UNITED
NATIONS

E

Economic and Social
Council

Distr.
GENERAL

TRADE/CEFACT/1999/3
6 January 1999

ORIGINAL : ENGLISH

ECONOMIC COMMISSION FOR EUROPE

COMMITTEE FOR TRADE, INDUSTRY AND ENTERPRISE DEVELOPMENT
Centre for the Facilitation of Procedures and Practices
For Administration, Commerce and Transport (CEFACT)
Item 5 of the provisional agenda
Fifth session, 15-17 March 1999

FINAL UN/EDIFACT MESSAGE DESIGN RULES FOR EDI

* * *

Submitted by the EWG Group*

This document has been submitted by the EDIFACT Working Group (EWG) for publication. It replaces previous document TRADE/WP.4/R.840/Rev.4 and supports version 4 of the UN/EDIFACT Syntax. Document TRADE/CEFACT/1999/2 provides the implementation timetable and strategy.

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

GE.99-

Contents

	Page
Foreword	**3**
1. INTRODUCTION	**9**
2. SCOPE	**9**
3. REFERENCES	**9**
4. MESSAGE DESIGN RULES	**9**
4.1 General	**10**
4.2 Messages	**10**
4.3 Sectionalised messages	**11**
4.4 Segment groups	**11**
4.5 Segments	**12**
4.6 Composite data elements	**14**
4.7 Simple data elements	**16**
4.8 External code lists	**17**
4.9 Code values	**17**
Annex A – Definitions	21
Annex B - Rules for naming data elements and segments	26
Annex C - Naming examples for the data element directory	30
Annex D - A model of the message design process	32
Annex E - Notation for Dependency notes	36

Foreword

i) Background

During the fifty-fifth session of the Meeting of Experts (UN/ECE/TRADE/WP.4/GE.1) on Data Elements and Automatic Data Interchange (TRADE/CEFACT/GE.1/1997/1, 7 April 1997), the Chair of GE.1 noted the need to have the requirements for Version 4 of the syntax, including interactive UN/EDIFACT, incorporated into the Message Design Rules. Accordingly, a proposal to establish a new group was made and accepted by the meeting. Following the authority granted to it during the fifty-fifth session of the Meeting of Experts, the ESG approved the terms of reference for the MDR Group for Version 4 of the Syntax, which is contained in Annex C of GE1/ESG/97N0048, dated 30 June 1997 (the report from the UN/EDIFACT Steering Group (ESG) Meeting of 19-20 May 1997). The ESG empowered the Version 4 Message Design Rules Group at its meeting of 22 August 1997.

The terms of reference for this new Message Design Rules (MDR) Group include the following major points:

1. The objective is to maintain and issue the UN/EDIFACT Message Design Rules in accordance with UN/EDIFACT procedures.
2. Incorporate Version 4 of EDIFACT Application Level Syntax Rules (ISO 9735) for both batch and interactive EDI into the Message Design Rules.
3. Deliver the UN/EDIFACT Message Design Rules for CEFACT approval by September 1998, allotting appropriate time for regional review.
4. Seek the views of the UN/EDIFACT constituency on all additions or changes to the Message Design Rules in a consultative manner through the widespread circulation of the group's proposals.
5. Establish a quality assurance process to ensure the delivery of a quality product in accordance with accepted quality review practices.
6. Report on its progress at each JRT to the JRT Steering Group.
7. The Revision to the Message Design Rules shall be subject to CSG approval.
8. Before submitting the Message Design Rules for final approval, the MDR Group shall develop an implementation timetable and strategy in consultation with the Co-Chairs of JTAG.

The Group followed much the same process as the group that prepared the design rules for Version 3 of the syntax. R.1083 set forth the organisational structure and working procedures for the two MDG Group teams: Draft and Edit Team (DET) and Quality Assurance Team (QAT). The time scheme for accomplishment of the current task was contained in the terms of reference.

ii) Message 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 previous MDR Group first analysed the UN/EDIFACT Batch Directory to determine the approach used for the design of existing segments and composite data element structures. From that analysis, the group developed a set of consistent rules that distinguish between when to design a segment and when to design a composite data element. This revision follows that philosophy established for batch. This version also incorporates the rules for interactive EDI messages, dependency notes and the use of repeating techniques.
EDI message designers have long struggled with the difficulty of placing the necessary data for electronic commerce within the framework of a standard EDI syntax in order to achieve a "semantically complete" exchange. Interactive EDI complicates this situation in that messages are more likely to consist of a series of exchanges, e.g., conducting an interactive telephone conversation, rather than sending a message and not necessarily expecting a response. Further, interactive EDI introduces new modelling concepts and techniques which result in structures that, when compared to traditional batch structures, raise questions about duplicate and overlapping functionality. In fact, the design philosophy for both batch and interactive EDI is virtually the same. However, the resultant structures can manifest themselves in a somewhat different representation.
Interactive EDI generally utilises modelling techniques that include conducting a complete business analysis and the development of process and data models. These models are used to resolve semantic differences within the description of the business process and to achieve unambiguous standards definitions for each component of the interactive EDI syntax, i.e., scenario, dialogue, role, message, message structure, information grouping and simple data elements. Because this is in a sense an "art" rather than a "science", one must introduce a sense of balance in reflecting the process model requirements without ambiguity while not introducing explosive growth in the standards.

The process model provides message function or business action and the set of required or optional business information as dictated by the business process. At this level the data requirements specified for messages are typically not at the data element level, but are identified in terms of business purpose. The data model details the logical relationships among data elements in the form of "entities" (objects that information describes), "attributes" (those data elements which describe the entity), and the cardinality relationships among "entities". It is at this point that the perspective of the structural representations for batch and interactive EDI begin to diverge. Interactive EDI is based on the development of modelled data requirements yet must use "non-modelled" structures to represent its requirements. In designing interactive structures emphasis is placed on the representation of data groupings which are communicated within the business process. These data groupings are drawn from one or more entities within the data model, which appear within an interactive EDI segment, sometimes referred to as "semantic units". These "semantic units" then are a collection of data elements required to communicate *a single business thought*.

