

---

---

**Joint ISO/TC 154 – UN/CEFACT  
Syntax Working Group (JSWG)  
publication of ISO 9735-3**

**equivalent to the official ISO publication:  
ISO 9735-3** (Second edition 2002-07-01)

**Electronic data interchange for  
administration, commerce and transport  
(EDIFACT) — Application level syntax rules  
(Syntax version number: 4, Syntax release  
number: 1) —**

Part 3:

**Syntax rules specific to interactive EDI**



## Contents

Page

Foreword.....	iv
Introduction.....	vi
1 <b>Scope</b> .....	1
2 <b>Conformance</b> .....	1
3 <b>Normative references</b> .....	2
4 <b>Terms and definitions</b> .....	2
5 <b>I-EDI interchange structure</b> .....	2
6 <b>I-EDI message within a transaction</b> .....	3
7 <b>Dialogue control</b> .....	5
<b>Annex A</b> (informative) <b>Examples illustrating segment sequences</b> .....	7
<b>Annex B</b> (informative) <b>I-EDI functions, states and events</b> .....	10
<b>Annex C</b> (informative) <b>A model of the I-EDI process</b> .....	16

## Foreword

ISO (the International Organization for Standardization) is a worldwide federation of national standards bodies (ISO member bodies). The work of preparing International Standards is normally carried out through ISO technical committees. Each member body interested in a subject for which a technical committee has been established has the right to be represented on that committee. International organizations, governmental and non-governmental, in liaison with ISO, also take part in the work. ISO collaborates closely with the International Electrotechnical Commission (IEC) on all matters of electrotechnical standardization.

International Standards are drafted in accordance with the rules given in the ISO/IEC Directives, Part 3.

The main task of technical committees is to prepare International Standards. Draft International Standards adopted by the technical committees are circulated to the member bodies for voting. Publication as an International Standard requires approval by at least 75 % of the member bodies casting a vote.

Attention is drawn to the possibility that some of the elements of this part of ISO 9735 may be the subject of patent rights. ISO shall not be held responsible for identifying any or all such patent rights.

ISO 9735-3 was prepared by Technical Committee ISO/TC 154, *Processes, data elements and documents in commerce, industry and administration* in collaboration with UN/CEFACT through the Joint Syntax Working Group (JSWG).

This second edition cancels and replaces the first edition (ISO 9735-3:1998). However ISO 9735:1988 and its Amendment 1:1992 are provisionally retained for the reasons given in clause 2.

Furthermore, for maintenance reasons the Syntax service directories have been removed from all parts of the ISO 9735 series. They are now consolidated in a new part, ISO 9735-10.

At the time of publication of ISO 9735-1:1998, ISO 9735-10 had been allocated as a part for “Security rules for interactive EDI”. This was subsequently withdrawn because of lack of user support, and as a result, all relevant references to the title “Security rules for interactive EDI” were removed in this second edition of ISO 9735-3.

Definitions from all parts of the ISO 9735 series have been consolidated and included in ISO 9735-1.

ISO 9735 consists of the following parts, under the general title *Electronic data interchange for administration, commerce and transport (EDIFACT) — Application level syntax rules (Syntax version number: 4, Syntax release number: 1)*:

- *Part 1: Syntax rules common to all parts*
- *Part 2: Syntax rules specific to batch EDI*
- *Part 3: Syntax rules specific to interactive EDI*
- *Part 4: Syntax and service report message for batch EDI (message type — CONTRL)*
- *Part 5: Security rules for batch EDI (authenticity, integrity and non-repudiation of origin)*
- *Part 6: Secure authentication and acknowledgement message (message type — AUTACK)*
- *Part 7: Security rules for batch EDI (confidentiality)*
- *Part 8: Associated data in EDI*
- *Part 9: Security key and certificate management message (message type — KEYMAN)*

— *Part 10: Syntax service directories*

Further parts may be added in the future.

Annexes A, B and C of this part of ISO 9735 are for information only.

## Introduction

This part of ISO 9735 includes the rules at the application level for the structuring of data in the interchange of electronic messages in an open environment, based on the requirements of either batch or interactive processing. These rules have been agreed by the United Nations Economic Commission for Europe (UN/ECE) as syntax rules for Electronic Data Interchange for Administration, Commerce and Transport (EDIFACT) and are part of the United Nations Trade Data Interchange Directory (UNTDID) which also includes both batch and interactive Message Design Guidelines.

