UNITED NATIONS **E**

Economic and Social Council

Distr.
RESTRICTED

TRADE/WP.4/R.840/Rev.3
31 January 1997
ENGLISH ONLY

ECONOMIC COMMISSION FOR EUROPE

COMMITTEE ON THE DEVELOPMENT OF TRADE

Working Party on Facilitation of International Trade Procedures

(Item 3 of the provisional agenda of the Meetings of Experts on Data Elements and Automatic Data Interchange (GE.1) Fifty-fifth session, 19-20 March 1997)

UN/EDIFACT
Message Design Rules for Batch EDI

* * *

Submitted by the Message Design Rules (MDR) Group*

This document was prepared and submitted by the MDRG as agreed at the September 1996 session.

The Group of Experts is expected to:

approve this document.

* The present document is reproduced in the form in which it was received by the secretariat.

GE.97-

UN/EDIFACT
Message Design Rules for Batch EDI

Release 1
1997-03-17

Contents

		Page
Foreword		4
1	**Introduction**	8
2	**Scope**	8
3	**References**	8
4	**Message Design Rules**	10
4.1	General	10
4.2	Messages	10
4.3	Sectionalised messages	11
4.4	Segment groups	11
4.5	Segments	11
4.6	Composite data elements	13
4.7	Simple data elements	15
4.8	Identification of external code lists	16
4.9	Code values	16
Annex A	Definitions	17
Annex B	Rules for naming data elements and segments	21
Annex C	Naming examples for the data element directory	25
Annex D	A model of the message design process	27

Foreword

i) Background

During the fiftieth session of the Meeting of Experts (UN/ECE/TRADE/WP.4/GE.1) on Data Elements and Automatic Data Interchange the UN/EDIFACT Message Design Guidelines (MDG) Group submitted its first formal report to the GE.1 (TRADE/WP.4/R.1083). This document, approved at that session, defined the tasking for the MDG Group as:

a) the review of the documents set forth in the Terms of Reference, and additional comments received, thereto,

b) provide a draft recommended Revision 3 to the existing Message Design Guidelines (R.840/Rev.2) including those comments received that remove inconsistencies to that document,

c) development of a new MDG draft document that includes new concepts and procedures once general consensus has been reached on those items through procedures developed by the MDG Group, and

d) development of comprehensive procedures to maintain newly developed MDG versions.

R.1083 set forth the organizational structure and working procedures for the two MDG Group teams: Draft and Edit Team (DET) and Quality Assurance Team (QAT) as well as the time schemes for accomplishment of the above listed tasking.

The DET has produced drafts of R.840 which were reviewed by the QAT and this has resulted in QAT raising two fundamental questions which they felt had to be answered in the revision of R.840. These questions were :

a) Is it possible for the message designer to be able to differentiate clearly, and consistently, between knowing when to design a segment and when to design a composite data element?

b) Can UN/EDIFACT take advantage of the naming and defining principles described in the ISO 11179 (Specification and standardization of data elements) standard?

The DET addressed the QAT request for clarification and as a result redrafted the Message Design Guidelines. This redraft resulted in the document contents being confined to only the rules for message design rather than being a combination of these rules and explanatory guidelines for message design. Accordingly, the document was renamed ‘Message Design Rules’ (MDR).

The draft Message Design Rules document (TRADE/WP.4/CRP.40/Add.1) was distributed for regional review following the March 1996 session of WP.4/GE.1. The comments received from the regions, message design and technical groups, and individual submissions were subsequently processed by the DET and incorporated, as appropriate, in this final draft.

This preface explains the background and philosophy behind the decisions which resulted in this document. The analysis also led to the specification of a model of the design process which can be found in Informative Annex D.

ii) Segment and composite design philosophy

In order to develop a set of consistent rules that distinguish between when to design a segment and when to design a composite data element, the UN/EDIFACT directories were first analysed to determine the approach used for the design of existing segment and composite data element structures. In these directories there are many examples of simple, concise structures and many examples of complex, hybrid structures which highlighted that the design principles lacked precision and consistency. Further reference sources were examined and included documents on database design, object oriented methodology and normalisation practices. This led to the following conclusions :

a) Two of the fundamental precepts of message design are the provision of definitions for the meaning of the data and the extraction of a name directly from these definitions to label the data.

b) A message represents the structure for the data of a defined information flow between two parties.

c) Data elements represent an elementary unit of information. For pragmatic reasons, UN/EDIFACT provides within the message structure, a hierarchical series of subordinate structures based on specific data elements. These subordinate structures are; the simple data element, the composite data element, the segment and the segment group.

d) Two data element structures can be identified within UN/EDIFACT to represent a single discrete unit of data. The first structure is the simple data element, which identifies a single unit of data whose characteristics are entirely defined by the specification of the data element within the UN/EDIFACT directories. The second structure is the composite data element, which can identify a single unit of data whose characteristics are not entirely specified within the UN/EDIFACT directories (i.e. non-UN/EDIFACT code lists) or whose representation characteristics can be provided in two different forms (i.e. coded and/or non-coded) or whose characteristics are defined by an aggregate of two data elements (i.e. a qualifier data element and a simple data element).

e) The rules contained herein for the design of data elements are consistent with the design philosophy of providing only two data element structures. In order to align current design practices with this philosophy, the design of a composite data element structure is limited to one of the eight formats specified in these rules.

f) The design rule which restricted the specification of a textual data element to a maximum length of an..17, an..35 or an..70 has been removed. This means that message designers can specify the maximum length of a data element that is appropriate for the data element (e.g. an..128). A maximum length must be specified for each data element.

g) Segments and segment groups are the basic building blocks of messages. A segment represents a functionally related set of data elements whose characteristics define a single distinct concept (e.g. a party, a place, a product, a service, a document, etc.). For pragmatic reasons, UN/EDIFACT has recommended that the approach to segment design be generic rather than specific.

h) The design of a generic segment provides a means for the segment to be used across a wide range of applications by having each specific instance of usage specified in the qualifier data element. This qualifier data element is the first data element in the segment and enables the usage context of the segment to be defined during message implementation.

i) For reasons of consistency, the segment design rules now require that all segments be specified with a qualifier data element. This qualifier data element can be specified with a status of conditional or a status of mandatory.

j) A segment structure consists of a qualifier data element and one or more additional data elements. All data elements specified within the segment represent the totality of characteristics required to define a single distinct concept. Therefore all data elements within the segment must directly relate to the concept represented by the segment. In the case where a data element represents a characteristic that is only partially related to the concept, a separate segment is required for this data element.

k) This rationalisation process, when carried out correctly, will not permit the same characteristic to be assigned more than once to the same concept. This means that a segment structure cannot have the same data element specified more than once within the segment structure. The only exception to this rule is the use of a qualified composite data element, since its qualifier data element may be used to assign different characteristics to the qualified composite data element.

l) One of the purposes of the segment definition is to describe the concept that is represented at a certain level of abstraction. The level of abstraction is normally dependent upon the information model which is used as the basis for developing a UN/EDIFACT message. Within the UN/EDIFACT directories there are segments with a broad level of abstraction (e.g. ATT - Attribute) and segments with narrower levels of abstraction (e.g. DOC - Document/ Message Details). When determining whether a segment duplicates or overlaps the functionality of another segment, it is important to compare only segments that have been developed at the same level of abstraction. It is not valid, for example, to compare the DOC segment with the ATT segment since the latter is so broad in terms of its level of abstraction that it could encompass all segments at a narrower level.

