re-used and refined in specific contexts;
- in the object-oriented approach, an object is defined by its characteristics, its attributes and also its methods describing the object's behaviour. As it has already been mentioned, in I-EDI, scenarios are established to define the dialogue that may take place between parties, between objects. These objects react differently according to the information received. If methods are associated to objects, they know exactly how to behave according to the values of their attributes and the messages they receive. Object-oriented analysis helps the behavioural aspect of objects to be taken into account;
- objects communicate between themselves by sending messages. Object-oriented messages consist in a reference of the object concerned, commands to specify the method to be processed and possibly parameters. An I-EDI message could correspond to an object-oriented message exchanged by parties - objects. It would be composed of commands which would make an application able to know exactly and promptly what to do, reducing the processing time. [4], [5]

3.2. Unicorn's Conception of I-EDI

The view being explored by the Unicorn group is that of having a singular scenario consisting of various interactive moves or building blocks. From each part of the scenario, trading partners can construct an end result from various interactions. Only relevant data and/or commands to move between predefined stages in the scenario are required. Small data exchange sets of information are proposed. These small data exchange sets are called *scenario component messages* (SCMs).

> *"Each party (object) within the data trading Community will be able to undertake certain standard processes and be able to submit and receive certain components of standardised data. The common processes available to different parties (objects) are the processing options (methods) they are interested in and can support. A method common to many objects in the UNICORN scenario is "book". The "book" method can relate to services and/or products such as hotel rooms, ferry places etc. Different parties will treat the "book" method differently, based upon products/services they have for sale, but each will respond to the buyer in a standard manner in terms of confirmation (the "confirm" method) even if different pieces of data will be required in different booking and confirmation circumstances. Common (core) data will be provided (such as booking reference number price etc.), whilst at the same time specific (to that type of booking) data will also be included. In this way we can see that some data is inherited into all methods at a generic level, but other, specific data, relating to individual booking types will also be evident in data transmissions between the buyer and the seller in any exchange." [6]*

In this context, the object-oriented concept of a message corresponds to the same concept in the interactive EDIFACT world. However, in object-oriented terms messages do not only contain data but also demand actions to be taken in many cases. Thus in the Unicorn implementation of I-EDI, the message exchange includes processing and data components. Only objects (parties) with certain

methods can implement these requests to process data. If a recipient object does not have the capacity (in terms of available methods) to process the data component of the exchange in the way in which the requester requires, it is possible for the responder to offer alternatives as well as rejections.

An example of how such "conversational" I-EDI will operate in the Unicorn scenario is given in Figure 24.

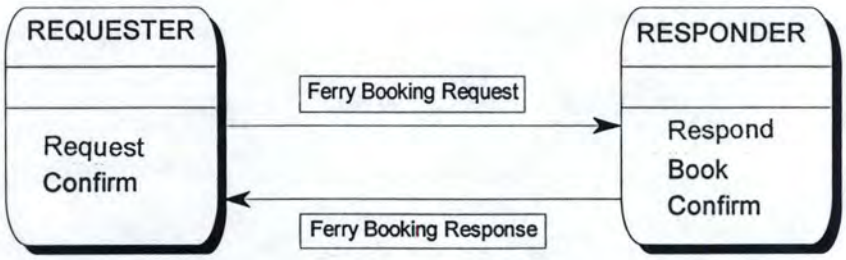


Figure 24: Example of Conversational I-EDI in Unicorn [6]

If the required methods (these are called *commands* in Unicorn terminology) exist in the recipient, then processing can take place. Thus a command is an integral component of the data exchange between a requester and a responder. One good example relates to time constraints, which would state that a response must be made within thirty minutes for the requester to be still interested in the initial request that has been made.

An example of how "navigational" I-EDI could operate in the Unicorn scenario is given in Figure 25.

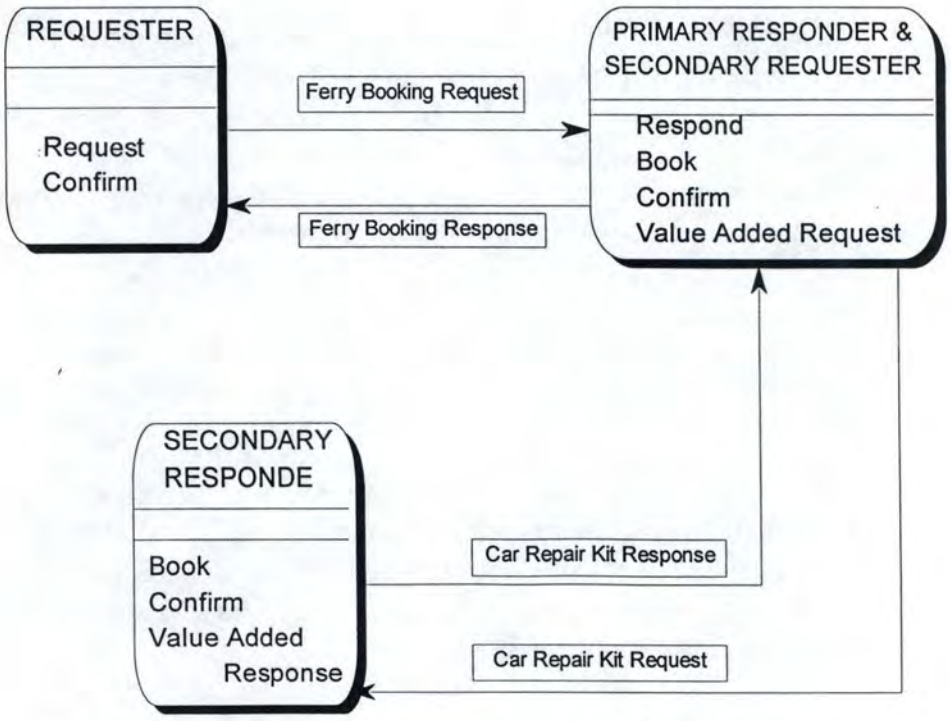


Figure 25: Example of Navigational I-EDI in Unicorn [6]

In this case the primary responder is not able to fully satisfy the requester's requirement for a ferry booking, therefore further contact is required. This further contact is for a specific component of the booking request, namely a car repair kit. The request is passed on to a secondary responder who is able to deal with this value added request. The secondary responder sends back the car repair kit response to the primary responder who then builds the full response to the initial request and sends it to the requester.

3.3. Structure of an I-EDI Message

An I-EDI message as envisaged by Unicorn is represented in Figure 26.

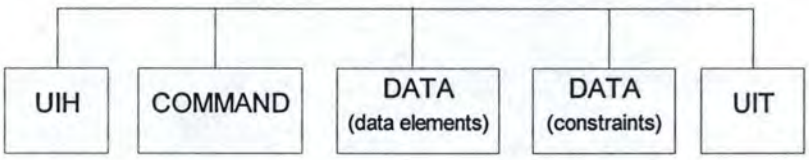


Figure 26: Structure of an I-EDI Message as Envisaged by Unicorn [6]

An I-EDI message starts with a service segment which is the equivalent of the UNH service segment in batch messages. It is followed by specific command information. The data component of the exchange follows and before closing the I-EDI message with a trailer service segment, constraint information can be added.

It is presently agreed that the data component of the exchange will observe the batch EDI syntax rule which states that each item of data is identified by its position in a message relative to a previous data item.

Clear specification in the data exchange context is needed. For example, some data elements will be necessary for all "book" methods to be able to be applied by a seller in the Unicorn scenario, namely names, booking reference numbers etc. This basic information constitutes the "core" data. For a travel booking, a departure date will be required. For a travel booking car ferry, further details will be needed. As we can see, object-oriented helps here too.

> *"What can be seen is continued inheritance of data throughout the hierarchy of data requirements for individual types of booking, sourced from a (small) common core of information requirements. It is in this way that the amount of information exchanged between parties will be restricted to a minimum in the I-EDI world. " [6]*

The I-EDI use of segments and composites is, as yet, undetermined in absolute terms. However, they agree to use data elements and code sets found in the current trade data element directory (TDED) for the data component of the exchange. In some cases, they intend to use composites and segments especially where they are deemed as mandatory for specific methods.

3.4. Conclusion

Generic scenario component messages can be defined (e.g. for "book") by including the data which will be re-usable in all business contexts where booking methods are needed. Depending on the type of booking further specification is required. This allows generic and specific scenario component messages agreed internationally to be defined. In addition, to meet specific requirements in different business communities further subdivision can be included using existing data elements.

The definition of a global and precise scenario has an impact on the speeding up of exchanges and conversations. A scenario defines the dialogue which may take place between parties. It is represented thanks to a state transition diagram of all allowed exchanges. Since the role and the responsibilities of different parties are well defined and commands are included in scenario component messages, applications are able to know exactly and promptly what to do.

It must be noted that, although the TDED will remain the source of all data elements in I-EDI, as far as time is concerned, interactive data elements will be specific, unlike data elements in batch EDI. In addition to this keeping scenario component messages relatively short, composites and segments may not be appropriate. However, as explained above, in some cases, some of them could still be of interest.

4. Glossary of Terms

EDI ..... Electronic Data Interchange
EDIFACT ..... Electronic Data Interchange For Administration, Commerce and Transport
I-EDI..... Interactive EDI
SCM ..... Scenario Component Message
TDED ..... Trade Data Element Directory
UNSM ..... United Nations Standard Message

5. References

[1] Niels Rasmussen
Data Modelling and Message Design - Substantive Comments on the UN/EDIFACT Message Design Guidelines
Canada, February 1993

[2] UN/ECE
Working Draft - EDIFACT - Application level syntax rules - Part 3: Syntax rules specific to interactive EDI, plus interactive EDI service directories
UN/EDIFACT WD 9735-3:1994
UN/ECE, Geneva (Switzerland), 20 September 1994

[3] UN/ECE, I-EDI Group
I-EDI Message Design Guidelines - Draft
UN/ECE, UK, 28 July 1994

[4] Richard Williams
Using Object-Oriented Techniques in EDIFACT : A Briefing Paper
UK, January 1995, Private communication

[5] Richard Williams
Basic Concepts and Definitions for Object-Orientation
UK, January 1995, Private communication

[6] Richard Williams
A Case for Using Object-Oriented Techniques for I-EDI Messages for the UNICORN Project
UK, May 1995, Private communication

7

Afterthoughts

1. Introduction

After having studied the UN/EDIFACT standard and some methods to overcome its deficiencies, it is time to assess the situation before closing this work.