This part of ISO 9735 may be used in any application, but messages using these rules may only be referred to as EDIFACT messages if they comply with other guidelines, rules and directories in the UNTDID. For UN/EDIFACT interactive messages, the message design rules for interactive usage apply. These rules are maintained in the UNTDID.

Communications specifications and protocols are outside the scope of this part of ISO 9735.

This is a new part, which has been added to ISO 9735. It provides for the exchange of EDIFACT messages in an interactive (conversational) EDI environment.

Interactive EDI (I-EDI) is characterized by the following:

- a formalized association between the two parties using a dialogue;
- the ability, dynamically, to direct the course of the I-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;
- a transaction is a controlled set of dialogues which can take place between two or more parties.

These characteristics differentiate I-EDI from batch EDI which is specified in ISO 9735-2 (syntax rules specific to batch EDI).

For consistency and in order to simplify the implementation of the standard for those users who wish to utilize both batch and interactive processing, this part of ISO 9735 has been aligned as far as possible with ISO 9735-2.

# Electronic data interchange for administration, commerce and transport (EDIFACT) — Application level syntax rules (Syntax version number: 4, Syntax release number: 1) —

## Part 3: Syntax rules specific to interactive EDI

### 1 Scope

This part of ISO 9735 specifies syntax rules specifically for the transfer of interactive messages to be interchanged between computer application systems. For the transfer of packages in an interactive environment, see ISO 9735-8.

### 2 Conformance

Whereas this part shall use a version number of “4” in the mandatory data element 0002 (Syntax version number), and shall use a release number of “01” in the conditional data element 0076 (Syntax release number), each of which appear in the segment UNB (Interchange header), interchanges continuing to use the syntax defined in the earlier published versions shall use the following Syntax version numbers, in order to differentiate them from each other and from this part:

- ISO 9735:1988: *Syntax version number: 1*
- ISO 9735:1988 (amended and reprinted in 1990): *Syntax version number: 2*
- ISO 9735:1988 and its Amendment 1:1992: *Syntax version number: 3*
- ISO 9735:1998: *Syntax version number: 4*

Conformance to a standard means that all of its requirements, including all options, are supported. If all options are not supported, any claim of conformance shall include a statement which identifies those options to which conformance is claimed.

Data that is interchanged is in conformance if the structure and representation of the data conforms to the syntax rules specified in this part of ISO 9735.

Devices supporting this part of ISO 9735 are in conformance when they are capable of creating and/or interpreting the data structured and represented in conformance with the standard.

Conformance to this part of ISO 9735 shall include conformance to ISO 9735-1 and ISO 9735-10.

When identified in this part of ISO 9735, provisions defined in related standards shall form part of the conformance criteria.

### 3 Normative references

The following normative documents contain provisions which, through reference in this text, constitute provisions of this part of ISO 9735. For dated references, subsequent amendments to, or revisions of, any of these publications do not apply. However, parties to agreements based on this part of ISO 9735 are encouraged to investigate the possibility of applying the most recent editions of the normative documents indicated below. For undated references, the latest edition of the normative document referred to applies. Members of ISO and IEC maintain registers of currently valid International Standards.

ISO 9735-1:2002, *Electronic data interchange for administration, commerce and transport (EDIFACT) — Application level syntax rules (Syntax version number: 4, Syntax release number: 1) — Part 1: Syntax rules common to all parts*