   For pragmatic reasons, it is recommended that a limit is established regarding the maximum broadest level of abstraction for segments. Since the object class term represents the dominant area of interest it also can be used to determine the level of abstraction. As a guideline, therefore, the maximum broadest level of abstraction is reached when the object class term is a word that equates to one of the words taken from the list of representation terms (e.g. quantity, rate, amount). Note that the ATT segment is broader than this maximum limit, but there is no intention to apply these principles retroactively.

m) Two message sub-structures can be identified within UN/EDIFACT to represent a single distinct concept. The first sub-structure is the stand-alone segment, which identifies a single concept whose characteristics are entirely defined by the data elements specified in the UN/EDIFACT directories for that segment. The second sub-structure is the segment group, which can identify a single concept whose characteristics are entirely defined by the data elements specified in the UN/EDIFACT directories for those segments within the segment group.

n) Within UN/EDIFACT, a segment group may also be used to define a dependency between one segment and another or to define a hierarchical relationship between segments. In these instances, the segment group represents the broad concept and the segment(s) within the segment group and the subordinate segment group(s) represent narrower concepts. Each narrower concept inherits all of the characteristics of the broad concept plus at least one additional distinguishing characteristic, which serves to differentiate the narrower concepts at the same level of abstraction.

o) A trigger segment is the first segment in a segment group and enables the usage context of the segment group to be defined during message implementation.

iii) Naming and defining

Following considerable analysis of ISO 11179 and in particular of ISO/IEC IS 11179-5 (Naming and identification principles for data elements), the adoption of section 6 (and informative Annex A) as part of the UN/EDIFACT message design rules is recommended.

In order to apply these principles fully in the UN/EDIFACT environment, the following adaptations were required:

a) The principles were extended to cover not only the naming of simple data elements but also to cover the naming of composite data elements and segments.

b) The use of qualifier terms in a data element name was modified to align this with the use of qualifier data elements in UN/EDIFACT.

c) The existing UN/EDIFACT term ‘,coded’ at end of the name of a simple data element has been replaced by the representation term of ‘code’ in accordance with ISO 11179-5.

Since business terms frequently form part of the name of a code value, the ISO 11179-5 naming convention is not considered appropriate for the structuring of names for code values.

It is recommended that the term ‘definition’ is used throughout the UN/EDIFACT directories in place of terms segment ‘function’, data element ‘description’, code value ‘description’, etc. This recommendation is in line with ISO 11179.

iv) Summary

In order to differentiate between a segment and a composite data element, this document explains what each construct should represent. A standard naming convention has also been introduced to enable the names of data elements and segments to be constructed in a consistent manner. This has required the introduction of terminology which is not present in R.840/Rev.2. Furthermore, it should be recognised that the Message Design Rules is a document intended for the message designer and the Technical Assessment Groups and is not aimed at being a tutorial or guideline for the novice.

The adoption of a more precise, consistent and coherent set of design principles will ensure that the data directories are of the highest standard, thus meeting the growing requirements of the UN/EDIFACT user community.

UN/EDIFACT Message Design Rules for Batch EDI

1 Introduction

The UN/EDIFACT Message Design Rules for Batch EDI have been prepared by the UN/EDIFACT (United Nations, Electronic Data Interchange for Administration, Commerce and Transport) Message Design Rules (MDR) Group. They have been developed on behalf of Message Design Groups and Technical Assessment Groups who are responsible for the development of UN/EDIFACT messages. They presume a knowledge of the UN/EDIFACT process and procedures and the EDIFACT Application Level Syntax Rules (ISO 9735). The Message Design Rules are specific to message design and intentionally exclude procedures and other specifications for related areas such as information modeling, directory production and message implementation.

The rules have been divided into a number of separate sections that follow the hierarchical structure of a message. Where a rule contains a term that appears in italics, the term is defined in Annex A (Definitions).

It should be noted that, unless the text refers specifically to a simple data element or to a composite data element, the term ‘data element’ applies to both a simple data element and a composite data element.

The Normative Annex B (Rules for naming data elements and segments) forms an integral part of this document and contains the naming convention for data elements and segments. This Annex includes a list of representation terms that shall be used in conjunction with the naming rules.

The Informative Annex C (Naming examples for the data element directory) is included for reference purposes only. It provides an example list of simple data elements and illustrates how the naming convention outlined in Annex B may be applied.

The Informative Annex D (A model of the message design process) is included to identify the area in the message development and implementation process where the message design rules are applied.

2 Scope

This document defines a mandatory set of rules that shall be used for the design and technical assessment of UN/EDIFACT messages and message components. The rules are designed to establish a consistent and objective basis for the design of messages which conform to the UN/EDIFACT Application Level Syntax Rules, versions 1,2 and 3.

3 References

The following standards contain provisions which, through reference in this document, constitute provisions of the rules contained herein.

ISO 646 Information processing - ISO 7-bit coded character set for information interchange

UN/EDIFACT Message Design Rules for Batch EDI

ISO 9735 Electronic data interchange for administration, commerce and transport (EDIFACT) - Application level syntax rules (Versions 1, 2 and 3)

ISO 11179-5 Information technology - Specification and standardization of data elements, Part 5 - Naming and identification principles for data elements

4 Message Design Rules

4.1 General

Rule 1: *Message*, *segment* and *data element* design shall not include UN/EDIFACT directory items marked for deletion.

Rule 2: *Segment* and *data element* design shall not include *syntax service directory* items.

Rule 3: *Message* design shall not include *syntax service directory* items, with the exception of the *Message Header* (UNH), Section Control (UNS) and *Message Trailer* (UNT) *service segments*.