In chapters 2 and 3, some EDIFACT deficiencies were highlighted. We think that the main problems are:

- the definition of composites and segments as having a single function. This leads to multiple uses of composite and segments structures, multifunctional composites and segments and overlapping in the use of the two structures;
- the many alternative approaches to mapping a set of data and the different ways of establishing relationships between data due to synchronic use for codes and data elements, and for composite and segments functions.

As we have seen, this can lead to confusion in mapping data and consequently to a variety of structures of messages.

Taking account of the methods to overcome these problems, detailed in chapters 5 and 6, this chapter presents first of all some afterthoughts about data modelling techniques used by MD9 and MD8, and afterwards about new structures suggested by N. Rasmussen and R. Williams.

2. The "Smooth" Method

We already feel that some message development groups are concerned and do their best to create better structured messages with the means at their disposal, in other words they model data to be transmitted and map it into a new message using current EDIFACT structures. We could call it the "smooth" method.

As we have seen in chapter 5, MD9 (Health Care Message Development Group) attempted to develop a correspondence between object-oriented and EDIFACT structures. The result is not bad. But the structures are too different to obtain a one-to-one correspondence.

Moreover, EDIFACT does not support all structuring information concepts. A comparison of EDIFACT against some selected structuring information concepts was undertaken by MD9. [1] Below is the outcome.

- **Optionality**

Optional elements are elements which can be omitted. Optionality is also applicable for relationships between elements: depending on cardinality constraints, elements must or must not participate in a relationship.

The concept of optionality is taken into account in EDIFACT: each element in composites, segments, segment groups and messages is characterised by its status C or M depending on whether it is conditional or mandatory.

- **Repetition**

Repetition is a collection of elements of the same type whose order is insignificant.

In EDIFACT, segments and segment groups only can be repeated in a message according to the definition of the message. The number of repetition is specified in the message definition.

- **Unordered Sequence**

An unordered sequence is also a collection of elements whose order is insignificant, however these elements can be of possible different types.

This concept can also be converted in EDIFACT by repeating segments or segment groups.

- **Ordered Sequence**

In an ordered sequence the order of the elements is important.

This concept is not supported by EDIFACT. However, in the implementation guidelines of a message, it is possible to specify explicitly the function of elements according to their order or their position in a sequence.

- **One-to-One Relationship**

A one-to-one relationship, where the number of elements in a relationship between types is one of each type, can be translated in EDIFACT by nesting of groups or order of segments and groups in a message.

- **One-to-Many Relationship**

In a one-to-many relationship, an element of type A can be related to one or more elements of type B, but every element of type B must only be related to one element of type B.

Such relationships can be made in EDIFACT through repetition of segments and/or groups of segments.

- **Many-to-Many Relationship**

In such a relationship, contrary to the one-to-many relationship, elements of type B can be related to more than one elements of type A.

This concept is not really supported by EDIFACT. However, it can be expressed on a semantic level by adding a reference number, making this unique from other parts of the message.

In the example detailed in chapter 5 on a request sent by a requester to a laboratory, this technique is used to take into account the fact that requested tests are performed on specimens. References to specimens are introduced in the segment group identifying requested tests.

- **Recursive Structure**

A recursive structure implies that an element of one type contains another element of the same type directly or through another type.

This is not supported by EDIFACT.

- **Single or Multiple Inheritance**

As detailed in a previous chapter, single or multiple inheritance allows a subclass to inherit all attributes and methods from one or more superclasses and to add its own characteristics and behaviours.

This mechanism is not supported by EDIFACT.

As we can see, some structuring information concepts are implicitly supported by EDIFACT, others, like ordered sequence, and many-to-many relationship require subterfuges to be mapped into EDIFACT structures, while the concepts of recursive structure and single or multiple inheritance are not supported at all by EDIFACT.

However, this is not completely true: MD8 (Tourism, travel and leisure development group) managed to map the recursive structure that exists between tourism objects in their message through adding a data element (whose value is "main", "related" or "external relationship") which defines the relationship between objects.

It should also be noted that although MD9 make use of such data modelling concepts as objects and attributes, they do not mention the concept of method. However, in the data model of the example given in the application of their method, the class REQUESTER has one method, namely send_request. And it seems that according to the example, the method is mapped into the message function. In other words, to one method corresponds one message type.

The solution to tourism, travel and leisure message design found by MD8 is interesting. They wanted a message which many tourism entities could be mapped into and with tourism entities that could be described in detail as well as in broad outline. They chose to classify data elements in three levels: level 1 (common data structures), level 2 (tourism object families) and level 3 (specific data structures), in such a way that every tourism object is mapped into the same segment group, namely segment group 1 (SG1). Each occurrence of that SG1 corresponds to one tourism object.

MD8's solution has several advantages:

- although tourism objects vary between each other, MD8 managed to establish a common structure to all of them. This makes the message well structured and easy to understand;
- the addition of a new object is realised through adding a new code defining the new object in the code list of 9919 Tourism and leisure object type, coded. This allows a very flexible evolution of the message. Moreover, this makes the message upward compatible and directories stable and easy to maintain and publish.

We think that such a method should be encouraged as often as possible because its advantages are worthwhile.

Although data modelling can help in designing better structured messages more radical changes in EDIFACT structures are required to tackle the problems at the root. But these radical changes would cost a lot to the EDIFACT community. That is why a "smooth" method is often preferred.

3. The "Radical" Method

3.1. Batch EDI

Concerning N. Rasmussen's proposal for batch EDI, we approve of his new composite structures, because of the fact that composites are very simple and contain one user data to which qualification may be required. This goes in the same direction as the opinion of A. Tornqvist, EDIFACT technical adviser, about composite and segments structures.

> *"A composite data element type could be created when the contents of a simple data element type might need further clarification, within the scope of the segment type in which it is contained.*

> *... a composite data element type would contain one item of user data. This means that one component data element would contain the actual user data, with all other components being used to enhance the meaning of the user data, so that it can be interpreted correctly by the receiving application." [2]*

However, that view differs in the fact that simple data elements in segments are not suppressed. What directly follows is already an answer to the use of composite structures.

> *"A simple data element type could then be used in a segment type as both a data element or in a composite data element type as a component data element. The proposed function of the segment type would determine when a simple data element type is to be used as a data element or needs a combination of supporting component data elements in a composite data element type to give its meaning." [2]*

As we reported in chapter 6, N. Rasmussen does not only suggest new EDIFACT structures but also rules for new data elements and for new codes in order to eliminate the different alternatives in mapping data. We have shown that there exist data elements, composites or segments which the same information can be mapped into and that this can lead to confusion.

Therefore, N. Rasmussen encourages generic data elements as often as possible as well as composite and segment structures in order to establish relationships between data only through the grouping of segments in segment groups. This increases the consistency of mapping data. Nevertheless, we must not neglect the fact that it may cost in economy in transmission because that solution may lead to more segments to transmit, which means more tags to send, thus more characters.

But because attribute type composites can be repeated in a segment, the transmission overhead is not completely true as far as the current syntax is concerned. We can try to apply the proposal to describe the height, length and depth of an element. The MEA segment could be something like this:

MEA MEASUREMENTS DETAILS		
Function: to specify measurement details		
CXXX MEASUREMENTS DETAILS n		
xxx1 Measurement application qualifier	M	an..3
xxx2 Measurement value	C	n..18
xxx3 Measurement unit qualifier	C	an..3

Hypothetical codes for xxx1 measurement application qualifier could be:

- "H" for height;
- "L" for length; and
- "D" for depth.

An hypothetical code for xxx3 measurement unit qualifier could be "cm" for centimetres.

In common EDIFACT, the segment tag has to be transmitted three times: for the height, length and depth. It corresponds to the first data stream below.

Applying N. Rasmussen's proposal, since the composite within the MEA segment can be repeated, one tag only would be required. It is illustrated in the second data stream below.

MEA+H:10:cm'MEA+L:2:cm'MEA+D:15:cm'

MEA+H:10:cm+L:2:cm+D:15:cm'

In this way, there is economy in transmission.

The idea of repeating composites is akin to the proposal of the EDIFACT syntax group.

> *"Data elements which may use the repetition technique are stand-alone data elements and composite data elements." [3]*

Adjacent occurrences of the same repeating data element in a segment are recommended to be separated by a repetition separator, namely the asterisk (*). This technique allows logical grouping together with upward compatibility. In addition, in some cases fewer separators are required. An example of it can be found in Appendix D. It must be noted that no repetition separator is required in N. Rasmussen's proposal since his new segments contain only composites.

According to the criteria which we established in chapter 3, N. Rasmussen proposal improves:

- the simplicity and flexibility of composite and segment structures,
- the stability and consistency of messages since a unique message structure is created when a message appears to be a subset of another one, and
- stability and consistency since logical grouping and upward compatibility are made possible thanks to the repetition technique. It also makes directories easy to publish and to maintain.

Since generic data elements are promoted, this method would not suit I-EDI because as far as time is concerned, interactive data elements will be specific.

A last advantage of N. Ramussen's proposal is the definition of composite and segment structures. No more ambiguity is possible. To each composite type corresponds a segment type. A composite is used to map a concept, like measurement, monetary amount, name, address, ... A segment envelops a possibly repeating composite.