Thus, the general philosophy of standards design is exactly the same as with batch EDI, however, the representation begins to diverge from the more familiar batch constructs. In turn this leads to a discussion of duplicate or overlapping functionality. What might very well be characterised in batch EDI, as duplicative or overlapping segment functionality (because of the presence of similar lower level structures) is clearly not the case with interactive EDI. Interactive EDI must convey the requisite data grouping which could very well consist of combinations of lower level structures used in multiple segments. However, any one structure itself is by definition very different in terms of the business thought that is being conveyed. It is this critical point of view that must be recognised in order to fully appreciate the distinctions that will exist in appearance of the resultant interactive standards design.

TRADE/WP.4/R.1237, Interactive Message Design Guidelines, recognised the principles for *"concise, efficient and simple"* message and supporting data structure design. Within this approval came the recognition of the following business requirements:

In very high volume interactive environments characterised by large transaction volumes and short response times, the use of qualifiers results in dramatic increases in processing and communications costs.
Interactive segment and composite data elements should be developed to meet the functional definition by including the most efficient use of simple data elements with the minimum use of qualifiers.
The positioning of stand-alone data elements and interactive composite data elements within interactive segments is important to the efficient processing of interactive messages.

Implicit and explicit qualification (See Annex A) is an acceptable methodology for achieving interactive business requirements.

Batch requires explicit qualification within the context of acceptable levels of abstraction as a means of developing reusable design structures. Interactive EDI permits the use of implicit qualification (e.g., qualification by the functional definition of the message or the location of the structure within the message) as the means for developing reusable design structures and for satisfying the requirement for concise, efficient and simple messages. The point of balance between the use of explicit and implicit structures comes in the art of design as well as the requirement to adhere to overall design philosophy.

As this revision contains rules applicable to both batch and interactive EDI, it was necessary to add to the existing rules and to categorise them into sections. It maintains the underlying philosophy, under which Revision 4 was built. However, since there are differences between batch and interactive message design, the key points have been categorised into relevant sections as follows:

a. Philosophy applicable to both batch and interactive message design:

1) The most important aspect of message design is the provision of definitions for the meaning of the data and the extraction of a name directly from these definitions to label the data.
2) A message represents the structure for the data of a defined information flow between two parties.
3) Segments and segment groups are one of the basic building blocks of messages. A segment and/or segment group 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.

4) 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.

5) 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 (see Annex B) 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.

6) 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.

7) 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.

8) 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.

9) The simple data element identifies a single unit of data whose characteristics are entirely defined by the specification of the data element within the UN/EDIFACT directories.

10) 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). However, a maximum length must be specified for each data element.

11) The syntax (ISO9735/1) is precise in its direction for use of dependency notes. Accordingly, the raising of new rules for dependency notes against messages, segment groups and segments directly followed its guidance. An analysis was undertaken of the use dependency notes on composite data element structures. The result of that analysis allowed the rules to be set that provides the correct formal notation to express relationships.

b. Applicable only to batch message design:

1) 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 following the segment tag in the segment and enables the usage context of the segment to be defined during message implementation.

2) 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 mandatory or conditional since, within a segment group, a segment may be qualified by another segment at a higher level, hence rendering further qualification unnecessary.

3) A segment structure consists of a qualifier data element and one or more additional data elements. 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.

4) 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.

5) In accordance with this philosophy, the repetition technique shall only be used in conjunction with qualified composite data elements or composite data elements containing the data elements 1131/3055. In the first instance, the qualifier data element may be used to assign different characteristics to the qualified composite data element. In the second instance, different values within de 1131/3055 provide the ability to reference differently assigned values for the characteristic being described.

6) Data elements represent an elementary unit of information. For pragmatic reasons, UN/EDIFACT provides two means of expressing a data element to define the specific characteristics of a segment; the simple data element and the composite data element.

7) The composite data element identifies 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).

8) 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.

c. Applicable only to interactive message design

There are several characteristics which differentiate interactive EDI from batch EDI. An interactive EDI transaction is an instance of a particular scenario. It consists of one or more dialogues, occurring either concurrently or sequentially between two or more parties. A dialogue consists of an interleaved pair of EDIFACT interchanges; an initiator interchange and a responder interchange. Interactive EDI is characterised by the following:

- A formalised association between the two parties using a dialogue.
- The ability, dynamically, to direct the course of the interactive EDI transaction, depending upon the result of earlier exchanges within the dialogue.
- Short response times.
- All the messages exchanged within one dialogue relate to the same business transaction.
  This results in the following major points of design philosophy for interactive messages:
  1) A transaction is a controlled set of dialogues which can take place between two or more parties.
  2) An interactive business transaction is described by a scenario. Scenarios describe the relationships and information exchanges between the parties to an interactive business transaction,
  3) A scenario describes the states through which the transaction progresses.
  4) The objective of the interactive message design is to meet interactive data interchange requirements with the minimum of complexity and redundancy. The aim is to support a business requirement which can be generically defined within the scope of a scenario.

5) Interactive messages must be concise, efficient and simple in terms of function and design.
6) Generic design should be held as a goal to be achieved for interactive EDI messages.
7) The scope of the term "generic" in an interactive EDI context, is likely to be aligned to an interactive EDI business case. For example, a scenario developed for interactive passenger reservations would provide the scope for messages to meet the business requirements of users of that scenario. It would, therefore, be designed to be generic across sectors in the industries to which that interactive EDI scenario applied, (e.g. rail, air, ferry, car rental, hotel, coach, tour operators, etc.).
8) Interactive EDI message design must allow messages to be developed that provide the flexibility, standardisation and efficiency needed to meet an interactive EDI business need.
9) The positioning of stand-alone data elements and interactive composite data elements within interactive segments is important to the efficient processing of interactive messages.
10) Implicit and explicit qualification (See Annex A) is an acceptable methodology for achieving interactive business requirements.

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 was recommended by the previous MDR Group. This revision does not alter that recommendation.

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.
d) There is recognition that the term "number" has been used inconsistently within many data directories, leading to ambiguity. Sometimes "number" refers to an arithmetic value. In other cases, "number" is used to refer to a value from an identification scheme (e.g., an insurance policy), to distinguish one instance from another. In order to differentiate between these two types of usage, the representation term of "identifier" has been introduced and the definition of "number" has been restricted to an arithmetic value.

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' be 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) Anti-collision techniques

A new technique has been introduced in Version 4 of the Syntax to permit the prevention of segment collision, by use of the UGH/UGT segment group.