4.2 Messages

Rule 4: The business function of a *message,* as specified in the *message definition,* shall not duplicate and shall not be represented by the business function of another *message*.

Rule 5: A *message* shall be uniquely identified by a *message type* code which shall be six alphabetic uppercase characters from the ISO 646 character set.

Rule 6: The name of a *message* shall be unique.

Rule 7: The name of a *message* shall be consistent with its *message definition*.

Rule 8: A *message* shall start with the *Message Header* (UNH) *service segment* and shall end with the *Message Trailer* (UNT) *service segment*.

Rule 9: A *message* shall be specified with at least one *segment* between the *Message Header* (UNH) *service segment* and the *Message Trailer* (UNT) *service segment*, that is not the Beginning of Message (BGM) *segment* or the Section Control (UNS) *service segment.*

Rule 10: A *message* shall be structured without *segment collision*.

Rule 11: The Beginning of Message (BGM) *segment* shall be the first *segment* that is not a service *segment*. It shall be a *stand-alone segment*, with a *status* of *mandatory* and a maximum number of occurrences of one.

Rule 12: The use of each *segment* and each *segment group* within a *message* shall be specified in the *data segment clarification section.*

Rule 13: The specification of the use of each *segment* in the *data segment clarification section* shall be consistent with its *segment definition* in the UN/EDIFACT *Segment Directory*.

Rule 14: The specification of the use of each *segment* and *segment group* in the *data segment clarification section* shall be consistent with its *status* and maximum number of occurrences as defined in the *segment table*.

Rule 15: The *segment table* structure shall be consistent with the *message definition*.

Rule 16: A *segment* or *segment group* specified in the *segment table* shall have a *status* of either *mandatory*, or *conditional*.

Rule 17 A *segment* or *segment group* specified in the *segment table* shall be assigned a maximum number of occurrences.

4.3 Sectionalised messages

Rule 18: The Section Control (UNS) *service segment* shall only be specified to prevent *segment collision* between the *segments* contained in one section of the *message* and the *segments* contained in the next section of the *message*. It shall be specified to separate these sections of the *sectionalised message* and shall have a maximum occurrence of one, a *status* of *mandatory* and shall appear as a *stand-alone segment* at the beginning of the detail section and/or the summary section.

Rule 19: The sections of a *sectionalised message* shall be identified in both the *data segment clarification section* and the *segment table*, and shall comprise one of:

- header section and detail section
- header section, detail section and summary section
- detail section and summary section

4.4 Segment groups

Rule 20: A *segment group* shall start with a *trigger segment*.

Rule 21: A *trigger segment* shall have a *status* of *mandatory.*

Rule 22: A *trigger segment* shall have a maximum number of occurrences of one.

Rule 23: A *segment group* shall contain at least one other *segment* or *segment group* in addition to the *trigger segment*.

4.5 Segments

Rule 24: A *segment* shall be identified by a *segment tag* which shall be three alphabetic uppercase characters from the ISO 646 character set. It shall not begin with the letter ‘U’.

Rule 25: The *segment tag* shall be unique within the UN/EDIFACT *Segment Directory.*

Rule 26: The *segment definition* shall:

- describe the purpose of the *segment*,
- be consistent with its *data element definitions,*
- not embed its *data element definitions* within it,
- be unique within the UN/EDIFACT *Segment Directory*,
- expand acronyms on the first occurrence,
- not contain abbreviations,
- not contain a gender bias,
- not contain the phrase ‘self explanatory’, ‘to be defined’, ‘to be provided’, or synonyms thereof.

Rule 27: The name of a *segment* shall be unique within the UN/EDIFACT *Segment Directory*.

Rule 28: The name of a *segment* shall be consistent with its *segment definition*.

Rule 29: The purpose of a *segment* shall not duplicate, and shall not be represented by, the purpose of another *segment* at the same *level of abstraction*.

Rule 30: A *segment* shall not contain the entire contents of another *segment*.

Rule 31: A *segment* shall be specified with a *qualifier data element* as the first *data element* within a *segment.*

Rule 32: A *segment* shall be specified with at least one *data element* in addition to the *qualifier data element*.

Rule 33: The *data element(s)* within a *segment* shall relate directly to the purpose of the *segment.*

Rule 34: A *data element* within a *segment* shall be specified with a *status* of either *mandatory* or *conditional*.

Rule 35: A *data element* may not be specified more than once in a *segment* unless it is a *qualified composite data element*.

Rule 36: The number of times a *qualified composite data element* can be specified in a *segment* shall not exceed the number of *code values* for the *qualifier data element* in the *composite data element*.

Rule 37: *Mandatory data elements* shall be specified at the beginning of the *segment*, followed by *conditional data elements*.

Rule 38: The addition of a *data element* to an existing *segment* shall be at the end of the *segment*.

Rule 39: A *stand-alone data element* shall only be replaced in a *segment* structure by a *composite data element*, if:

- in the *composite data element* structure, the *stand-alone data element* is specified, with its original *status*, as the first *component data element*;
- the *status* of the *composite data element* is equivalent to the *status* of the first *component data element*;
- the other *component data element(s)* have a *status* of *conditional*.

Rule 40: A *data element* shall not be deleted from an existing *segment*.

Rule 41: The *status* of a *data element* in an existing *segment* shall not be changed from *conditional* to *mandatory*.

4.6 Composite data elements

Rule 42: A *composite data element* shall be specified when there is a requirement:

- to refer to an *external code list* (using *simple data elements* 1131 & 3055),
- to combine *simple data elements* into a *simple data element pair,*
- to qualify further a *simple data element* (the qualification of which is not met by the *data element qualifier* that qualifies the *segment),*
- to qualify a *simple data element pair*.

Rule 43: The *composite data element definition* shall:

- describe the meaning of the *composite data element*,
- state what the *composite data element* is, not only what it is not,
- be consistent with its *data element definitions,*
- not embed its *data element definitions* within it,
- be unique within the UN/EDIFACT *Composite Data Element Directory*,
- be stated in singular form,
- expand acronyms on the first occurrence,
- not contain abbreviations,
- not contain a gender bias,
- not contain the phrase ‘self explanatory’, ‘to be defined’, ‘to be provided’, or synonyms thereof.

Rule 44: The name of a *composite data element* shall be unique within the UN/EDIFACT *Composite Data Element Directory.*

Rule 45: The name of *a composite data element* shall be consistent with its definition.

Rule 46: The name of a *composite data element* shall have the same *object class term* and *property term* as the coded and/or non-coded *simple data elements* specified in the *composite data element* structure, with the exception of the *simple data elements* 1131 and 3055.