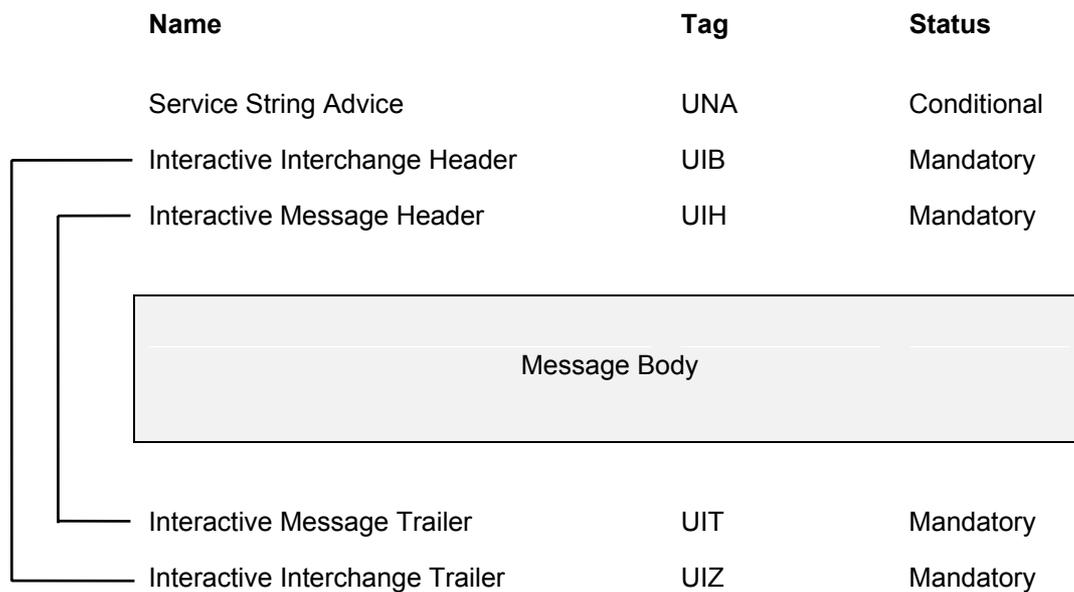
ISO 9735-10:2002, *Electronic data interchange for administration, commerce and transport (EDIFACT) — Application level syntax rules (Syntax version number: 4, Syntax release number: 1) — Part 10: Syntax service directories*

### 4 Terms and definitions

For the purposes of this part of ISO 9735, the terms and definitions given in ISO 9735-1 apply.

### 5 I-EDI interchange structure

The service string advice (if used) and the header and trailer service segments shall appear in an I-EDI interchange in the order shown in Figure 1.



**Figure 1 — I-EDI interchange structure**

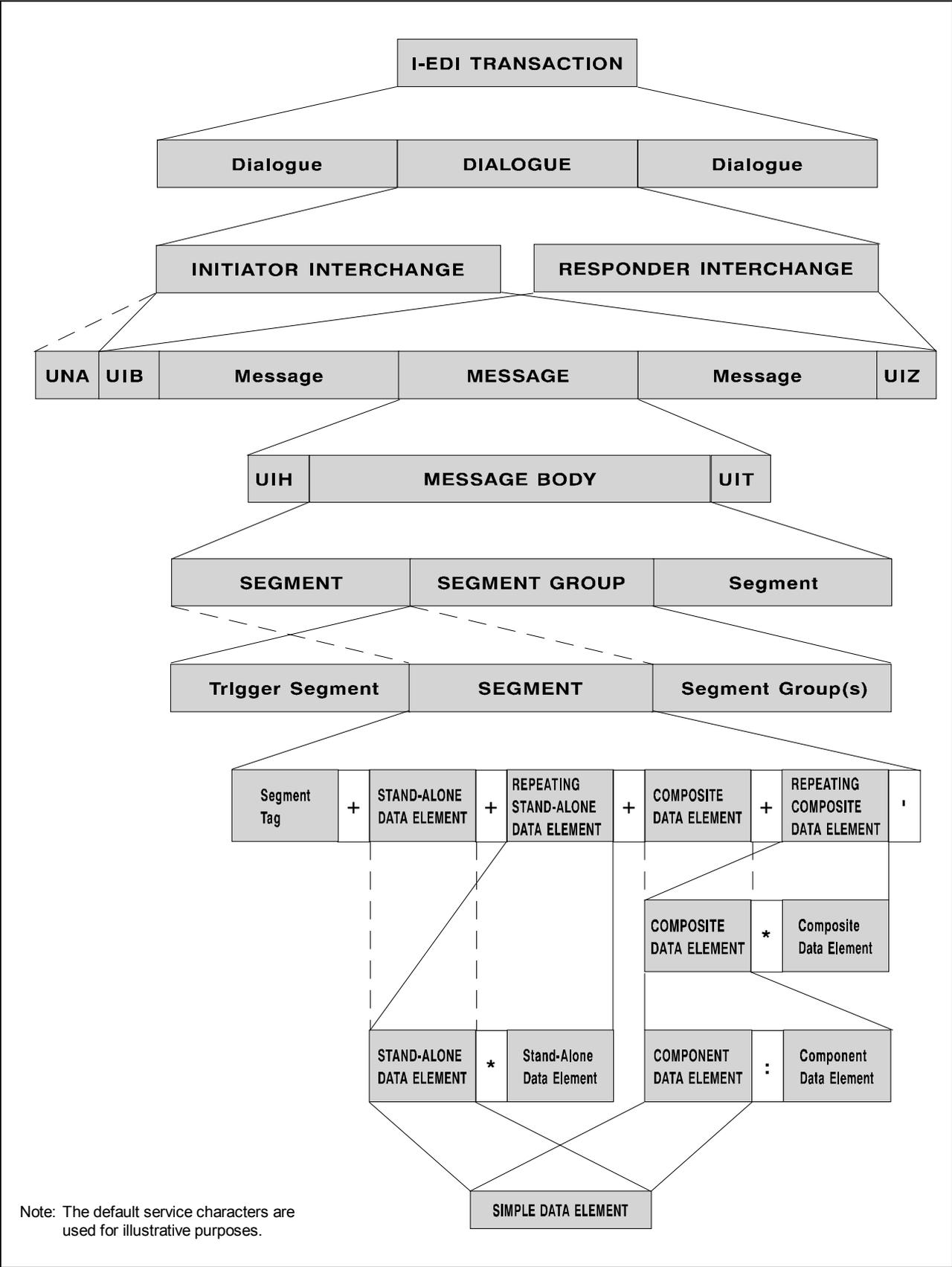
In Figure 1, the lines to the left show the pairing of header and trailer segments. For simplicity, an interchange containing only one message is shown.