A message specification shall ensure unambiguous identification of each message segment upon receipt. Identification shall be possible on the basis of the segment tag and the segment's position in the transferred message. Identification shall not depend on a segment's status or maximum number of occurrences.

The UGH/UGT segment group shall be used in a message specification when it is not otherwise possible to ensure unambiguous identification of each message segment upon receipt through the basis of the segment tag and the segment's position in the transferred message.

In this event, the UGH/UGT segment group shall be specified to surround the segment group which otherwise could not be unambiguously identified.

v) Dependency Notes

Dependency notes are a new concept that was added in Version 4 of the Syntax. If required, dependency notes can be used in the message, segment specification, or composite data element to express relationships.
In a dependency note, a list is defined as two or more entities (where an entity can be a segment group; a segment; a composite data element; a stand-alone data element or a component data element).
Any entity may be subject to more than one dependency note.

vi) Repetition Techniques

The specification of multiple occurrences of a message within a group or within an interchange; a group within an interchange; and a segment group and/or a segment within a message, which existed in the previous version of ISO 9735, has been extended in Version 4 of the syntax. The additional capability for the specification of multiple occurrences of a stand-alone data element and/or of a composite data element within a segment has been introduced. However, in batch EDI the capability of using repeating data elements has been restricted in line with its design philosophy.

vii) Summary

The major differences between this revision and revision 4 of the message design rules are:

- Integration of message design rules for interactive EDI,
- new rules for the anti-collision techniques,
- the use of dependency notes, and
- repetition techniques.

The group has been careful in raising additional message design rules in order to retain the fundamental philosophy upon which revision 4 was built, and accordingly, the rules resulting from the implementation of that philosophy.

For purposes of this revision, the new syntax includes the concept of interactive EDI, extends the coverage of character repertoire, introduces the techniques of dependency notes and repeating data element, and enhances the batch interchange, group, and message header segments, and introduces new segments for anti-collision. New rules have been established for interactive EDI, dependency notes, repeating data elements, and, anti-collision techniques. Additionally, the enhanced functionality of the Batch Message Header Segment (UNH) meant that the previous rule for mandating the Beginning of Message Segment (BGM) was deemed to no longer be valid. Hence the rule was changed accordingly. For the others, the analysis resulted in the conclusion that they were not applicable to message design, and therefore no rules were required.

The group considered the subjects of Security and Non-EDIFACT Objects, and as per the previous MDR group, decided that they were issues of implementation and not message design.

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 EDI

1. Introduction

The UN/EDIFACT (United Nations, Electronic Data Interchange for Administration, Commerce and Transport) Message Design Rules (MDR) Group has prepared the UN/EDIFACT Message Design Rules for EDI. 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 modelling, directory production and message implementation.

This revision retains the paragraphing structure introduced in revision 4. However, each of the original paragraphs has been further subdivided in order to accommodate the rules governing both batch and interactive EDI in this document. The paragraphing follows the hierarchical structure of a message. Where a rule contains a term that appears in Italics, the term is defined in Annex A (Definitions).

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 comply with the UN/EDIFACT Application Level Syntax Rules, versions 1, 2, 3 and 4. The exception to this is where Version 4 introduces the concept of dependency notes and repeating data elements, which shall apply only to Version 4 of the Syntax.

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
ISO 9735	Electronic data interchange for administration, commerce and transport (EDIFACT) - Application level syntax rules (Versions 1, 2, 3, and 4 (parts 1, 2 and 3)).
ISO 11179-5	Information technology - Specification and standardisation of data elements, Part 5 - Naming and identification principles for data elements.

4. Message Design Rules

In this initial version of Revision 5, the Source Rule Number (either from R.840/Rev4 or from R.1237) has been included at the end of the rule, e.g. “[R.840-Rule 13]”.

Where that rule has been amended, e.g. to incorporate interactive, an asterisk has been appended to the rule number, e.g. “[R.840-Rule 3*]”.

If it is a new rule, e.g. for dependency notes, no rule number has been quoted.

4.1 General

4.1.1 Rules common to batch and interactive EDI

Rule 1: *Message*, *segment* and *data element* design shall not include UN/EDIFACT directory items marked for deletion. [R.840-Rule 1]

Rule 2: *Segment* and *data element* design shall not include *syntax service directory* items. [R.840-Rule 2]

Rule 3: *Message* design shall not include *syntax service directory* items, with the exception of the *Message Header* (UNH/UIH), Section Control (UNS), Anti-collision segment group Header (UGH), Anti-collision segment group trailer (UGT) and *Message Trailer* (UNT/UIT) *service segments*. [R.840-Rule 3*]

Rule 4: A *dependency note* shall be specified with a valid *dependency identifier* as per Normative Annex E.

4.2 Messages

4.2.1 Rules common to batch and interactive EDI

Rule 5: 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* within the *target UN/EDIFACT directory.* [R.840-Rule 4*]

Rule 6: A *message* shall be identified by a *message type* code which shall be six alphabetic uppercase characters from the ISO 646 character set and shall be unique within both the UN/EDIFACT batch and interactive directories. [R.840-Rule 5*]

Rule 7: The name of a *message* shall be unique within the *target UN/EDIFACT directory*. [R.840-Rule 6*]

Rule 8: The name of a *message* shall be consistent with its *message definition*. [R.840-Rule 7]

Rule 9: A *message* shall start with the *Message Header* (UNH/UIH) *service segment* and shall end with the *Message Trailer* (UNT/UIT) *service segment*. [R.840-Rule 8*]

Rule 10: A *message* shall be structured without *segment collision*. [R.840-Rule 10]

Rule 11: The UGH/UGT *segment group* shall be used in a *message specification* in order to avoid *segment collisions* which otherwise could not be avoided. In this event, the UGH/UGT *segment group* shall be specified to surround the *segment group* which could not be unambiguously identified.

Rule 12: Wherever used, the UGH/UGT *segment group* shall have a *maximum number of occurrences* of one, and shall be specified with a *status* of *conditional* or *mandatory*, identical to the *status* of the *segment group* it surrounds.

Rule 13: In the UGH/UGT *segment group*, the UGH *segment* shall be the *trigger segment.* The UGT shall be the last *segment* in the *segment group*, shall be *mandatory* and shall be specified with a *maximum number of occurrences* of one.