Rule 47: A *composite data element* shall have one of the following structures:

Structure 1:

Coded Data Element		See Note 1
Non-coded Data Element		See Note 1

Structure 2:

Coded Data Element		
Code list qualifier	(1131)	
Code list responsible agency, coded	(3055)	

Structure 3:

Coded Data Element		See Note 1
Code list qualifier	(1131)	
Code list responsible agency, coded	(3055)	
Non-coded Data Element		See Note 1

Structure 4:

Qualifier Data Element		
Coded Data Element		

Structure 5:

Qualifier Data Element		
Non-coded Data Element		

Structure 6:

Qualifier Data Element		
Coded Data Element		See Note 1
Non-coded Data Element		See Note 1

Structure 7:

Qualifier Data Element		
Coded Data Element		
Code list qualifier	(1131)	
Code list responsible agency, coded	(3055)	

Structure 8:

Qualifier Data Element		
Coded Data Element		See Note 1
Code list qualifier	(1131)	
Code list responsible agency, coded	(3055)	
Non-coded Data Element		See Note 1

Note 1: The data element forms part of a simple data element pair.

Rule 48: The names of the *data elements* of a *simple data element pair* shall have an identical *object class* and *property term*. The coded *simple data element* shall have a *representation term* of 'code' or a *representation term* of 'number'. The non-coded *simple data element* shall have a *representation term* of 'description' or a *representation term* of 'name'.

Rule 49: The *data element definitions* of a *simple data element pair* shall be identical with exception of the introductory phrase. The coded *simple data element* shall begin its definition with 'Code specifying the...'. The non-coded *simple data element*, if it has a *representation term* of 'description', shall begin its definition with 'Free form description of...' and if it has a *representation term* of 'name' shall begin its definition with 'Name of...'.

Rule 50: A *component data element* shall be specified with a *status* of either *mandatory*, or *conditional*.

Rule 51: A *qualifier data element* within a *composite data element* shall be specified with a *status* of *mandatory***.**

4.7 Simple data elements

Rule 52: The *simple data element definition* shall:

- describe the meaning of the *simple data element*,
- state what the *simple data element* is and not only what it is not,
- be unique within the UN/EDIFACT *Data Element Directory*,
- be stated in singular form,
- expand acronyms on the first occurrence,
- not contain abbreviations,
- not contain a gender bias,
- not contain the phrase 'self explanatory', 'to be defined', 'to be provided', or synonyms thereof.

Rule 53: The name of a *simple data element* shall be unique within the UN/EDIFACT *Data Element Directory.*

Rule 54: The name of a *simple data element* shall be consistent with its definition.

Rule 55: A coded *simple data element* which is not defined as a *qualifier data element* shall have a name which ends with the *representation term* of 'code' or with the *representation term* of 'number'.

Rule 56: A *simple data element* shall be specified with a *data value representation*.

Rule 57: A coded *simple data element* that has an associated UN/EDIFACT *Code List* shall be specified with a *data value representation* of at least an..3.

4.8 Identification of external code lists

Rule 58: The identification of an *external code list* shall be achieved by using *simple data elements* 1131 (Code List Qualifier) and 3055 (Code List Responsible Agency, coded) within a *composite data element*.

Rule 59: A coded *stand-alone data element*, or a coded *component data element* not directly followed by *component data elements* 1131 and 3055, shall have at least one code value specified for the *data element* in the UN/EDIFACT *Code List Directory*, except where an UN/ECE approved recommendation or an ISO *code list* has been identified.

4.9 Code values

Rule 60: A *code value* name and *code value definition* shall be within the scope of the *simple data element* name and *simple data element definition* to which it belongs.

Rule 61: The *code value definition* shall:

- be unique within the *code list* for the *simple data element* to which it belongs,
- be stated in the singular,
- state what the code value represents,
- expand acronyms on the first occurrence,
- not contain abbreviations,
- not contain the phrase ‘self explanatory’, ‘to be defined’, ‘to be provided’, or synonyms thereof.

Rule 62: The *code value* name shall be consistent with the *code value definition*.

Rule 63: A *code value* specified in the UN/EDIFACT *Code List Directory* shall conform to a *data value representation* of an..3.

Rule 64: A *code value* for a *qualifier data element* shall be specified only in the UN/EDIFACT *Code List Directory*.

Annex A
(normative)

Definitions

NOTES:

1. **When a word or phrase appears in italics within a definition, this means that a definition exists in this annex for this word or phrase.**

2. **The definitions reference the following UN/EDIFACT documents:**
   R.1023 - TRADE/WP.4/GE.1/R.1023 (Rules for Presentation of Standardized Messages and Directories Documentation)
   R.1241 - TRADE/WP.4/GE.1/R.1241 (EDIFACT Application Level Syntax Rules, Part 1)

A.1 **code list:** The complete set of *data element values* of a coded *simple data element.* [R.1241]

A.2 **code list directory:** A listing of identified and specified *code lists.* [R.1241]

A.3 **code value:** A coded representation of a permissible *data element value*.

A.4 **code value definition:** A statement that describes the meaning of a *code value* and permits its differentiation from all other *code values* in a *code list*. Referred to as 'code value description' in document R.1023.

A.5 **component data element:** A *simple data element* used within a *composite data element.* [R.1241]

A.6 **composite data element:** An identified, named and structured set of functionally related *component data elements*, as defined in a *composite data element specification*. It represents a single unit of data.

A.7 **composite data element definition:** A statement that describes the meaning of a *composite data element* and permits its differentiation from all other *composite data elements*. Referred to as 'composite data element description' in document R.1023.

A.8 **composite data element directory:** A listing of identified, and named *composite data elements* with their *composite data element specification.* [R.1241]

A.9 **composite data element specification:** The description of a *composite data element* in a *composite data element directory*, including the specification of the position and *status* of the *component data elements* constituting the *composite data element.* [R.1241]

A.10 **conditional:** A type of *status*, used in a *message specification*, *segment specification*, or *composite data element specification*, to specify that a *segment group, segment*, *composite data element*, *stand-alone data element* or *component data element* is used optionally or when the appropriate conditions occur. [R.1241]

A.11 **data element:** A unit of *data* described in a *data element specification*. There are two forms of data element, a *simple data element* and a *composite data element.*

A12 **data element definition:** A statement that describes the meaning of a *simple data element* (*simple data element definition*) or a *composite data element* (*composite data element definition*).