In conclusion, the proposal has many advantages. However, we do not really see why an entity type composite can have only one qualifier and can not be repeated within the segment. We do not think that the distinction between an entity and an attribute at the composite structure level is appropriate. But this distinction can be made during the mapping of the object-oriented data model into the message: to each trigger segment corresponds an entity, while to other depending segments correspond attributes of that entity.

3.2. Interactive EDI

The view adopted by R. Williams in I-EDI is totally different from the preceding one for batch EDI. The reason is simple: the context is completely different. The context of I-EDI is conversational and navigational. In addition time constraints must be taken into account.

Here, the object-oriented approach is entirely exploited: most object-oriented aspects have all their acceptation.

A party is an object with attributes which define the party, and methods which are actions a party can take. General classes are defined in order to be specialised depending on the business community. Methods can then be customised and adapted to specific needs, keeping the same denomination.

Although the implementation of the methods vary, in other words the way actions are undertaken vary, other parties (objects) do not know anything about it, it is hidden. This is the strength of inheritance, polymorphism and encapsulation.

Objects communicate through messages with commands and parameters. Parties send EDIFACT messages to each other which contain a command, the action to be undertaken, as well as user data and constraints.

All this is defined in a general scenario which can also be adapted to specific needs, taking advantage of the reusability of all the elements.

Based on the components of an object-oriented message, they recommend transmitting in an I-EDI message a command, and data. Constraints have been added to meet I-EDI requirements.

In this view, the object-oriented approach has been used to model the whole environment in which the transmission of messages takes place. On the other hand, for batch EDI, the concepts of entity and attributes were used to give composites and segments new structures. But as far as the Unicorn proposal about I-EDI is concerned at this stage, we only know that messages will be divided into three parts: command, data elements and constraints. Nothing as yet has been decided about segments and composites. We just know that messages will be small and that data elements will be specific. That is the main reason why the proposal of structuring composites and segments for batch EDI cannot be applied as it is.

If data elements are specific, most of them should be self-supporting, no more qualifiers would be required. However, some of them could still need a code list identifier and a code list responsible agency. Moreover, in order to break down a large data element, a repeating specific data element could be required. In this case, these data elements could be grouped into a "set" as opposed to self-supporting data elements that would be stand-alone.

In this way, such "sets" and stand-alone data elements are of the same level and their function is to provide a specific unit of information.

The reason for this "set" is also explained by the compression and omission techniques used in EDIFACT.

User data, in an I-EDI message, would consist of some "sets" and stand-alone data elements which could be grouped into groups - like segment groups in batch EDI - in order to establish relationships between data.

A data item would still be defined by its position in relation to data around it. Data mapping would be realised using data modelling concepts of objects and attributes: to each trigger data element would correspond an object while all the other depending elements would be attributes.

Although one inconvenience is that, in this way, I-EDI messages are not upward compatible, the advantages of the suggestion for I-EDI are as follows:

- simplicity and consistency are increased since there is only one level of information, specific elements whose relationships can be established only through the grouping of these elements;
- efficiency is taken into account: constraints are transmitted in the message and the use of specific data elements is promoted. Moreover, thanks to the definition of scenarios, methods of parties and the transmission of commands, applications know exactly and promptly what to do with data exchanges.
- Flexibility is also improved in that scenarios can be adapted to special business needs and although parties seem to behave in a similar way, through the denomination of their methods, their implementation can be different.

4. Glossary of Terms

CEN ..... Comité Européen de Normalisation
EDI..... Electronic Data Interchange
EDIFACT ..... Electronic Data Interchange For Administration, Commerce and Transport
I-EDI ..... Interactive EDI
MD8 ..... Tourism, Travel and Leisure Message Development Group
MD9 ..... Health Care Message Development Group
SG ..... Segment Group
UN/EDIFACT..... United Nations Rules for Electronic Data Interchange For Administration, Commerce and Transport

5. References

[1] BSI Standards, CEN
Investigation of syntaxes for existing interchange formats to be used in healthcare
CR 1350:1993
BSI, Belgium, September 1993

[2] Anders Tornqvist
..A View on UN/EDIFACT Building Blocks
EDIFACTs - UN/EDIFACT News for Europe, Issue 2 April 1994, p.11-13.

[3] UN/ECE
EDIFACT - Application level syntax rules - Part 1: Syntax rules common to both batch and interactive EDI - Working Draft, Release 1
WD 9735-1
UN/ECE, UK, 20 September 1994

Conclusion

Throughout this paper, we analysed the UN/EDIFACT Standard and we presented some possible methods for overcoming its deficiencies using the object-oriented approach.

In the process, we studied the use of composite, segment and message structures in EDIFACT. Some deficiencies of EDIFACT were identified. We found that the main problem is the definition of composite and segments structures as having a single function. This leads to multiple uses of composite and segment structures, multifunctional composites and segments as well as overlapping in the use of these two structures. There are also many existing alternative approaches to mapping a set of data and different ways of establishing relationships between data due to synchronic use for codes and data elements, and for composite and segment functions. This can lead to confusion in mapping data and consequently to a variety of message structures. In conclusion, five criteria to be taken into account while elaborating new EDIFACT structures were identified: simplicity, flexibility, stability, consistency and efficiency.

In our attempt to find solutions to the problems, we presented two methods used by message development groups: using the object-oriented approach, they model the data before mapping it into the new message. MD9 established a correspondence between object-oriented and EDIFACT structures and MD8 created a message that could be used to transmit information about any tourism object.

Any effort undertaken by message development groups to model data for mapping into a new message must be encouraged. Of course it will not solve all the problems, and the mapping of an object-oriented model into EDIFACT structures does not go without some hitches, as MD9 demonstrated. But it is undeniably the first step towards better structured messages.

The N. Rasmussen 's proposal for batch EDI, as well as R. Williams' for I-EDI, are more radical since they define new EDIFACT structures in order to

improve the standard. For batch EDI, new data elements were suggested using object-oriented concepts. For I-EDI the whole context of transmission of messages was defined in object-oriented terms. They are the best ways to solve problems.

Implementing that solution though, as far as batch EDI is concerned, may be very expensive since it would question all currently well established batch EDIFACT. In the meantime, lists of recommended segment groups and rules about message structuring into sections could be created. In addition, the idea of using the composite structure only when further clarification for a data element is required could be considered as a guideline to create new composites.

The situation in I-EDI is different since the interactive world is still changing and less well defined. It is thus easier to install new structures. It seems that the members of the Unicorn project are now negotiating and collaborating with the I-EDI group in order to find arrangements for interactive messages.

The overall topic of improving the EDIFACT standard using the object-oriented approach is vast and challenging! We are aware that our paper, with its limits in time and ressources is but a modest contribution and that deeper analysis of a much wider scope would be of great interest. A long-term project with a team of researchers would certainly bring other points into view.

Even if we did not find "the" solution, we hope we have at least shown that current definitions of EDIFACT structures may lead to confusion and ambiguities in mapping data, and that tackling the problem with the help of the object-oriented approach is worthwhile and should be further pursued.

Glossary of terms

EDI.....Electronic Data Interchange
EDIFACT .....Electronic Data Interchange For Administration, Commerce and Transport
I-EDI.....Interactive EDI
MD8.....Tourism, Travel and Leisure Message Development Group
MD9.....Health Care Message Development Group
UN/EDIFACT.....United Nations Rules for Electronic Data Interchange For Administration, Commerce and Transport

Global Glossary of Terms

ANSI.....American National Standard Institute
CDE.....Composite Data Element
CEN.....Comité Européen de Normalisation
DE.....Data Element
EDCD.....EDIFACT Composite Data Element Directory
EDCL.....EDIFACT Code lists
EDED.....EDIFACT Data Element Directory
EDI.....Electronic Data Interchange
EDIFACT.....Electronic Data Interchange For Administration, Commerce and Transport
EDMD.....EDIFACT Standard Message Directory
EDSD.....EDIFACT Segment Directory
I-EDI.....Interactive Electronic Data Interchange
ISO.....International Organisation for Standardisation
MD8.....Tourism, Travel and Leisure Message Development Group
MD9.....Health Care Message Development Group
MDG.....Message Development Group
OO.....Object-Oriented
PAEB.....Pan American EDIFACT Board
SCM.....Scenario Component Message
SDE.....Simple Data Element
SG.....Segment Group
TC.....Technical Committee
TT&L.....Tourism, Travel and Leisure
UN/ECE.....United Nations Economic Commission for Europe
UN/EDIFACT.....United Nations Rules for Electronic Data Interchange For Administration, Commerce and Transport
UN/GTDI.....United Nations Guidelines for Trade Data Interchange
UN/TDED.....United Nations Trade Data Element Directory
UN/TDID.....United Nations Trade Data Interchange Directory
UNCID.....Uniform Rules of Conduct for the Interchange of Trade Data by Teletransmission
UNSM.....United Nations Standard Message
WEEB.....Western European EDIFACT Board

Collected References

BSI Standards, CEN
Investigation of syntaxes for existing interchange formats to be used in healthcare
CR 1350:1993
BSI, Belgium, September 1993

Peter Coad and Edward Yourdon
Object-Oriented Analysis, 2nd edition,
Prentice Hall, USA, 1991

CEN/TC 251
Medical Informatics
N 94-068
CEN, Belgium, May 1994

UN/ECE, I-EDI Group
I-EDI Message Design Guidelines - Draft
UN/ECE, UK, 28 July 1994

Joint Rapporteurs Team
EDITIN - EDIFACT Message TINRSP V. 1.0 - Temporary Version
Wien (Austria), 26-30 September 1994

Ebba Lonaeus & Henning Klwitter
An Alternative Approach to Message Design
SWEDIFACT Finans, Sweden, 26 October 1994

MD8
Minutes of MD8 Meeting
DOC 94.04 MIN.04
UN/ECE, Brussels (Belgium), 7-9 November, 1994

MD8
EDITIN - EDIFACT Messages TINRSP & TINREQ, V 0 - Release 2.0
Additional Information
Brussels (Belgium), 7-9 November, 1994

Niels Rasmussen
Data Modelling and Message Design - Substantive Comments on the UN/EDIFACT Message Design Guidelines
Canada, February 1993

SITPRO and PFA
The UN/EDIFACT Workshop
SITPRO and PFA, U.K., August 1994

David A. Taylor, Ph. D.
Object-Oriented Technology - A Manager's Guide
Addison-Wesley, USA, 1994

Anders Tornqvist
..A View on UN/EDIFACT Building Blocks
EDIFACTs - UN/EDIFACT News for Europe, Issue 2 April 1994, p.11-13.