Rule 14: The purpose of each *segment* and each *segment group* within a *message* shall be specified in the *data segment clarification section.* [R.840-Rule 12]

Rule 15: 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 Directories.* [R.840-Rule 13]

Rule 16: 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*. [R.840-Rule 14]

Rule 17: The *segment table* structure shall be consistent with the *message definition* and with the *data segment clarification section*. [R.840-Rule 15*]

Rule 18: A *segment* or *segment group* specified in the *segment table* shall have a *status* of either *mandatory*, or *conditional*. [R.840-Rule 16]

Rule 19: A *segment* or *segment group* specified in the *segment table* shall be assigned a *maximum number of occurrences*. [R.840-Rule 17]

Rule 20: A *dependency note* shall not be specified against *segments* or *segment groups* with a *status* of *mandatory.*

Rule 21: *Dependency notes* specified for the same *segment* or *segment group* shall not be in conflict.

Rule 22: A *dependency note* shall list the *position identifiers* of at least two distinct *segment groups,* or one *segment group* and one *segment*, or two *segments*.

Rule 23: A *dependency note* list shall only specify the *position identifiers* of *segments* and/or *segment groups* at the same hierarchical level and within the same parent structure (*message* or *segment group*).

4.2.2 Rules applicable to batch EDI

Rule 24: 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.* [R.840-Rule 9]

Rule 25: 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. [R.840-Rule 11]

4.2.3 Rules applicable to interactive EDI

Rule 26: A *message* shall be specified with at least one *segment* between the *Message Header* (UIH) *service segment* and the *Message Trailer* (UIT) service segment, that is not the Section Control (UNS) *service segment*. [R.1237-Rule 50*]

4.3 Sectionalised messages

4.3.1 Rules common to batch and interactive EDI

Rule 27: 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. [R.840-Rule 18]

Rule 28: 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
- [R.840-Rule 19]

4.4 Segment groups

4.4.1 Rules common to batch and interactive EDI

Rule 29: A *segment group* shall start with a *trigger segment*. [R.840-Rule 20]

Rule 30: A *trigger segment* shall have a *status* of *mandatory.* [R.840-Rule 21]

Rule 31: A *trigger segment* shall have a *maximum number of occurrences* of one. [R.840-Rule 22]

Rule 32: A *segment group* shall contain at least one other *segment* or *segment group* in addition to the *trigger segment*. [R.840-Rule 23]

4.5 Segments

4.5.1 Rules common to batch and interactive EDI

Rule 33: A *segment* shall be identified by a *segment tag* which shall be three alphabetic uppercase characters from the ISO 646 character set. A *user segment* shall not begin with the letter 'U'. [R.840-Rule 24]

Rule 34: The *segment tag* shall be unique within both the UN/EDIFACT batch and interactive *Segment Directories.* [R.840-Rule 25*]

Rule 35: 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 *target* 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, unless the phrase is an intrinsic part of the *segment definition*.
- [R.840-Rule 26*]

Rule 36: The name of a *segment* shall be unique within the *target UN/EDIFACT Segment Directory*. [R.840-Rule 27*]

Rule 37: The name of a *segment* shall be consistent with its *segment definition*. [R.840-Rule 28]

Rule 38: The purpose of a *segment* shall not duplicate, and shall not be represented by, the purpose of another *segment* within the *target UN/EDIFACT directory*, at the same *level of abstraction*. [R.840-Rule 29]

Rule 39: The *data element(s)* within a *segment* shall relate directly to the purpose of the *segment.* [R.840-Rule 33]

Rule 40: A *data element* within a *segment* shall be specified with a *status* of either *mandatory,* or *conditional*. [R.840-Rule 34]

Rule 41: For all *data elements* following the *qualifier data element* in a *segment*, the *mandatory data elements* shall be specified first, followed by the *conditional data elements*. [R.840-Rule 37]

Rule 42: The addition of a *data element* to an existing *segment* shall have a *status* of *conditional* and shall be specified at the end of the *segment*. [R.840-Rule 38]

Rule 43: 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*, or a *conditional 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*.
- [R.840-Rule 39*]

Rule 44: The deletion of a *data element* from an existing *segment* shall result in that *segment* being marked for deletion in the *target UN/EDIFACT Segment Directory*. [R.840-Rule 40*]

Rule 45: The change of *status* from *conditional* to *mandatory* of a *data element* in an existing *segment* shall result in that *segment* being marked for deletion in the *target UN/EDIFACT Segment Directory*. [R.840-Rule 41*]

Rule 46: A *data element within* a *segment* shall be specified with a *maximum number of occurrences*. [R.1237-Rule 37*]

Rule 47: A *dependency note* shall not be specified against *data elements* with a *status* of *mandatory.*

Rule 48: *Dependency notes* specified for the same *data element* shall not be in conflict.

Rule 49: A *dependency note* shall list the *position identifiers* of at least two distinct *data elements*.

Rule 50: A *dependency note* shall list only *position identifiers* contained in the *segment* to which the *dependency note* applies.

Rule 51: A *dependency note* shall only be used to express relationships within a *segment* between:

- *stand-alone data elements*, or
- *composite data elements*, or
- *stand-alone data elements* and *composite data elements*
-

4.5.2 Rules applicable to batch EDI

Rule 52: A *segment* shall not contain the entire contents of another *segment*. [R.840-Rule 30]

Rule 53: A *segment* shall be specified with a *qualifier data element* as the first *data element* following the *segment tag* within a *segment.* [R.840-Rule 31]

Rule 54: A *segment* shall be specified with at least one *data element* in addition to the *qualifier data element* that qualifies the *segment*. [R.840-Rule 32]

Rule 55: The same *data element* shall not be specified at more than one position in a *segment.* [R.840-Rule 35*]

Rule 56: Only *qualified composite data elements* or *composite data elements* containing the *data elements* 1131/3055 can be *repeating data elements*. [R.840-Rule 35*]

Rule 57: A qualified *composite data element* shall be specified with a *maximum number of occurrences* less than or equal to, the number of code values specified for the *qualifier data element* of the *composite data element*. . [R.840-Rule 36*]

4.5.3 Rules applicable to interactive EDI

Rule 58: A *segment* shall not contain only the complete set of *data elements* of another *segment* in the interactive *Segment Directory*. [R.1237-Rule 30*]

Rule 59: A *segment* shall be specified with at least one *data element* that is not a *qualifier data element*.