For the specification of the service string advice, see ISO 9735-1:2002, annex A.

For the specification of the interactive header and trailer segments, see ISO 9735-10.

NOTE Segments for use in UN/EDIFACT messages are defined in the United Nations Trade Data Interchange Directory (UNTDID).

6 I-EDI message within a transaction



NOTE The default service characters are used for illustrative purposes.

Figure 2 — I-EDI message within a transaction

**Legend:**

An **I-EDI TRANSACTION** contains:

- Dialogue(s)

A **DIALOGUE** contains:

- an Initiator Interchange
- a corresponding Responder Interchange

An **INITIATOR INTERCHANGE** contains:

- UNA, Service String Advice, if used
- UIB, Interactive Interchange Header
- message(s), if used
- UIZ, Interactive Interchange Trailer

A **RESPONDER INTERCHANGE** contains:

- UIB, Interactive Interchange Header
- message(s), if used
- UIZ, Interactive Interchange Trailer

A **MESSAGE** contains:

- UIH, Interactive Message Header
- a message body
- UIT, Interactive Message Trailer

A **MESSAGE BODY** contains:

- segment(s) and/or segment group(s)

A **SEGMENT GROUP** contains:

- a trigger segment
- segments(s) and possibly segment group(s)

A **SEGMENT** contains:

- a segment tag
- stand-alone data element(s) and/or composite data element(s) and/or repeating stand-alone data element(s) and/or repeating composite data element(s)

A **REPEATING STAND-ALONE DATA ELEMENT** is:

- one or more occurrences of the same stand-alone data element

A **REPEATING COMPOSITE DATA ELEMENT** is:

- one or more occurrences of the same composite data element

A **COMPOSITE DATA ELEMENT** contains:

- two or more component data elements

A **COMPONENT DATA ELEMENT** is:

- a simple data element

A **STAND-ALONE DATA ELEMENT** is:

- a simple data element

A **SIMPLE DATA ELEMENT** contains:

- a single data element value

**Figure 2 — I-EDI message within a transaction** *(continued)*

## 7 Dialogue control

An I-EDI transaction, which is an instance of a particular scenario, 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.