UN/ECE
Design of UN/EDIFACT Messages: Guidelines and Rules
TRADE/WP.4/R.840/Rev.2
UN/ECE, Geneva (Switzerland), 11 January 1994

UN/ECE
The 92.1 UN/EDIFACT Composite Data Element Directory
UN/ECE, Geneva (Switzerland), 1992

UN/ECE
The 92.1 UN/EDIFACT Segment Directory
UN/ECE, Geneva (Switzerland), 1992

UN/ECE
The 92.1 UN/EDIFACT Message Directory
UN/ECE, Geneva (Switzerland), 1992

UN/ECE
Working Draft - EDIFACT - Application level syntax rules - Part 1: Syntax rules common to both batch and interactive EDI - Working Draft, Release 1
UN/EDIFACT WD 9735-1:1994
UN/ECE, 20 September 1994

UN/ECE
Working Draft - EDIFACT - Application level syntax rules - Part 3: Syntax rules specific to interactive EDI, plus interactive EDI service directories UN/EDIFACT WD 9735-3:1994
UN/ECE, 20 September 1994

UN/ECE, Syntax Development Group
EDIFACT Syntax : Corrigendum
TRADE/WP.4/R.1082/Corr.1
UN/ECE, France, 1 September 1994

Richard Williams
Using Object-Oriented Techniques in EDIFACT : A Briefing Paper
UK, January 1995, Private communication

Richard Williams
Basic Concepts and Definitions for Object-Orientation
UK, January 1995, Private communication

Richard Williams
A Case for Using Object-Oriented Techniques for I-EDI Messages for the UNICORN Project
UK, May 1995, Private communication

Appendices

A

Analysis of UN/EDIFACT Structures

1. The Use of the Composite Structure

Below, for each reason for using the composite structure, the corresponding CDEs are listed.

1.1. To Allow Data to be Expressed in Two Different Ways: Coded and/or in Clear

Below is a list of CDEs that allow a data to be expressed in coded and/or in clear:

1. C002 DOCUMENT/MESSAGE NAME
2. C040 CARRIER
3. C056 DEPARTMENT OR EMPLOYEE DETAILS
4. C100 TERMS OF DELIVERY
5. C110 PAYMENT TERMS
6. C202 PACKAGE TYPE
7. C215 SEAL ISSUER
8. C219 MOVEMENT TYPE
9. C220 MODE OF TRANSPORT
10. C224 EQUIPMENT TYPE AND SIZE
11. C228 TRANSPORT MEANS
12. C240 PRODUCT CHARACTERISTIC
13. C241 DUTY/TAX/FEE TYPE
14. C242 PROCESS TYPE AND DESCRIPTION
15. C244 TEST METHOD
16. C262 REASON FOR CHANGE
17. C273 ITEM DESCRIPTION
18. C288 PRICING GROUP
19. C514 SAMPLE LOCATION DETAILS
20. C515 TEST REASON
21. C517 LOCATION IDENTIFICATION
22. C519 RELATED LOCATION ONE IDENTIFICATION
23. C522 INSTRUCTION
24. C524 HANDLING INSTRUCTION

25. C543 AGREEMENT TYPE IDENTIFICATION
26. C552 ALLOWANCE/CHARGE INFORMATION
27. C553 RELATED LOCATION TWO IDENTIFICATION
28. C556 STATUS REASON
29. C941 RELATIONSHIP
30. C942 MEMBERSHIP CATEGORY
31. C944 MEMBERSHIP STATUS
32. C945 MEMBERSHIP LEVEL
33. C948 EMPLOYMENT CATEGORY
34. C950 QUALIFICATION CLASSIFICATION
35. C951 OCCUPATION
36. C953 CONTRIBUTION TYPE
37. C956 ATTRIBUTE DETAILS
38. C960 REASON FOR CHANGE

1.2. To Repeat the Same Specific Data Element

The following CDEs contain a specific SDE repeated several times:

1. C058 NAME AND ADDRESS
2. C059 STREET
3. C108 TEXT LITERAL
4. C208 IDENTITY NUMBER RANGE
5. C210 MARKS AND LABELS
6. C233 SERVICE
7. C236 DANGEROUS GOODS LABEL

1.3. To Associate a Qualifier with a Generic Data Element

What follows is a list of CDEs which associate a qualifier with a generic DE:

1. C076 COMMUNICATION CONTACT
2. C138 PRICE MULTIPLIER INFORMATION
3. C186 QUANTITY DETAILS
4. C206 IDENTIFICATION NUMBER
5. C223 DANGEROUS GOODS SHIPMENT FLASHPOINT
6. C239 TEMPERATURE SETTING
7. C270 CONTROL
8. C279 QUANTITY DIFFERENCE INFORMATION
9. C507 DATE/TIME/PERIOD
10. C523 NUMBER OF UNIT DETAILS
11. C526 FREQUENCY DETAILS

1.4. To Associate the 1131 and 3055 DEs with a DE

Below is a list of CDEs which group together a DE and the 1131/3055 DEs:

1. C082 PARTY IDENTIFICATION DETAILS
2. C107 TEXT REFERENCE
3. C214 SPECIAL SERVICES IDENTIFICATION
4. C218 HAZARDOUS MATERIAL
5. C229 CHARGE CATEGORY
6. C231 METHOD OF PAYMENT
7. C237 EQUIPMENT IDENTIFICATION

8. C246 CUSTOMS IDENTITY CODES
9. C292 PRICE CHANGE INFORMATION
10. C528 COMMODITY/RATE DETAIL
11. C533 DUTY/TAX/FEE ACCOUNT DETAIL
12. C536 CONTRACT AND CARRIAGE CONDITION
13. C537 TRANSPORT PRIORITY
14. C545 INDEX IDENTIFICATION
15. C549 MONETARY FUNCTION
16. C550 REQUIREMENT/CONDITION IDENTIFICATION
17. C551 BANK OPERATION
18. C554 RATE/TARIFF CLASS DETAIL
19. C555 STATUS EVENT
20. C601 STATUS DETAIL
21. C703 NATURE OF CARGO
22. C901 APPLICATION ERROR DETAIL
23. C955 ATTRIBUTE TYPE

1.5. To Group Together Related DEs in Order to Fulfil the Function Required

Below is the list of the CDEs which contain related DEs:

1. C045 BILL LEVEL IDENTIFICATION
2. C078 ACCOUNT IDENTIFICATION
3. C080 PARTY NAME
4. C088 INSTITUTION IDENTIFICATION
5. C112 TERMS/TIME INFORMATION
6. C128 RATE DETAILS
7. C174 VALUE/RANGE
8. C200 CHARGE
9. C203 RATE/TARIFF CLASS
10. C205 HAZARD CODE
11. C211 DIMENSIONS
12. C212 ITEM NUMBER IDENTIFICATION
13. C213 NUMBER AND TYPE OF PACKAGES
14. C222 TRANSPORT IDENTIFICATION
15. C232 GOVERNMENT ACTION
16. C234 UNDG FLASHPOINT
17. C235 HAZARD IDENTIFICATION
18. C243 DUTY/TAX/FEE DETAIL
19. C280 RANGE
20. C286 SEQUENCE INFORMATION
21. C329 PATTERN DESCRIPTION
22. C401 EXCESS TRANSPORTATION INFORMATION
23. C402 PACKAGE TYPE IDENTIFICATION
24. C501 PERCENTAGE DETAILS
25. C502 MEASUREMENT DETAILS
26. C503 DOCUMENT/MESSAGE DETAILS
27. C504 CURRENT DETAILS
28. C506 REFERENCE
29. C509 PRICE INFORMATION
30. C516 MONETARY AMOUNT
31. C521 BUSINESS FUNCTION
32. C527 STATISTICAL DETAILS
33. C529 PROCESSING INDICATOR
34. C531 PACKAGING DETAILS
35. C532 RETURNABLE PACKAGE DETAILS

36. C534 PAYMENT INSTRUCTION DETAILS
37. C546 INDEX VALUE
38. C701 ERROR POINT DETAILS
39. C849 PARTIES TO INSTRUCTION
40. C850 STATUS OF INSTRUCTION
41. C878 CHARGE ALLOWANCE ACCOUNT

2. The use of the Segment Structure

Below, for each type of composite, the corresponding segments are listed.

2.1. Segments that Contain a Single CDE

Below is a list of segments that contain a single CDE. The CDE tag follows the name of the segment that it is related to.

1. ARD	AMOUNTS RELATIONSHIP DETAILS	C549	*
2. CNT	CONTROL TOTAL	C270	*
3. COM	COMMUNICATION CONTACT	C076	*
4. DIM	DIMENSIONS	C211	*
5. DTM	DATE/TIME/PERIOD	C507	*
6. EQA	ATTACHED EQUIPMENT	C237	
7. EQN	NUMBER OF UNITS	C523	*
8. ERC	APPLICATION ERROR IDENTIFICATION	C901	*
9. GDS	NATURE OF CARGO	C703	
10. GIS	GENERAL INDICATOR	C529	*
11. MOA	MONETARY AMOUNT	C516	*
12. PAI	PAYMENT INSTRUCTION	C534	*
13. PCD	PERCENTAGE DETAILS	C501	*
14. PRC	PROCESS IDENTIFICATION	C242	*
15. QTY	QUANTITY	C186	*
16. REL	RELATIONSHIP	C941	*
17. RFF	REFERENCE	C506	*
18. RNG	RANGE DETAILS	C280	*
19. RTE	RATE DETAILS	C128	*
20. TMP	TEMPERATURE	C239	*
21. TPL	TRANSPORT PLACEMENT	C222	

The segments marked with an asterisk contain a CDE which is used only in that segment.