Rule 60: When a segment is not a qualified segment, the *mandatory data elements* shall be specified first, followed by the *conditional data elements*. [R.1237-Rule 36*]

Rule 61: In a *segment* which requires qualification, the *qualifier data element* shall be specified as the first *data element* in the *segment*.

Rule 62: A segment qualifier shall not be a *repeating data element*.

4.6 Composite data elements

4.6.1 Rules common to batch and interactive EDI

Rule 63: 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 *target* 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, unless the phrase is an intrinsic part of the composite data element definition.
- [R.840-Rule 43*]

Rule 64: The name of a *composite data element* shall be unique within the *target* UN/EDIFACT *Composite Data Element Directory.* [R.840-Rule 44*]

Rule 65: The name of *a composite data element* shall be consistent with its definition. [R.840-Rule 45]

Rule 66: 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 'identifier'. The non-coded *simple data element* shall have a *representation term* of 'description' or a *representation term* of 'name'. [R.840-Rule 48*]

Rule 67: 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...'. [R.840-Rule 49]

Rule 68: A *component data element* shall be specified with a *status* of either *mandatory*, or *conditional*. [R.840-Rule 50]

Rule 69: A *component data element* shall not be a *repeating data element*.

Rule 70: The change of *status* from *conditional* to *mandatory* of a *component data element* in an existing *composite data element* shall result in that *composite data element* being marked for deletion in the *target UN/EDIFACT Composite Directory*.

Rule 71: The *status* of the coded *simple data element* shall be *mandatory* in the *composite data element structure* that consists solely of a *coded simple data element* followed by *simple data elements* 1131 and 3055 (i.e. as per Structure 2 in Rule 81).

Rule 72: The addition of a *simple data element* to an existing *composite data element* shall have a *status* of *conditional* and shall be specified at the end of the *composite data element*. [R.1237-Rule 26*]

Rule 73: The deletion of a *component data element* from an existing *composite data element* shall result in that *composite data element* being marked for deletion in the *target* UN/EDIFACT *Composite Directory*.

Rule 74: A *dependency note* shall only be specified against *component data elements* with a *status* of *conditional.*

Rule 75: *Dependency notes* specified for the same *component data element* shall not be in conflict.

Rule 76: A *dependency note* shall list the *position identifiers* of at least two distinct *component data elements*.

Rule 77: A *dependency note* shall list only *position identifiers* contained in *composite data element* to which the *dependency note* applies.

4.6.2 Rules applicable to batch EDI

Rule 78: 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*.
- [R.840-Rule 42]

Rule 79: 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. [R.840-Rule 46]

Rule 80: A *qualifier data element* within a *composite data element* shall be specified with a *status* of *mandatory***.** [R.840-Rule 51]

Rule 81: 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
1131
3055

Structure 3:

Coded Data Element	See Note 1
1131	
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
1131

3055

Structure 8:

Qualifier Data Element	
Coded Data Element	See Note 1
1131	
3055	
Non-coded Data Element	See Note 1

Note 1: The data element forms part of a *simple data element pair*.
[R.840-Rule 47]

4.6.3 Rules applicable to interactive EDI

Rule 82:

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*.
- to group data representing a *single business purpose*.
-

Rule 83: The name of a *composite data element* shall be consistent with its *composite data element definition.*

Rule 84: A *composite data element* shall not contain only the complete set of *component data elements* of another *composite data element* in the interactive *Composite Directory*. [R.1237-Rule 17*]

Rule 85: In a *composite data element* that requires qualification, the *qualifier data element* shall be specified as the first *component data element* in the *composite data element*. [R.1237-Rule 24*]

Rule 86: In a qualified *composite data element*, the mandatory *simple data elements* shall be specified immediately following the composite *qualifier data element*. In an unqualified *composite data element*, the mandatory *simple data elements* shall be specified first. [R.1237-Rule 23*]

Rule 87: The qualifier of a *component data element* shall follow the *component data element* it qualifies. [R.1237-Rule 25*]

4.7 Simple data elements

4.7.1 Rules common to batch and interactive EDI

Rule 88: 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, unless the phrase is an intrinsic part of the data element definition.
- [R.840-Rule 52]

Rule 89: The name of a *simple data element* shall be unique within the UN/EDIFACT *Data Element Directory.* [R.840-Rule 53]

Rule 90: The name of a *simple data element* shall be consistent with its definition. [R.840-Rule 54]

Rule 91: 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 'identifier'. [R.840-Rule 55*]

Rule 92: A *simple data element* shall be specified with a *data value representation*. [R.840-Rule 56]

Rule 93: 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. [R.840-Rule 57]

4.8 External code lists

4.8.1 Rules common to batch and interactive EDI

There are no rules applicable to this section.

4.8.2 Rules applicable to batch EDI

Rule 94: The identification of an *external code list* shall be achieved by using *simple data elements* 1131 and 3055 within a *composite data element*. [R.840-Rule 58]

Rule 95: 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. [R.840-Rule 59]

4.8.3 Rules applicable to interactive EDI

There are no rules applicable to this section.

4.9 Code values

4.9.1 Rules common to batch and interactive EDI

Rule 96: 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. [R.840-Rule 60]

Rule 97: 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, unless the phrase is an intrinsic part of the code value definition.
- [R.840-Rule 61]

Rule 98: The *code value* name shall be consistent with the *code value definition*. [R.840-Rule 62]

Rule 99: A *code value* for a *qualifier data element* shall be specified only in the UN/EDIFACT *Code List Directory*. [R.840-Rule 64]

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 Standardised Messages and Directories Documentation)**
   ISO9735: **EDIFACT Application Level Syntax Rules, Part 1**

A.1 **batch EDI:** Electronic Data Interchange in which no strong requirements exist for formalised data exchange with an interactive immediacy in query and response between parties. [1]

A.2 **characteristic:** A feature or quality of an object. [2]

A.3 **code list:** The complete set of *data element values* of a coded *simple data element*. (ISO 9735) [3]

A.4 **code list directory:** A listing of identified and specified *code lists.* (ISO 9735) [4]

A.5 **code value:** A coded representation of a permissible *data element value*. [5]

A.6 **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) [6]

A.7 **component data element:** A *simple data element* used within a *composite data element.* (ISO 9735) [7]