A.13 data element directory: A listing of identified, named and specified *simple data elements* (*simple data element directory*) or *composite data elements* (*composite data element directory*). [R.1241]

A.14 data element specification: The specification of a *composite data element* in a *composite data element directory* (*composite data element specification*), or of a *simple data element* in a *simple data element directory* (*simple data element specification*). [R.1241]

A.15 data element value: A specific instance of a *simple data element*, represented as specified in a *simple data element specification* and, if the *simple data element* is coded, in a *code list.* [R.1241]

A.16 data segment clarification section: The sub-section of a *message specification* that defines the use of each *segment* and *segment group* within the *message*.

A.17 data value representation: The types of characters allowed (e.g. alphabetic, numeric) and conditions of length relating to the *data element values* of a *simple data element.* [R.1241]

A.18 external code list: A list of *code values*, not included in a UN/EDIFACT *Code List Directory*, that is maintained and published by a recognised Maintenance Agency.

A.19 level of abstraction: The degree to which broad or common characteristics are promoted over narrow or specific characteristics.

A.20 mandatory: A type of *status*, used in a *message specification*, *segment specification*, or *composite data element specification*, to specify that a *segment group, segment*, *composite data element*, *stand-alone data element* or *component data element* shall be used at least one time. [R.1241]

A.21 message: An identified, named and structured set of functionally related *segments*, covering the requirements for a specific type of transaction (e.g. invoice), as described in a *message specification*; a message starts with a *message header* and ends with a *message trailer*. In *transfer*, a message is a specific ordered set of *segments* in conformance with a *message specification.* [R.1241]

A.22 message definition: The sub-section of a *message specification* that describes the function of the *message* and permits its differentiation from all other *messages*. Referred to as 'functional definition' in document R.1023.

A.23 message directory: A listing of identified and named *messages* each with its *message specification.* [R.1241]

A.24 message header: The *service segment* starting and uniquely identifying a *message.* [R.1241]

A.25 message specification: The description of a *message* in a *message directory*, including the specification of the position, *status* and maximum number of occurrences of the *segments* and *segment groups* constituting the *message.* [R.1241]

A.26 message trailer: The *service segment* ending a *message.* [R.1241]

A.27 message type: Code identifying a type of *message.* [R.1241]

A.28 object class term: The first component of the name of a *simple data element*, a *composite data element* or a *segment* which represents the dominant area of interest.

A.29 property term: A required component of the name of a *simple data element* or an optional component of the name of a *composite data element* or a *segment*. It is subordinate to the *object class term* and represents the distinguishing characteristic or property of the dominant area of interest.

UN/EDIFACT Message Design Rules for Batch EDI

A.30 qualified composite data element: A *composite data element* whose first *component data element* is a *qualifier data element*.

A.31 qualifier data element: A *simple data element* whose *data element value*, when extracted from the associated UN/EDIFACT *code list directory*, gives specific meaning to a *composite data element* or a *segment*.

A.32 qualifier term: A name component retained as a *code value* name in the *code list* for a *qualifier data element*.

A.33 representation term: A required component of the name of a *simple data element*. It is defined in a controlled set of representation terms and describes the form of the set of valid *data element values* for a *data element*.

A.34 sectionalised message: A message specification that has been divided into two or more pre-defined sections.

A.35 segment: An identified, named and structured set of functionally related *composite* and/or *stand-alone data elements*, as defined in a *segment specification*. In the *segment specification*, a segment starts with a *qualifier data element*. There are two types of segment; a *user segment* and a *service segment*.

A.36 segment collision: Segment collision occurs if the sequential processing of the *segment* order within any instance of a *message* will result in ambiguous identification of any *segment*, in terms of its ordinal position in the *message specification*.

A.37 segment definition: A statement that describes the purpose of a *segment* and permits its differentiation from all other *segments*. Referred to as 'segment description' in document R.1023.

A.38 segment directory: A listing of identified and named *segments* with their *segment specification.* [R.1241]

A.39 segment group: An identified hierarchical set of *segments* and/or *segment groups* within a *message.* [R.1241]

A.40 segment specification: The description of a *segment* in a *segment directory*, including the specification of the position, *status* and maximum number of occurrences of the *data elements* constituting the *segment.* [R.1241]

A.41 segment table: The sub-section of a *message specification* that defines in a table the hierarchical structure of a *message*. A segment table identifies, and specifies the position, the *status* and maximum number of occurrences of, the *segments* and *segment groups* within the *message*.

A.42 segment tag: A *simple data element* uniquely identifying a *segment*, by reference to a *segment directory.* [R.1241].

A.43 service segment: A *segment* used i). in service *messages*; ii). to control the transfer of data. A service segment specification contains only service *composite data elements* and/or service *simple data elements.* The *segment tag* of a service segment begins with the letter 'U'.

A.44 simple data element: A *data element* containing a single *data element value*. There are two uses of a *simple data element*: within a *composite data element* (*component data element*); and within a *segment* outside a *composite data element* (*stand-alone data element).* [R.1241]

A.45 simple data element directory: A listing of identified and named *simple data elements* with their simple *data element specification.* [R.1241]

A.46 simple data element pair: A set of two *simple data elements*, one of which provides information in coded form, the other provides the descriptive equivalent of the coded form.

A.47 stand-alone data element: A *simple data element* used within a *segment* without being in a *composite data element.* [R.1241]

A.48 stand-alone segment: A *segment* used within a *message* without being in a *segment group.*

A.49 status: An *attribute* of a *segment*, a *composite data element* or a *simple data element* identifying the rules for the presence or absence of the *segment/data element* in the usage of a *message*. The types of status are *conditional* and *mandatory.* [R.1241]

A.50 syntax service directory: A directory containing service messages, *service segments*, service composite data elements and service data elements maintained as part of ISO 9735 (EDIFACT - Application level syntax rules).

A.51 trigger segment: The *segment* starting a *segment group.* [R.1241]

A.52 user segment: A *segment* specified in the UN/EDIFACT *segment directory.* The *segment tag* of a user segment does not begin with the letter ‘U’.

Annex B
(normative)

Rules for naming data elements and segments

B.1 Introduction

This annex contains the rules for defining structured names for UN/EDIFACT simple data elements, composite data elements and segments. These rules are derived from the guidelines and principles described in document ISO 11179-5, clause 6 (Guidelines for structured naming conventions). In certain instances, these guidelines have been adapted to the UN/EDIFACT environment. In particular, the guidelines have been extended to cover not only the naming of simple data elements but also to cover the naming of composite data elements and segments. The use of qualifier terms has been modified to align this with the use of qualifier data elements in UN/EDIFACT.

The rules are classified as:

Semantic rules	rules that define how the meaning is assigned to the content of a name.
Syntactic rules	rules that define the structure of the name and the order and occurrence of the name components.
Lexical rules	rules that define the word form and vocabulary principles that apply to a name and its components.