2.2. Segments that Contain SDEs Only

Below is a list of segments that contain only SDEs. The number in brackets indicates the number of SDEs that the segment includes.

1. AJT	ADJUSTMENT DETAILS	(2)
2. ALI	ADDITIONAL INFORMATION	(7)
3. AUT	AUTHENTICATION RESULT	(2)
4. CPS	CONSIGNMENT PACKING SEQUENCE	(3)
5. DLI	DOCUMENT LINE IDENTIFICATION	(2)
6. DMS	DOCUMENT/MESSAGE SUMMARY	(3)

2.3. Segments that Contain one CDE and one SDE Only

Below is a list of segments which are composed of a single CDE and a SDE only.

1. AGR AGREEMENT IDENTIFICATION
2. FCA FINANCIAL CHARGES ALLOCATION
3. PGI PRODUCT GROUP INFORMATION
4. PRI PRICE DETAILS
5. SEL SEAL NUMBER
6. SEQ SEQUENCE DETAILS
7. SGP SPLIT GOODS PLACEMENT
8. STA STATISTICS

2.4. Other Segments

In the list below, the segments are composed of CDEs and/or SDEs:

1. ALC ALLOWANCE OR CHARGE
2. APR ADDITIONAL PRICE INFORMATION
3. ATT ATTRIBUTE
4. BGM BEGINNING OF MESSAGE
5. BII BILL ITEM IDENTIFICATION
6. BUS BUSINESS FUNCTION
7. CCI CHARACTERISTIC/CLASS IDENTIFICATION
8. CNI CONSIGNMENT INFORMATION
9. COT CONTRIBUTION DETAILS
10. CPI CHARGE PAYMENT INSTRUCTIONS
11. CST CUSTOMS STATUS OF GOODS
12. CTA CONTACT INFORMATION
13. CUX CURRENCIES
14. DGS DANGEROUS GOODS
15. DLM DELIVERY LIMITATIONS
16. DOC DOCUMENT/MESSAGE DETAILS
17. EMP EMPLOYMENT DETAILS
18. EQD EQUIPMENT DETAILS
19. FII FINANCIAL INSTITUTION INFORMATION
20. FTX FREE TEXT
21. GID GOODS ITEM DETAILS
22. GIN GOODS IDENTITY NUMBER
23. GIR RELATED IDENTIFICATION
24. GOR GOVERNMENTAL REQUIREMENTS
25. HAN HANDLING INSTRUCTIONS
26. IMD ITEM DESCRIPTION
27. IND INDEX DETAILS
28. INP PARTIES TO INSTRUCTION
29. LIN LINE ITEM
30. LOC PLACE/LOCATION IDENTIFICATION
31. MEA MEASUREMENTS
32. MEM MEMBERSHIP DETAILS
33. NAD NAME AND ADDRESS
34. PAC PACKAGE
35. PAT PAYMENT TERMS BASIS
36. PCI PACKAGE IDENTIFICATION
37. PIA ADDITIONAL PRODUCT IDENTIFICATION
38. PIT PRICE ITEM LINE

39. PSD PHYSICAL SAMPLE DESCRIPTION
40. QVR QUANTITY VARIANCES
41. RCS REQUIREMENTS AND CONDITIONS
42. SCC SCHEDULING CONDITIONS
43. SPS SAMPLING PARAMETERS FOR SUMMARY STATISTICS
44. STS TRANSPORT STATUS REPORT
45. TAX DUTY/TAX/FEE DETAILS
46. TCC TRANSPORT CHARGE/RATE CALCULATIONS
47. TDT DETAILS OF TRANSPORTS
48. TEM TEST METHOD
49. TMD TRANSPORT MOVEMENT DETAILS
50. TOD TERMS OF DELIVERY
51. TSR TRANSPORT SERVICE REQUIREMENTS

3. Classification of Messages

Below, messages are listed according to their source.

3.1.1. MD1 Trade

Below is the list of the messages developed by MD1 Trade:

1. DELFOR DELIVERY SCHEDULE MESSAGE;
2. DELJIT DELIVERY JUST IN TIME MESSAGE;
3. DESADV DESPATCH ADVICE MESSAGE;
4. INVOIC INVOICE MESSAGE;
5. INVRPT INVENTORY REPORT;
6. ORDCHG PURCHASE ORDER CHANGE MESSAGE;
7. ORDERS PURCHASE ORDER MESSAGE;
8. ORDRSP PURCHASE ORDER RESPONSE MESSAGE;
9. PARTIN PARTY INFORMATION MESSAGE;
10. PRICAT PRICE/SALES CATALOGUE MESSAGE;
11. QALITY QUALITY DATA MESSAGE;
12. QUOTES QUOTE MESSAGE;
13. REMADV REMITTANCE ADVICE MESSAGE[1];
14. REQOTE REQUEST FOR QUOTE MESSAGE.
15. SLSRPT SALES DATA REPORT MESSAGE;
16. STATAC STATEMENT OF ACCOUNT MESSAGE.

3.1.2. MD2 Transport

Below is a list of the messages developed by MD2 Transport:

1. BAPLIE BAYPLAN/STOWAGE PLAN - OCCUPIED AND EMPTY LOCATIONS MESSAGE;
2. BAPLTE BAYPLAN/STOWAGE PLAN - TOTAL NUMBERS MESSAGE;
3. IFCSUM INTERNATIONAL FORWARDING AND CONSOLIDATION SUMMARY MESSAGE;
4. IFTMAN ARRIVAL NOTICE MESSAGE;

[1]This message was developed by MD1 together with MD4.

5. IFTMBC BOOKING CONFIRMATION MESSAGE;
6. IFTMBF FIRM BOOKING MESSAGE;
7. IFTMBP PROVISIONAL BOOKING MESSAGE;
8. IFTMCS INSTRUCTION CONTRACT STATUS MESSAGE;
9. IFTMFR INTERNATIONAL FORWARDING AND TRANSPORT MESSAGE FRAMEWORK;
10. IFTMIN INSTRUCTION MESSAGE;
11. IFTSTA INTERNATIONAL MULTIMODAL STATUS REPORT MESSAGE.

3.1.3. MD3 Customs

Below is a list of the messages developed in association with the Customs Co-operation Council:

1. CUSCAR CUSTOMS CARGO REPORT MESSAGE;
2. CUSDEC CUSTOMS DECLARATION MESSAGE;
3. CUSREP CUSTOMS REPORT MESSAGE.
4. CUSRES CUSTOMS RESPONSE MESSAGE;

3.1.4. MD4 Finance

Below is a list of the messages developed by MD4 Finance:

1. BANSTA BANKING STATUS MESSAGE;
2. CREADV CREDIT ADVICE MESSAGE [1];
3. CREEXT EXTENDED CREDIT ADVICE MESSAGE [1];
4. DEBADV DEBIT ADVICE MESSAGE [1];
5. DIRDEB DIRECT DEBIT MESSAGE;
6. DOCADV DOCUMENTARY CREDIT ADVICE MESSAGE [1];
7. DOCAPP DOCUMENTARY CREDIT APPLICATION MESSAGE [1];
8. DOCINF DOCUMENTARY CREDIT ISSUANCE INFORMATION MESSAGE [1];
9. PAYEXT EXTENDED PAYMENT ORDER MESSAGE [1];
10. PAYMUL MULTIPLE PAYMENT ORDER MESSAGE;
11. PAYORD PAYMENT ORDER MESSAGE;
12. REMADV REMITTANCE ADVICE MESSAGE.

3.1.5. MD5 Construction

Below is a list of the messages developed by MD5 Construction:

1. CONDPV DIRECT PAYMENT VALUATION MESSAGE;
2. CONEST ESTABLISHMENT OF CONTRACT MESSAGE;
3. CONITT INVITATION TO TENDER MESSAGE;
4. CONPVA PAYMENT VALUATION MESSAGE;
5. CONQVA QUANTITY VALUATION MESSAGE;
6. CONTEN TENDER MESSAGE.

---

[1]These messages were developed by MD4 together with the Pan American EDIFACT Board (PAEB).

3.1.6. Australia/New Zealand EDIFACT Board

Below is a list of the messages developed by The Australian/New Zealand EDIFACT Board:

1. PAXLST PASSENGER LIST MESSAGE;
2. PAYDUC PAYROLL DEDUCTIONS ADVICE MESSAGE;
3. SUPCOT SUPERANNUATION CONTRIBUTIONS ADVICE MESSAGE;
4. SUPMAN SUPERANNUATION MAINTENANCE MESSAGE.

B

Data Modelling to Design EDIFACT Messages

1. Health Care Messages

Below are the definitions of the ATT, PID, PRO, PTY and SPE segments[1].

ATT ATTRIBUTE
Function: This segment describes an attribute relating to a person, object, product, etc.

Zzzz	ATTRIBUTE QUALIFIER	M	an..3
Czzz	ATTRIBUTE DETAILS	C	
Zzzz	Attribute, coded	C	an..3
1131	Code list qualifier	C	an..3
3055	Code list responsible agency	C	an..3
Zzzz	Attribute	C	an..35

PID PERSON IDENTIFIER
Function: To identify personal name, title and sex.

3035	PARTY QUALIFIER	M	an..3

[1] The definitions of these segments come from MEDRPT, Medical Service Report Message (TRADE/WP.4/R.815).