A.8 **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. Additionally, in an *I-EDI transaction*, it may represent a *single business purpose*. [8]

A.9 **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) [9]

A.10 **composite data element directory:** A listing of identified and named *composite data elements* with their *composite data element specification.* (ISO 9735) [10]

A.11 **composite data element specification:** The description of a *composite data element* in a *composite data element directory*, including the specification of the position, *status* and any *dependency note*s of the *component data elements* constituting the *composite data element.* [11]

A.12 **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. (ISO 9735) [12]

A.13 **data element:** A unit of data described in a *data element specification*. There are two classes of data element, a *simple data element* and a *composite data element.* (ISO 9735) [13]

A.14 **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*). [14]

A.15 **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*). (ISO 9735) [15]

A.16 **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*). (ISO 9735) [16]

A.17 **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.* (ISO 9735) [17]

A.18 **data segment clarification section:** The sub-section of a *message specification* that defines the use of each *segment* and *segment group* within the *message*. [18]

A.19 **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.* (ISO 9735) [19]

A.20 **dependency identifier:** An *identifier* used in a *dependency note* to specify the type of dependency between the related data structures listed in the *dependency note.* (ISO 9735) [20]

A.21 **dependency note:** A note used: i. in a *message specification* to express relationships between *segments,* between *segments groups,* or between *segments* and *segment groups*; ii. in a *segment specification* to express relationships between *data elements*; iii. in a *composite data element specification* to express relationships between *component data elements*. [21]

A.22 **dialogue:** A two-way conversation between an initiator and responder within an *interactive EDI transaction*. It is formally composed of a pair of interchanges. (ISO 9735) [22]

A.23 **explicit qualification**: Providing specific meaning to a *simple data element* through use in conjunction with a *qualifier data element*. [23]

A.24 **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. [24]

A.25 **identifier:** A character or group of characters used to identify or name an item of data and possibly to indicate certain properties of that data (ISO 9735) [25].

A.26 **I-EDI (Interactive EDI):** The exchange of pre-defined and structured data within a *dialogue*, which conforms to the syntax of Parts 1 and 3 of ISO9735 for some business purpose, between a pair of co-operating processes, in a timely manner. (ISO 9735) [26]

A.27 **I-EDI transaction:** An instance of a *scenario*. It consists of one or more *dialogues*. (ISO 9735) [27]

A.28 **implicit qualification**: Providing specific meaning to a *simple data element* by means of the functional definition(s) of an interactive message type, interactive *segment* type, and/or *composite data element* type to which it belongs; or by the position of the *simple data element* within an interactive *segment* or interactive *composite data element*. [28]

A.29 **level of abstraction:** The degree to which broad or common characteristics are promoted over narrow or specific characteristics. [29]

A.30 **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. (ISO 9735) [30]

A.31 **maximum number of occurrences:** The highest possible number of times something can come into being at a specified location within a *message* or its component parts. [31]

A.32 **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.* (ISO 9735) [32]

A.33 **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) [33]

A.34 **message directory:** A listing of identified and named *messages* each with its *message specification.* (ISO 9735) [34]

A.35 **message header:** The *service segment* starting and uniquely identifying a *message.* (ISO 9735) [35]

A.36 **message specification:** The description of a *message* in a *message directory*, including the specification of the position, *status, maximum number of occurrences*, and any *dependency notes* of the *segments* and *segment groups* constituting the *message.* [36]

A.37 **message trailer:** The *service segment* ending a *message.* (ISO 9735) [37]

A.38 **message type:** Code identifying a type of *message.* (ISO 9735) [38]

A.39 **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. [39]

A.40 **position identifier:** An *identifier* used in a *dependency note* to identify an entity (*segment group*, *segment*, or *data element*) by its position in the parent entity. (ISO 9735) [40]

A.41 **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. [41]

A.42 **qualified composite data element:** A *composite data element* whose first *component data element* is a *qualifier data element*. [42]

A.43 **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*. [43]

A.44 **qualified segment:** A *segment* whose first *data element* is a *qualifier data element*. [44]

A.45 **qualifier term:** A name component retained as a *code value* name in the *code list* for a *qualifier data element*. [45]

A.46 **repeating data element:** A *composite data element* or *stand-alone data element* having a maximum occurrence of greater than one in the *segment specification*. (ISO 9735) [46]

A.47 **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*. [47]

A.48 **scenario:** A formal specification of a class of business activities having the same business goal. (ISO 9735) [48]

A.49 **sectionalised message:** A message specification that has been divided into two or more pre-defined sections. [49]

A.50 **segment:** An identified, named and structured set of functionally related *composite data elements* and/or *stand-alone data elements*, as described in a *segment specification*. There are two types of segment; a *user segment* and a *service segment*. [50]

A.51 **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*. [51]

A.52 **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) [52]

A.53 **segment directory:** A listing of identified and named *segments* with their *segment specification.* (ISO 9735) [53]

A.54 **segment group:** An identified hierarchical set of *segments* and/or *segment groups* within a *message.* (ISO 9735) [54]

A.55 **segment specification:** The description of a *segment* in a *segment directory*, including the specification of the position, *status, maximum number of occurrences* and any *dependency notes* of the *data elements* constituting the *segment.* [55]

A.56 **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, maximum number of occurrences* of, and any *dependency notes* of, the *segments* and *segment groups* within the *message*. [56]

A.57 **segment tag:** A *simple data element* uniquely identifying a *segment*, by reference to a *segment directory.* (ISO 9735) [57]

A.58 **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'. [58]

A.59 **single business purpose:** Data representing a contiguous whole of all or a portion of a business process. [59]

A.60 **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).* (ISO 9735) [60]

A.61 **simple data element definition**: A statement that describes the meaning of a *simple data element* and permits the differentiation from all other *simple data elements*. Referred to as 'data element description' in document. (R.1023). [61]

A.62 **simple data element directory:** A listing of identified and named *simple data elements* with their simple *data element specification.* (ISO 9735) [62]

A.63 **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. [63]

A.64 **simple data element specification:** The set of attributes characterising a *simple data element* in a *simple data element directory.* (ISO 9735) [64]

A.65 **stand-alone data element:** A *simple data element* used within a *segment* without being in a *composite data element.* (ISO 9735) [65]

A.66 **stand-alone segment:** A *segment* used within a *message* without being in a *segment group.* [66]

A.67 **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.* (ISO 9735) [67]

A.68 **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). [68]

A.69 **target UN/EDIFACT directory:** The next maintainable UN/EDIFACT directory (batch or interactive) for which the proposed item is intended. [69]

A.70 **trigger segment:** The *segment* starting a *segment group.* (ISO 9735) [70]

A.71 **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'. [71]

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*..