One of the fundamental principles specified in IS0 11179, and supported by UN/EDIFACT, is that the definition should be developed first and the name should be extracted from it.

B.2 Semantic rules

The components of a name are: *object class term*, *property term and representation term.*

Rule B1: The *object class term* shall represent the dominant area of interest.

For example, in the names of the *simple data elements*:

<u>Reference</u> version number
<u>Amount</u> function code
<u>Movement</u> type description

the words underlined are *object class terms.*

Rule B2: The *property term* shall represent the distinguishing characteristic or property of the dominant area of interest. The property term shall occur naturally in the definition.

For example, in the names of the *simple data elements*:

Reference <u>version</u> number
Amount <u>function</u> code
Movement <u>type</u> description

the words underlined are property *terms.*

Rule B3: The *representation term* shall describe the form of the set of valid *data element values* for a *simple data element*.

For example, in the names of the *simple data elements*:

Reference version <u>number</u>
Amount function <u>code</u>
Movement type <u>description</u>

the words underlined are *representation terms.*

Rule B4: The *representation term* shall be one of the terms specified in the list of approved *representation terms* (see clause B.5).

Rule B5: A *qualifier term* shall only be specified as a *code value* name in the *code list* for a *qualifier data element* that qualifies the *simple data element*.

Rule B6: The *qualifier terms* shall enable a specific meaning to be assigned to a name of the *simple data element* when it is associated with a *qualifier data element*.

For example, in the *code list* for the *qualifier data element* named ‘Cost total amount qualifier’, the *code value* names*:*

Budget period
Accounting period
Tax period

are *qualifier terms* for the *simple data element* named ‘Cost total amount’, when this *simple data element* is associated with the *qualifier data element* named ‘Cost total amount qualifier’.

B.3 Syntactic rules

Rule B7: a) The name of a *simple data element* which is not a *qualifier data element* shall be in the format of:

Object Class Term	Property Term	Representation Term

b) The name of a *simple data element* which is a *qualifier data element* shall be in the format of:

Object Class Term	**Property Term**	**Representation Term**	**'qualifier'**

c) The name of a *composite data element* shall be in the format of:

Object Class Term	**<u>Property Term</u>**

d) the name of a *segment* shall be in the format of:

Object Class Term	**<u>Property Term</u>**

All terms are mandatory unless underlined. Underlined terms are optional.

Rule B8: One and only one *object class term* shall be present in the name of a *data element* or *segment*.

Rule B9: One and only one *property term* shall be present in the name of a *simple data element*

Rule B10: One and only one *property term* may be present in the name of a *composite data element* or *segment*.

Rule B11: One and only one re*presentation term* shall be present in the name of a *simple data element* name.

Rule B12: The name of a *qualifier data element* shall end with the word 'qualifier' following the *representation term*.

B.4 Lexical rules

Rule B13: A name shall be in singular form.

Rule B14: An *object class term* or a *property term* may be composed of one or more words.

Rule B15: A *representation term* shall be composed of one word only.

Rule B16: The components of a name and the words in a multi-word term, shall be separated by the space character.

Rule B17: Special characters shall not be used in a name.

Rule B18: Abbreviations, acronyms and initialisms shall not be used in a name.

Rule B19: A name shall not contain redundant words.

For example, in the name of the *simple data element* 'Employee last name', the *property term* is 'last name' and the *representation term* is 'name'. When the *property term* and *representation term* are combined, one occurrence of the word 'name' is redundant and is deleted.

B.5 List of representation terms

The following list contains the permissible *representation terms*. Requests for additional *representation terms* shall be processed in accordance with the UN/EDIFACT Data Maintenance Request procedures.

amount A number of monetary units. It is normally associated with a type of currency.

code A character string that represents a member of a set of values.

description A series of sentences representing a person, object, place, event or concept. This can be applied both for definitions, which are generally one or two sentences, and for longer textual passages.

name A word or phrase that constitutes the distinctive designation of a person, object, place, event or concept. What the person, object, place, event or concept is known by or called.

number An arithmetical expression representing a particular value. Note: This may often be used to imply sequence or a member of a series.

percent A rate expressed in hundredths between values that have the same unit of measure.

quantity A number of non-monetary units. It is normally associated with a unit of measurement.

rate A quantity or amount measured with respect to another measured quantity or amount.

Annex C
(informative)

Naming examples for the data element directory

C.1 Introduction

This informative annex lists a selection of simple data elements extracted from the UN/EDIFACT D.97A Data Element Directory and indicates how the naming convention outlined in Annex B could be applied. This annex is for information only and is NOT intended to be applied as Data Maintenance Requests to the directory.

C.2 D.97A Data Element Directory

Tag	Data Element Name	Example name
1000	Document/message name	Document name
1050	Sequence number	Sequence number
1056	Version	Version number
1058	Release	Release number
1153	Reference qualifier	Reference number qualifier
1154	Reference number	Reference number
1220	Number of copies of document required	Copies required quantity
1222	Configuration level	Configuration level number
1245	Status indicator, coded	Status code
1366	Document/message source	Document source description
1476	Control agency	Control agency identification number
1490	Consolidation item number	Item number
3035	Party qualifier	Party type qualifier
3036	Party name	Party name
3039	Party id. identification	Party code
3397	Name status, coded	Name status code
3398	Name component	Name component description
4043	Class of trade, coded	Trade class code
5004	Monetary amount	Monetary amount
5393	Price multiplier qualifier	Price multiplier qualifier
5394	Price multiplier	Price multiplier amount
5402	Rate of exchange	Exchange rate
5419	Rate type qualifier	Rate type qualifier
6173	Size qualifier	Size number qualifier

6174	Size	Size number
6341	Currency market exchange, coded	Currency market exchange code
6343	Currency qualifier	Currency code qualifier
6345	Currency, coded	Currency code
6347	Currency details qualifier	Currency details qualifier
6348	Currency rate base	Currency rate base value
6350	Number of units	Units number
6353	Number of units qualifier	Units number qualifier
6426	Number of stages	Stages count number
6428	Actual stage count	Stages actual count number
6432	Significant digits	Significant digits number
6434	Statistical concept identifier	Statistical concept code
7124	UNDG number	United Nations Dangerous Goods number
7436	Level one id.	Level one number
7500	Type of damage	Damage type description
7501	Type of damage, coded	Damage type code
7502	Damage area	Damage area description
7503	Damage area identification	Damage area code
8211	Permission for transport, coded	Transport permission code
8341	Haulage arrangements, coded	Haulage arrangements code
8351	Hazard code identification	Hazard code
9430	Footnote set identifier	Footnote set code
9432	Footnote identifier	Footnote code
9434	Code name	Code value name

Annex D
(informative)

A model of the message design process

The need to design UN/EDIFACT messages arises when parties identify that their business requirements are based on information flows that need to be exchanged electronically.