C082	PARTY IDENTIFICATION DETAILS	C	
3039	Party ID Identification	M	an..17
1131	Code list qualifier	C	an..3
3055	Code list responsible agency	C	an..3
Czzz	PERSON NAME	C	
Zzzz	Name format qualifier	M	an..3
Zzzz	Name component	C	an..35
Zzzz	Name component	C	an..35
Zzzz	Name component	C	an..35
Zzzz	Name component	C	an..35
Zzzz	Name component	C	an..35
Zzzz	Name component	C	an..35
Zzzz	Name component	C	an..35
Zzzz	Name component	C	an..35
Czzz	SEX DETAILS	C	
Zzzz	Sex, coded	C	an..3
1131	Code list qualifier	C	an..3
3055	Code list responsible agency	C	an..3
Zzzz	Sex	C	an..35
Czzz	TITLED DETAILS	C	
Zzzz	Titled, coded	C	an..3
1131	Code list qualifier	C	an..3
3055	Code list responsible agency	C	an..3
Zzzz	Titled	C	an..35

PRO PROCEDURAL INFORMATION
Function: To identify a set of planned or performed procedures.

Zzzz	PROCEDURE TYPE QUALIFIER	C	an..3
Czzz	PROCEDURE DETAILS	C	
Zzzz	Procedure identification	C	an..8
1131	Code list qualifier	C	an..3
3055	Code list responsible agency	C	an..3
Zzzz	Procedure	C	an..35
Zzzz	Procedure	C	an..35

PTY PRIORITY
Function: The segment is used to communicate priority information

Zzzz	PRIORITY TYPE QUALIFIER	C	an..3

Czzz	PRIORITY DETAILS	C	
Zzzz	Prioriry, coded	C	an..3
1131	Code list qualifier	C	an..3
3055	Code list responsible agency	C	an..3
Zzzz	Priority	C	an..35

SPE SPECIMEN DESCRIPTION
Function: To identify a physical specimen.

Czzz	SPECIMEN QUALIFIER	M	an..3
Czzz	SPECIMEN DESCRIPTION DETAILS	C	
Zzzz	Specimen description identification	C	an..8
1131	Code list qualifier	C	an..3
3055	Code list responsible agency	C	an..3
Zzzz	Specimen description	C	an..35
Czzz	SPECIMEN LOCATION DETAILS	C	
Zzzz	Specimen location identification	C	an..8
1131	Code list qualifier	C	an..3
3055	Code list responsible agency	C	an..3
3236	Specimen location	C	an..35
Czzz	SPECIMEN SELECTION METHOD DETAILS	C	
Zzzz	Specimen selection method identification	C	an..8
1131	Code list qualifier	C	an..3
3055	Code list responsible agency	C	an..3
Zzzz	Specimen selection method	C	an..35
Czzz	SPECIMEN STATUS, CODED	C	an..3

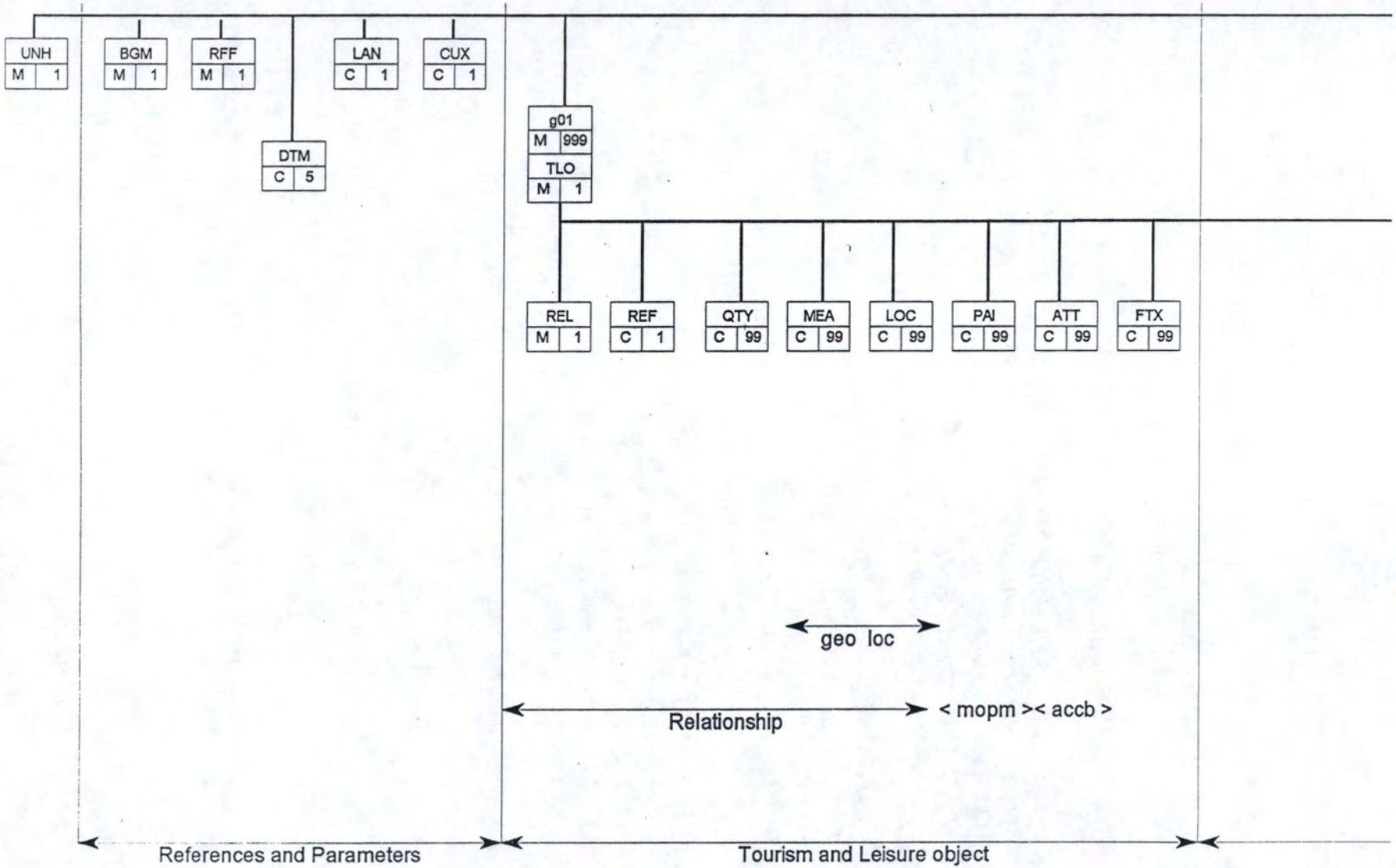
---

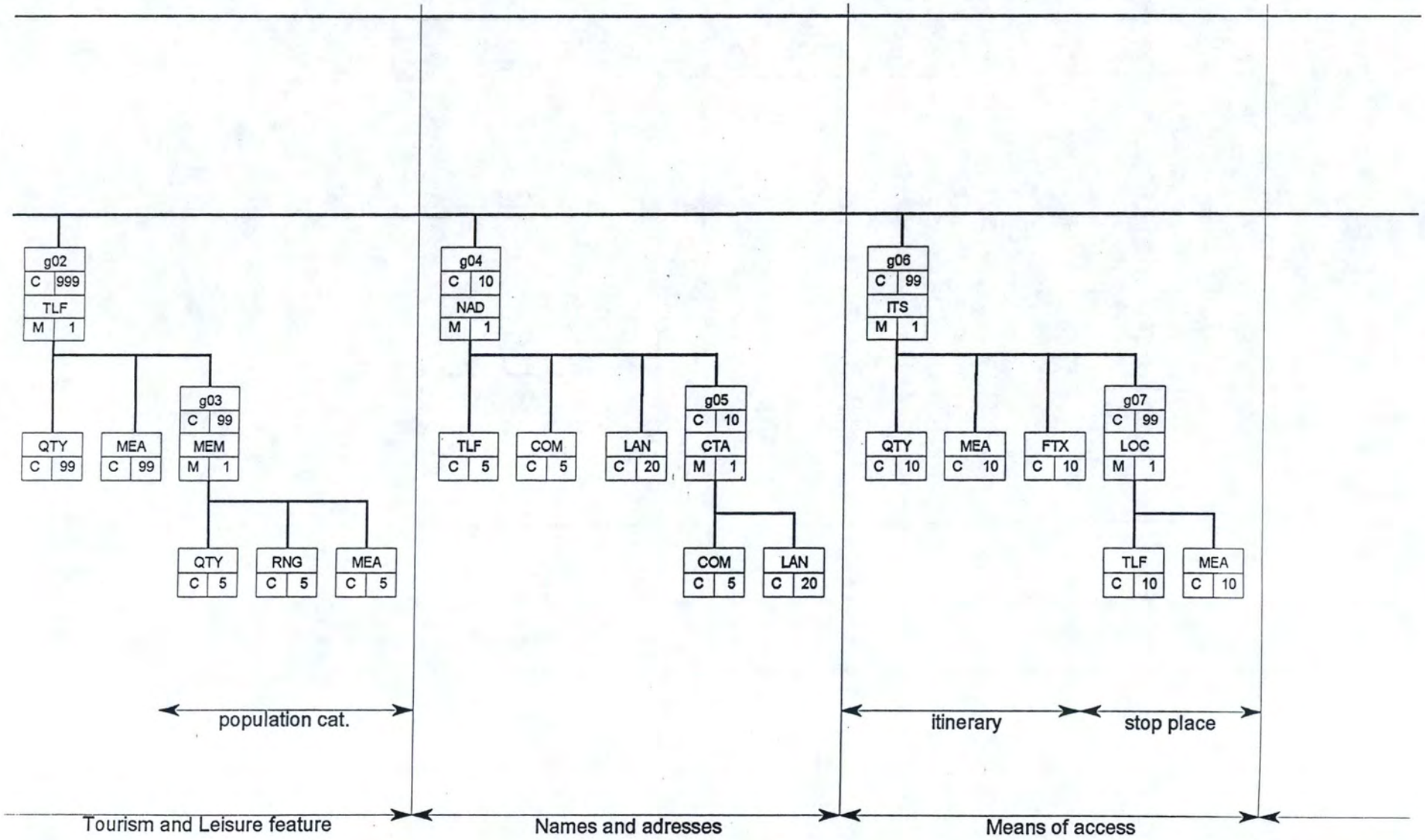
2. Branching Diagram of TINRSP Tourism Information Response Message