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 initials 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.
identifier	A character string used to identify and distinguish uniquely, one instance of a value within an identification scheme.
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 identifier
1058	Release	Release identifier
1153	Reference qualifier	Reference identifier qualifier
1154	Reference number	Reference identifier
1222	Configuration level	Configuration level number
1245	Status indicator, coded	Status code
1366	Document/message source	Document source description
1475	Message type identifier	Message type code
1476	Control agency	Control agency identifier
1490	Consolidation item number	Item identifier
3035	Party qualifier	Party function code 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
5402	Rate of exchange	Exchange rate
6174	Size	Size number
6341	Currency market exchange, coded	Currency market exchange code
6345	Currency, coded	Currency code
6350	Number of units	Units number
6353	Number of units qualifier	Unit type code qualifier
6426	Number of stages	Stages count number
6428	Actual stage count	Stages actual count number

Tag	Data Element Name	Example name
6432	Significant digits	Significant digits number
6434	Statistical concept identifier	Statistical concept code
7124	UNDG number	United Nations Dangerous Goods identifier
7436	Level one id.	Level one identifier
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.

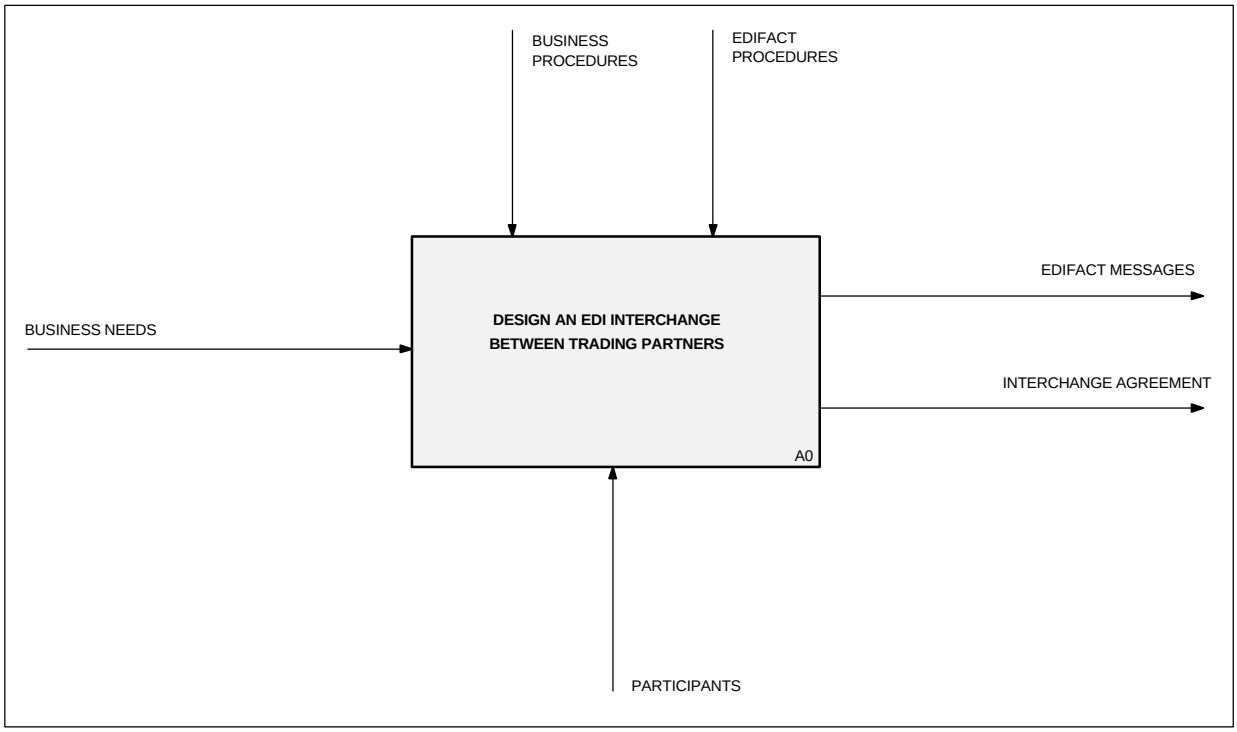

Figure D.1 Design an EDI interchange between trading partners

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.