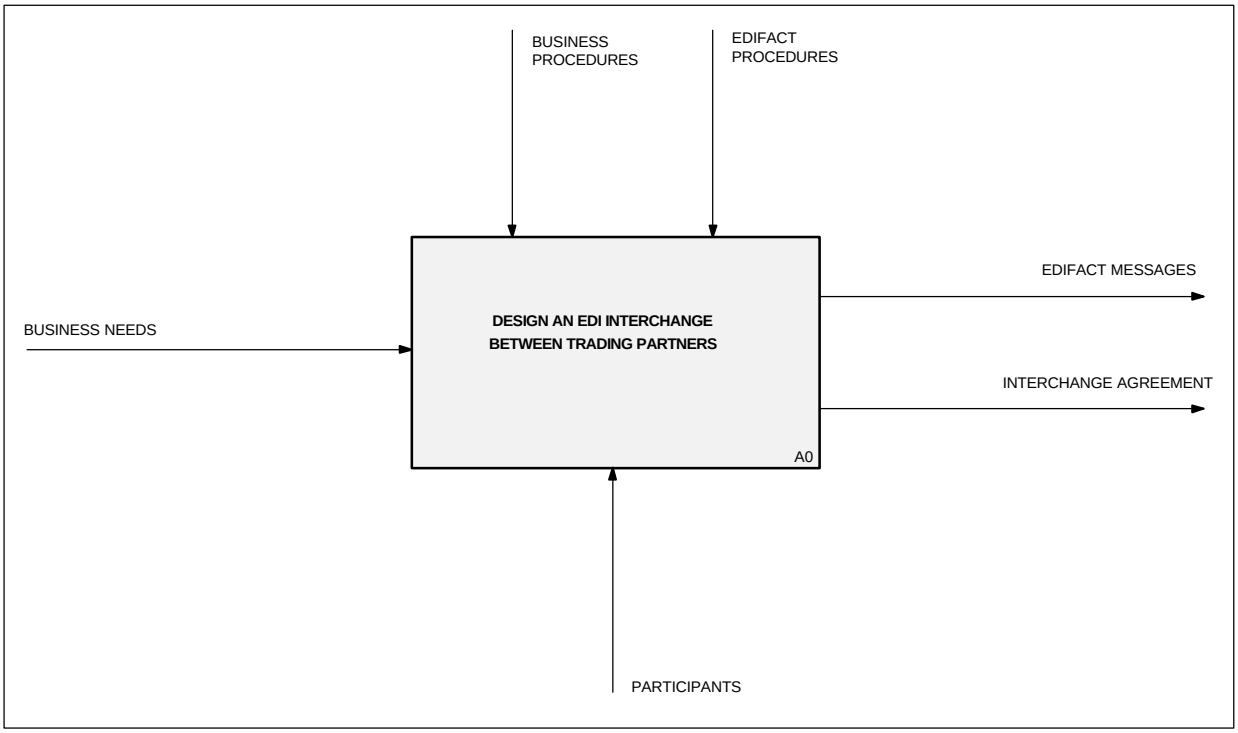

Figure D.1 Design an EDI interchange between trading partners

The analysis and identification of the core business needs serves as the basic input to the design activity. Company, legal and/or business procedures provide additional input to the design process. The UN/EDIFACT procedures ensure that the requisite standards are available, and they support in a standardised manner a wide range of business applications.

Once it has been decided to conduct a business process by an electronic means, the requirements of the individual business functions can be established. This will likely entail generating a model of the key aspects of the business environment under study.

Following agreement on the overall requirements by all the parties involved, the specific system requirements can be defined. These system requirements will specify all the information flows which will pass among the various parties and the functions which they are designed to carry out.

In the business scenario that is being defined, all of the information flows, both in a manual and a electronic form, would be described. All the business constraints between these flows would also be identified. This includes identifying each event which triggers a given information flow.