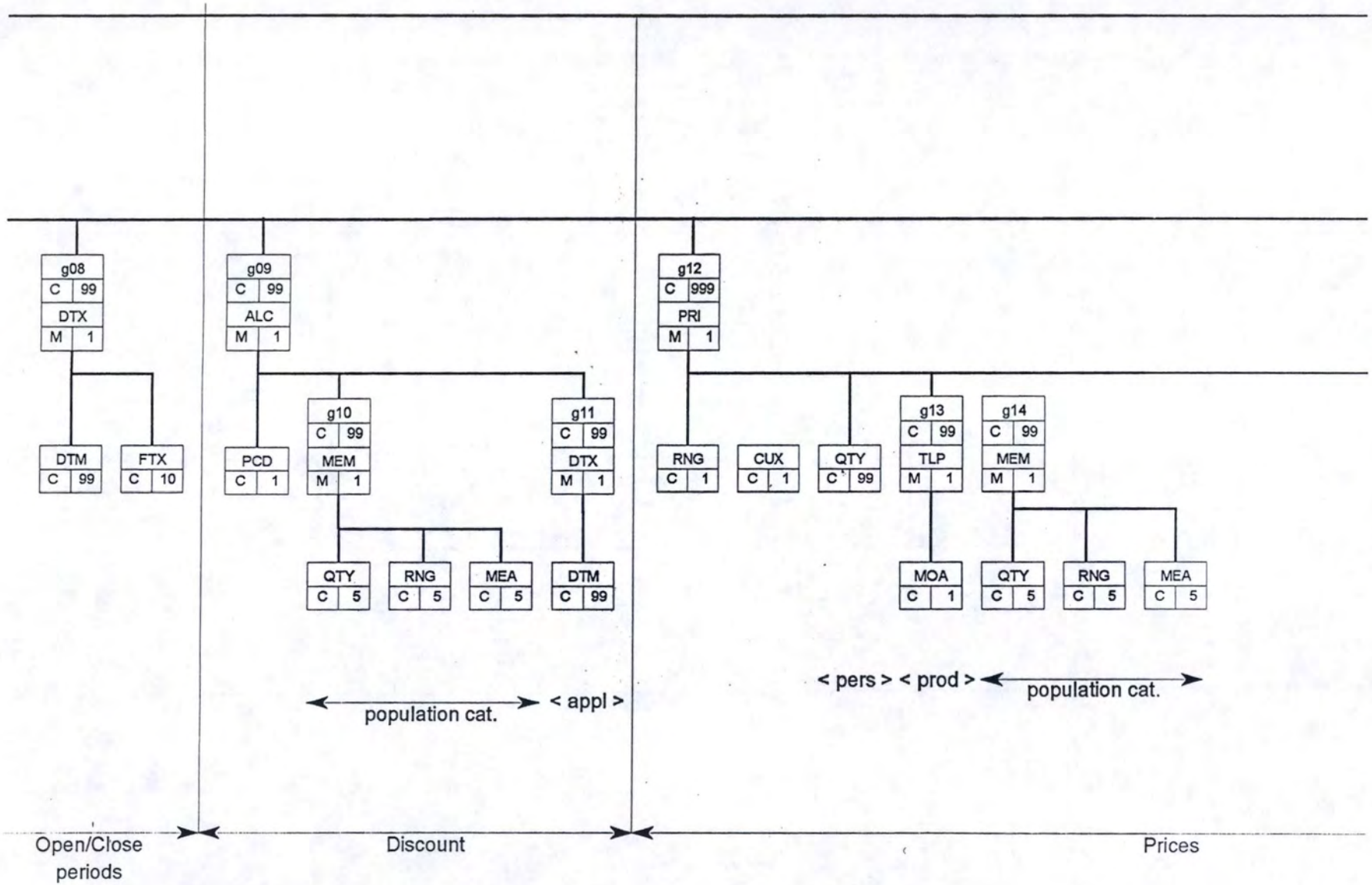
What follows is the branching diagram of TINRSP, Tourism Information Response Message.[1]

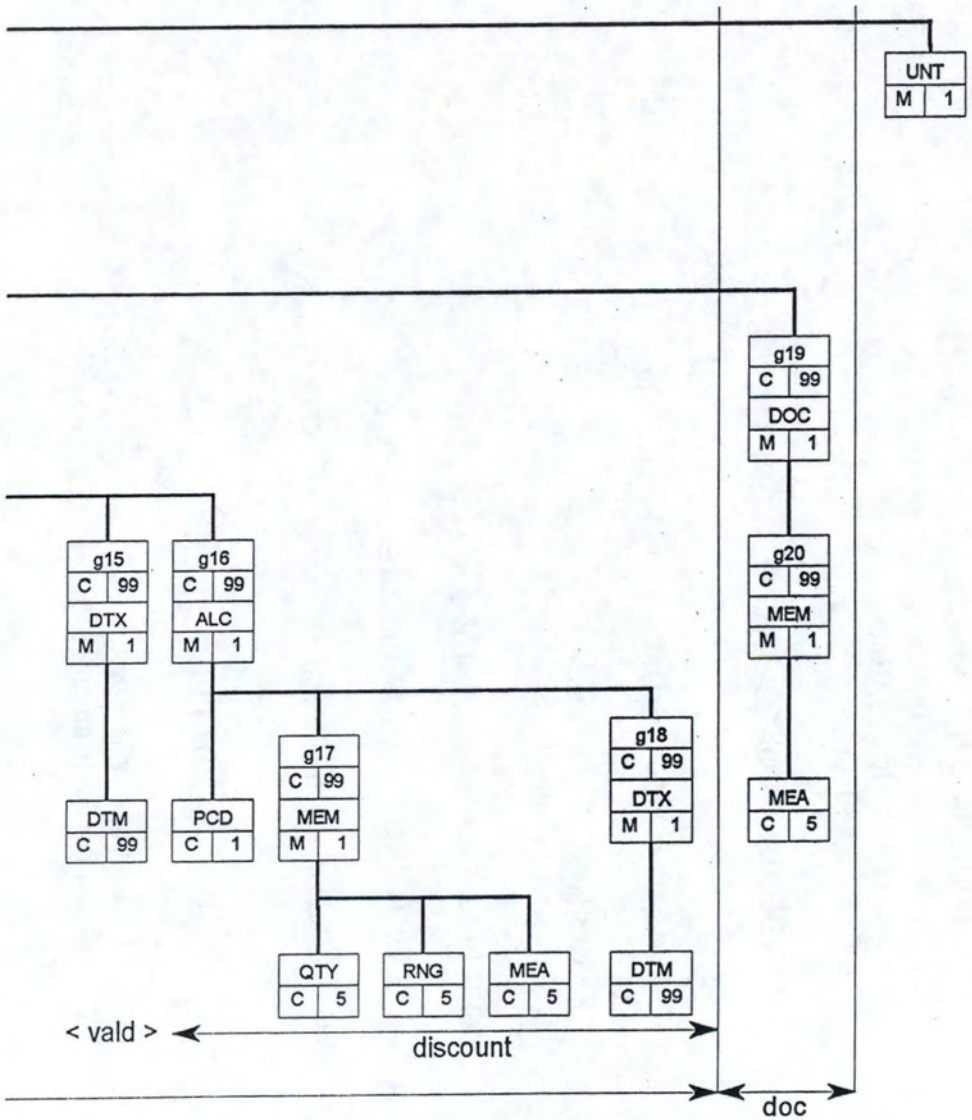
---

[1] MD8
Minutes of MD8 Meeting
DOC 94.04 MIN.04
UN/ECE, Brussels (Belgium), 7-9 November, 1994


UNH
M 1
BGM
M 1
RFF
M 1
DTM
C 5
LAN
C 1
CUX
C 1
g01
M 999
TLO
M 1
REL
M 1
REF
C 1
QTY
C 99
MEA
C 99
LOC
C 99
PAI
C 99
ATT
C 99
FTX
C 99
geo loc
Relationship
< mopm >< accb >
References and Parameters
Tourism and Leisure object


g02
C 999
TLF
M 1
QTY
C 99
MEA
C 99
g03
C 99
MEM
M 1
QTY
C 5
RNG
C 5
MEA
C 5
g04
C 10
NAD
M 1
TLF
C 5
COM
C 5
LAN
C 20
g05
C 10
CTA
M 1
COM
C 5
LAN
C 20
g06
C 99
ITS
M 1
QTY
C 10
MEA
C 10
FTX
C 10
g07
C 99
LOC
M 1
TLF
C 10
MEA
C 10
population cat.
itinerary
stop place
Tourism and Leisure feature
Names and adresses
Means of access


g08
C 99
DTX
M 1
DTM
C 99
FTX
C 10
g09
C 99
ALC
M 1
PCD
C 1
g10
C 99
MEM
M 1
QTY
C 5
RNG
C 5
MEA
C 5
g11
C 99
DTX
M 1
DTM
C 99
< appl >
population cat.
g12
C 999
PRI
M 1
RNG
C 1
CUX
C 1
QTY
C 99
g13
C 99
TLP
M 1
MOA
C 1
g14
C 99
MEM
M 1
QTY
C 5
RNG
C 5
MEA
C 5
< pers > < prod >
population cat.
Open/Close periods
Discount
Prices


UNT
M 1
g19
C 99
DOC
M 1
g20
C 99
MEM
M 1
MEA
C 5
g15
C 99
DTX
M 1
DTM
C 99
g16
C 99
ALC
M 1
PCD
C 1
g17
C 99
MEM
M 1
QTY
C 5
RNG
C 5
MEA
C 5
g18
C 99
DTX
M 1
DTM
C 99
< vald >
discount
doc

3. Specifications of TLF & TLO Segments

3.1. TLF, TOURISM AND LEISURE FEATURES

Below are the specifications of the TLF, TOURISM AND LEISURE FEATURES[1].