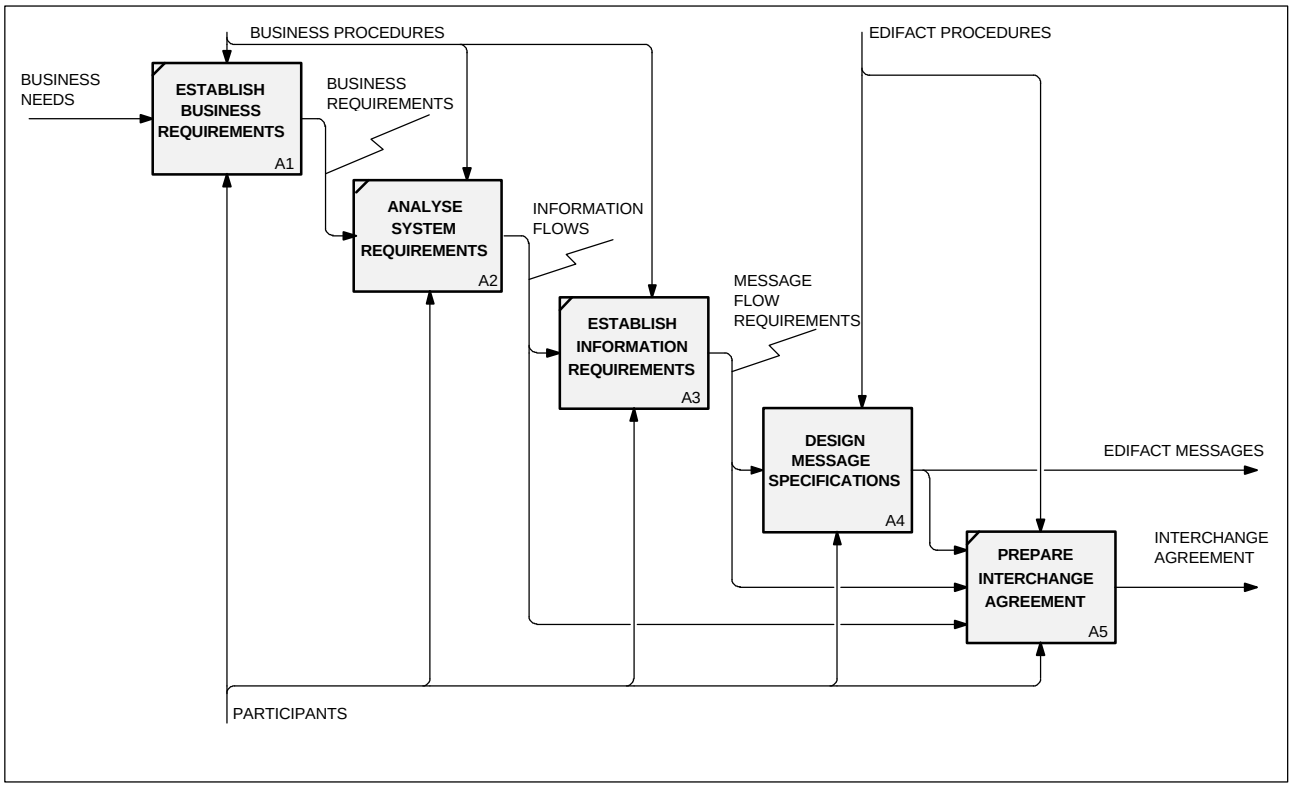

Figure D.2 Design an EDI interchange between trading partners

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 Modelling 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) fulfils 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.

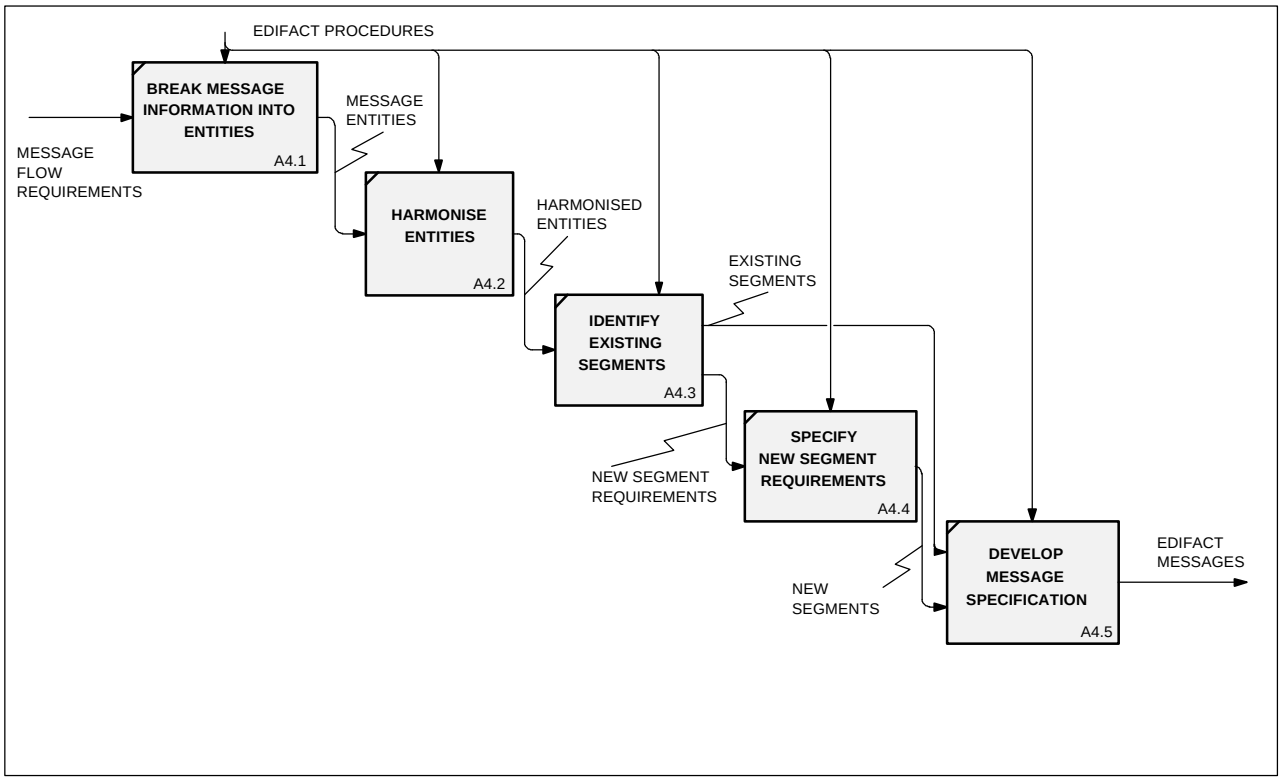

Figure D.3 Design message specifications

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 modelling/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.

Annex E
(Normative)

Notation for Dependency Notes

The notation for dependency notes comprise a dependency identifier followed by a list, in parenthesis, of position identifiers separated by commas e.g. D3(030, 060, 090). The position identifiers point to the related data structures such as segments, segment groups and data elements.

Dependency identifiers are described below.

D1 ONE AND ONLY ONE
One and only one of the position identifiers in the list shall be present.

D2 ALL OR NONE
If one position identifier in the list is present, the rest shall be present.

D3 ONE OR MORE
At least one of the position identifiers in the list shall be present.

D4 ONE OR NONE
No more than one position identifier in the list shall be present.

D5 IF FIRST, THEN ALL
If the first position identifier in the list is present, then all of the others shall be present. It is permissible that one or more of the position identifiers not specified as the first position identifier in the list may be present, without requiring the first position identifier to be present.

D6 IF FIRST, THEN AT LEAST ONE MORE
If the first position identifier in the list is present, then at least one more shall be present. It is permissible that one or more of the position identifiers not specified as the first position identifier in the list may be present, without requiring the first position identifier to be present.

D7 IF FIRST, THEN NONE OF THE OTHERS
If the first position identifier in the list is present, then none of the others shall be present.