The following transfers shall take place:

- An initiator begins a dialogue by sending an interchange header segment to a responder, optionally preceded by a UNA, and optionally followed by a message.
- The responder replies to the initiator with an interchange header segment, optionally followed by a message (note that the values of the UNA sent by the initiator also apply to the responder).
- The initiator sends a query message to the responder.
- The responder replies to the initiator with a response message.
- The initiator and responder exchange additional messages, as necessary.
- The initiator ends the dialogue by sending an interchange trailer segment to the responder, optionally preceded by a message.
- The responder replies to the initiator with an interchange trailer segment, optionally preceded by a message.

The following variations are possible:

For each message from the initiator to the responder there may be zero, one, or more than one message from the responder to the initiator, and vice versa.

UIR service segments may be interleaved with messages.

A dialogue can be prematurely terminated at any time by either party, by using a UIR service segment.

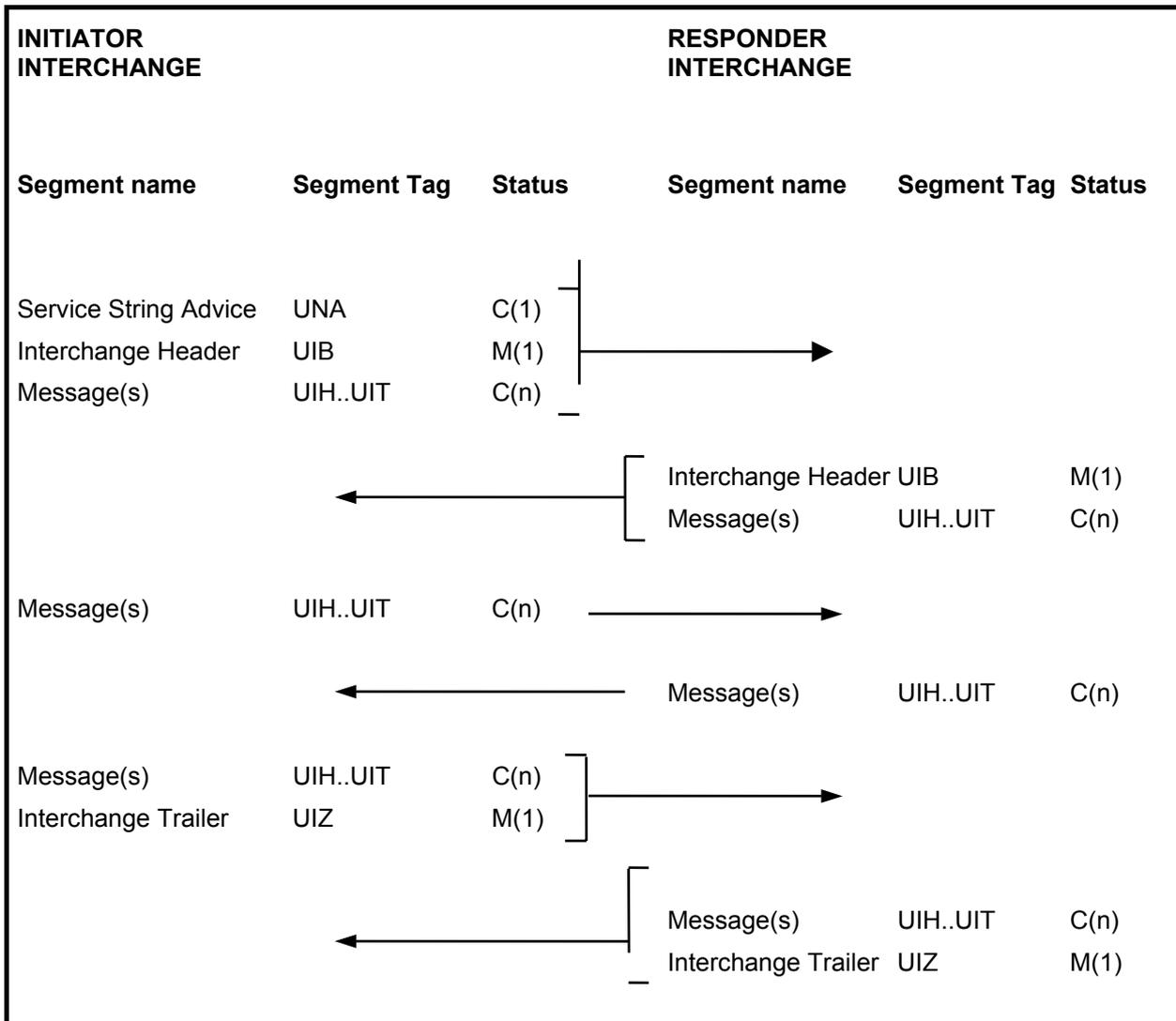
A message or messages may be combined with:

- the interchange header, or
- the interchange trailer, or
- both the interchange header and the interchange trailer (a complete dialogue).