TLF TOURISM AND LEISURE FEATURES

Function: To describe features related to a tourism and leisure object, such as facilities, services, installations.

CXXC TOURISM AND LEISURE FEATURES	M		
9913 Tourism and leisure feature type, coded	C	an	..3
9911 Tourism and leisure feature, coded	C	an	..3
9912 Tourism and leisure feature	C	an	..70

9915 TOURISM AND LEISURE FEATURES MODE OF USE, CODED
C an ..3

Examples of 9913 Tourism and leisure feature type, coded values:
XX4 Type of cooking
XX5 Restaurant chef's name
XX6 Restaurant speciality
XX7 Service to customers
XXB Inside installation & equipment

3.2. TLO, TOURISM AND LEISURE OBJECT

Below are the specifications of the TLO, TOURISM AND LEISURE OBJECT[1].

TLO TOURISM AND LEISURE OBJECT

Function: To identify a tourism and leisure object, such as an accomodation, a food and drink establishment, a theme park, an archeological site, an activity, a museum, etc.

---

[1] MD8
EDITIN - EDIFACT Messages TINRSP & TINREQ, V 0 - Release 2.0
Additional Information
Brussels (Belgium), 7-9 November, 1994

CXXD TOURISM AN LEISURE OBJECT	M		
9917 Tourism and leisure object family, coded	M	an	..3
9919 Tourism and leisure object type, coded	C	an	..3
9910 TOURISM AND LEISURE OBJECT	USUAL NAME		
	C	an	..70

Examples of 9917 Tourism and leisure object family, coded values:
XX1 Accommodation
XX2 Activity
XX3 Food & Drink
XX4 Theme Park

Examples of 9919 Tourism and leisure object type, coded values if the tourism object family specified in 9917 is "accommodation":
X1B Apartment
X1C B&B
X1G Camping site
X1Q Hotel
X1Z University campus

C

TT&L Scenario for Reservations

1. TT&L Scenario for Reservations

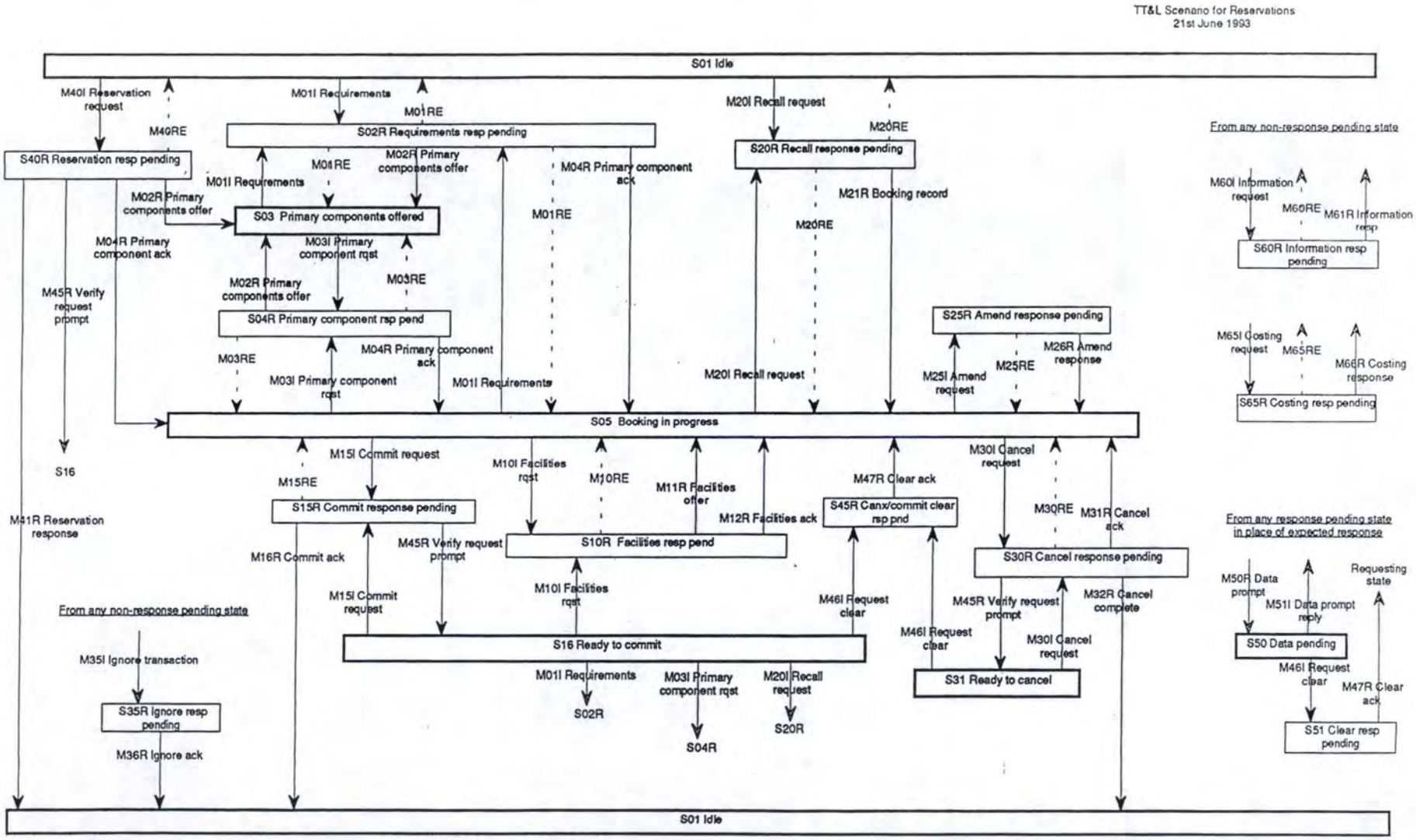
What follows is the tourism, travel and leisure scenario for reservations documented as a state transition diagrams[1]. We draw attention to the fact that it is not the Unicorn scenario.

State and message reference numbers are of the form:

S = state;
R = response pending;
M = message;
I = sent by initiator;
R = sent by responder;
RE = error response.

Non-response pending states (i.e. control with initiator) are shown in bold.

[1]MD8
Minutes of MD8 Meeting
DOC 94.04 MIN.04
UN/ECE, Brussels (Belgium), 7-9 November, 1994


TT&L Scenario for Reservations
21st June 1993
S01 Idle
M40I Reservation request
M40RE
S40R Reservation resp pending
M01I Requirements
M01RE
S02R Requirements resp pending
M01I Requirements
M01RE
M02R Primary components offer
M02R Primary components offer
S03 Primary components offered
M04R Primary component ack
M04R Primary component ack
M03I Primary component rqst
M03RE
M02R Primary components offer
S04R Primary component rsp pend
M03RE
M03I Primary component rqst
M04R Primary component ack
M01I Requirements
M01RE
M45R Verify request prompt
S16
M41R Reservation response
M20I Recall request
M20RE
S20R Recall response pending
M21R Booking record
M20RE
M20I Recall request
S25R Amend response pending
M25I Amend request
M25RE
M26R Amend response
S05 Booking in progress
M15I Commit request
M15RE
S15R Commit response pending
M16R Commit ack
M45R Verify request prompt
M15I Commit request
M10I Facilities rqst
M10RE
M11R Facilities offer
M12R Facilities ack
S10R Facilities resp pend
M10I Facilities rqst
S16 Ready to commit
M01I Requirements
S02R
M03I Primary component rqst
S04R
M20I Recall request
S20R
M47R Clear ack
S45R Canx/commit clear rsp pnd
M46I Request clear
M46I Request clear
M30I Cancel request
M30RE
M31R Cancel ack
S30R Cancel response pending
M45R Verify request prompt
M32R Cancel complete
M30I Cancel request
S31 Ready to cancel
From any non-response pending state
M35I Ignore transaction
S35R Ignore resp pending
M36R Ignore ack
From any non-response pending state
M60I Information request
M60RE
M61R Information resp
S60R Information resp pending
M65I Costing request
M65RE
M66R Costing response
S65R Costing resp pending
From any response pending state in place of expected response
M50R Data prompt
M51I Data prompt reply
Requesting state
S50 Data pending
M46I Request clear
M47R Clear ack
S51 Clear resp pending
S01 Idle

D

The Repetition Technique

1. The Repetition Technique

Implementing the repetition technique[1] implies that fewer separators are required. The example below illustrates it.

ABC EXAMPLE SEGMENT

1234	FIRST DATA ELEMENT	M	3	an..3
C123	COMPOSITE ONE	M	2	
0111	First element of C123	M		an..3
0222	Second element of C123	C		an..3
0333	Third element of C123	C		an..3
C456	COMPOSITE TWO	C	1	
1111	First element of C456	M		an..3
2222	Second element of C456	C		an..3
3333	Third element of C456	C		an..3

[1] UN/ECE
EDIFACT - Application level syntax rules - Part 1: Syntax rules common to both batch and interactive EDI - Working Draft, Release 1
UN/EDIFACT WD 9735-1:1994
UN/ECE, 20 September 1994

If only data (xxx) for data element 1111 of C456 must be transmitted, and

- if the repetition technique is not used, the data stream would contain:

  ABC++++++xxx '

- if the repetition technique is used, the data stream would contain:

  ABC+++xxx '