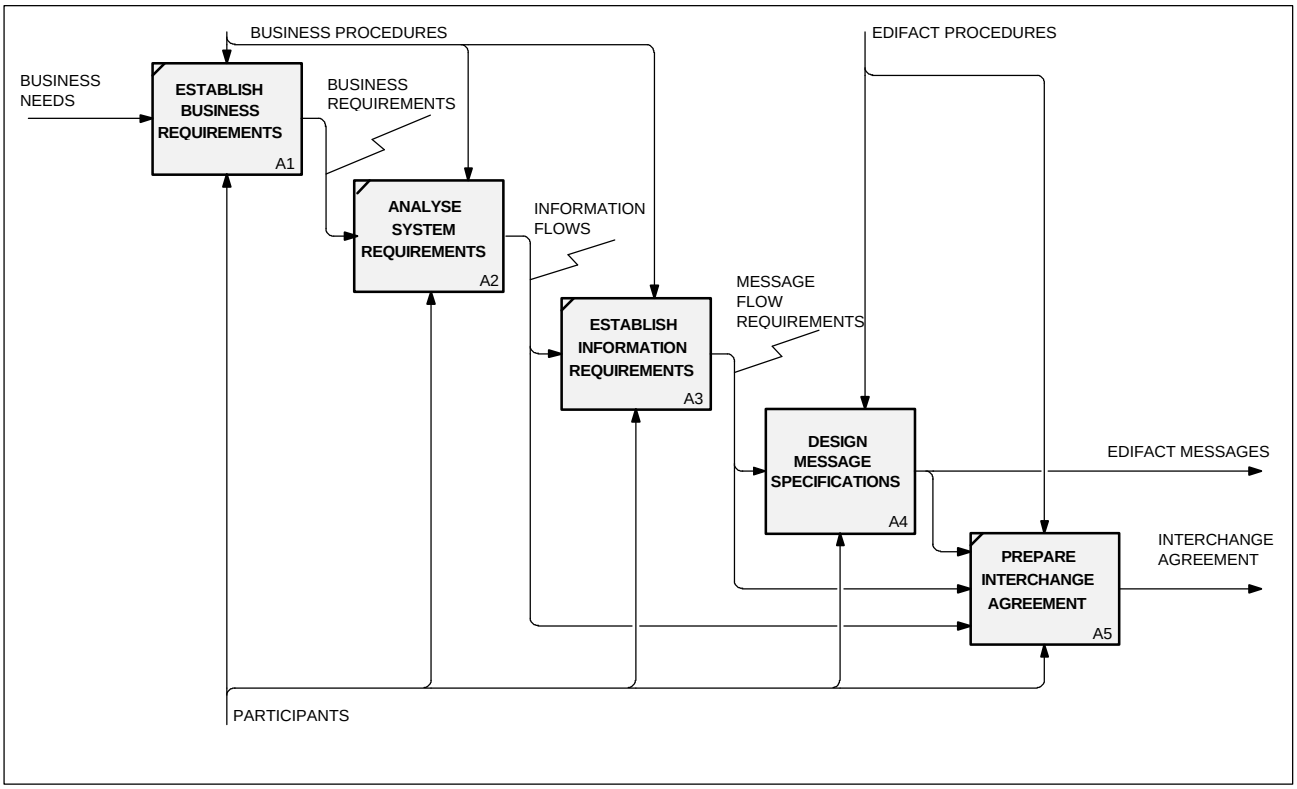

Figure D.2 Design an EDI interchange between trading partners

Once the system requirements have been specified, each of the information flows are examined, in order to establish their adequacy for transmission by electronic means. For those information flows selected as candidates for an electronic means of transfer, these are analysed in detail to determine the precise data requirements of the particular information flows.

The data requirements for each electronic information flow can be structured into an identified and named set of entities, each of which describe a specific subject through their respective attributes. It is from these entity definitions and/or data models that an initial message specification is developed. See also document TRADE/WP.4/GE.1 R.1212 (Draft Business and Information Modeling Framework for UN/EDIFACT).

The current UN/EDIFACT Message Directory should be examined in order to determine whether an existing message corresponds to the required business function.

If an existing message(s) fulfills the function, the message specification is compared to that contained in the UN/EDIFACT Message Directory. Data Maintenance Requests for any additional requirements not covered by the existing message would then be raised and submitted in accordance with the UN/EDIFACT procedures.

If no existing UN/EDIFACT message is found suitable, a Data Maintenance Request for a new message would be prepared and submitted through the UN/EDIFACT process.

When the individual messages have been identified, formal interchange agreements are normally prepared between the parties who wish to participate in the electronic exchange.

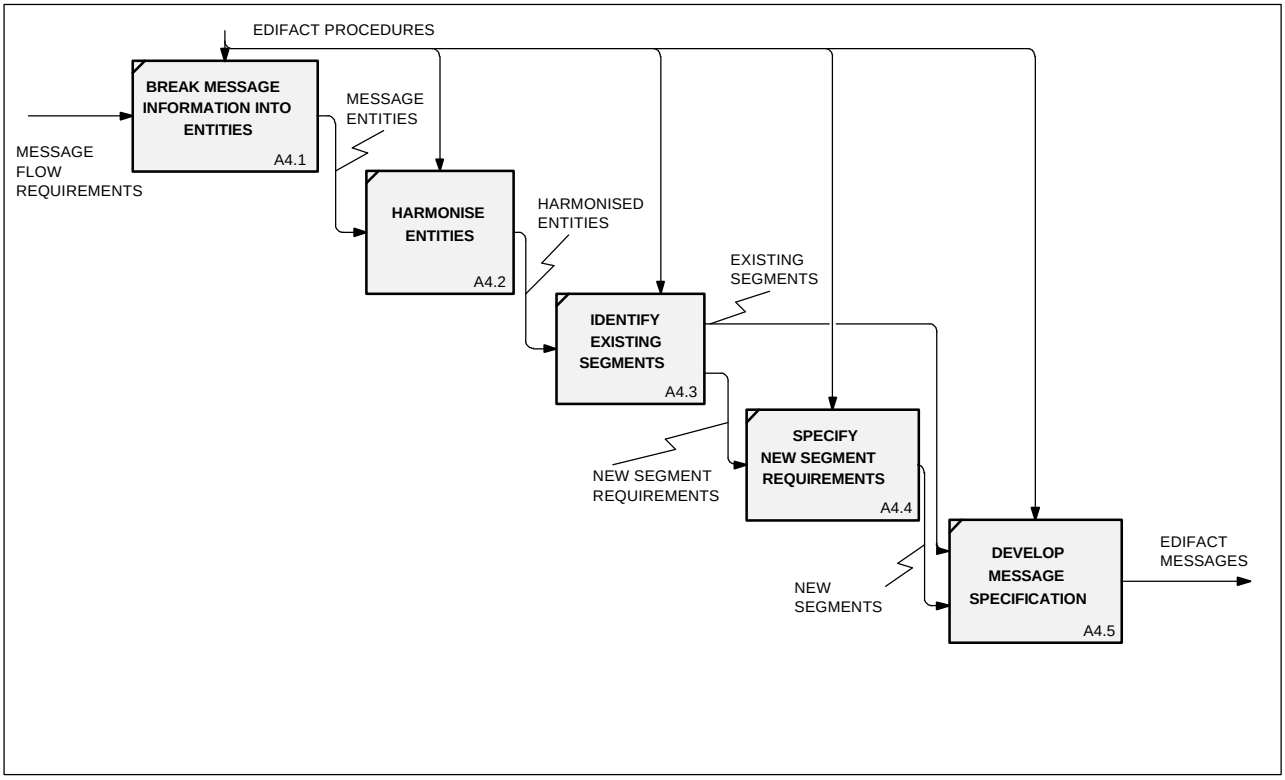

Figure D.3 Design message specifications

The data requirements for an information flow are analysed in order to break them down into entities representing a specific subject. All of the attributes for an entity shall refer to the subject being defined by the entity.

Specific modeling/normalisation techniques should be used to ensure that each attribute refers to a single piece of information for the subject. If an attribute is not totally dependent on the subject being addressed by the entity, a new entity should be created to express the complete subject of the attribute.

Once the entities have been harmonised, the UN/EDIFACT Segment Directory is examined to select existing segments to meet the requirements for each entity identified. The requirements of a single entity may be met by a single segment or a combination of segments specified in a segment group. The requirements for one or more attributes may be met by requesting additional qualifier data element code values for an existing segment or composite data element.

For those remaining attributes not covered by an existing UN/EDIFACT segment, one or more new segments would be requested following the message design rules. Data Maintenance Requests for the new segment(s), new data elements, and new code values would need to be raised and submitted in accordance with the UN/EDIFACT procedures.

The message specification can be developed, respecting the message design rules, once the data requirements for a particular information flow have been assigned to existing and/or new segments. The message structure, including the mandatory or conditional status and the number of occurrences of each segment and/or segment group, are determined from the information flow requirements.