Whilst exchange of data controlled by the initiator is a common mode of operation for interactive applications, the I-EDI syntax does not exclude other modes of operation.

See annex A for examples.

Figure 3 shows a flow diagram of two interchanges which together form a dialogue.



The arrows indicate the direction of data flow. Note that UNA is only sent by initiator.

The status indicates Mandatory (M) or Conditional (C), together with an indication of allowed repetition.

**Figure 3 — Flow diagram of two I-EDI interchanges**

## Annex A (informative)

### Examples illustrating segment sequences

**Example a)** Message pairs with first and final message combined with interchange header and trailer:

Initiator UIB...UIH...Segment(s) and/or Segment Group(s)...UIT

Responder UIB...UIH...Segment(s) and/or Segment Group(s)...UIT

Initiator UIH...Segment(s) and/or Segment Group(s)...UIT

Responder UIH...Segment(s) and/or Segment Group(s)...UIT

Initiator UIH...Segment(s) and/or Segment Group(s)...UIT

Responder UIH...Segment(s) and/or Segment Group(s)...UIT

etc.

Initiator UIH...Segment(s) and/or Segment Group(s)...UIT...UIZ

Responder UIH...Segment(s) and/or Segment Group(s)...UIT...UIZ

**Example b)** Message pairs with separate interchange header and trailer, and with UNA (note that UNA is only sent by initiator, and therefore also applies to responder):

Initiator UNA...UIB

Responder UIB

Initiator UIH...Segment(s) and/or Segment Group(s)...UIT

Responder UIH...Segment(s) and/or Segment Group(s)...UIT

Initiator UIH...Segment(s) and/or Segment Group(s)...UIT

Responder UIH...Segment(s) and/or Segment Group(s)...UIT

Initiator UIH...Segment(s) and/or Segment Group(s)...UIT

Responder UIH...Segment(s) and/or Segment Group(s)...UIT

etc.

Initiator UIZ

Responder UIZ

**Example c)** A single message combined with interchange header and trailer (a complete dialogue):

Initiator UIB... UIH...Segment(s) and/or Segment Group(s)...UIT...UIZ

Responder UIB... UIH...Segment(s) and/or Segment Group(s)...UIT...UIZ

**Example d)** Multi-message sequences with final message combined with interchange trailer:

Initiator UIB

Responder UIB

Initiator UIH....Segment(s) and/or Segment Group(s)...UIT

Responder UIH(F).Segment(s) and/or Segment Group(s)...UIT

UIH(L).Segment(s) and/or Segment Group(s)...UIT

## ISO 9735-3:2002

Initiator UIH...Segment(s) and/or Segment Group(s)...UIT...UIZ  
Responder UIH...Segment(s) and/or Segment Group(s)...UIT...UIZ

**Example e)** Message pairs with separate interchange header and trailer, with UNA, and with embedded UIR pairs:

Initiator UNA...UIB  
Responder UIB  
Initiator UIH...Segment(s) and/or Segment Group(s)...UIT  
Responder UIH...Segment(s) and/or Segment Group(s)...UIT  
etc.  
Initiator UIR...Report function, coded = 'n' (Query status)  
Responder UIR...Report function, coded = 'n' (Status report)  
Initiator UIH...Segment(s) and/or Segment Group(s)...UIT  
Responder UIH...Segment(s) and/or Segment Group(s)...UIT  
etc.  
Initiator UIZ  
Responder UIZ

**Example f)** Message pairs with separate interchange header and trailer, and with UNA. UIR used to report severe error detected by Responder:

Initiator UNA...UIB  
Responder UIB  
Initiator UIH...Segment(s) and/or Segment Group(s)...UIT  
Responder UIH...Segment(s) and/or Segment Group(s)...UIT  
Initiator UIH...Segment(s) and/or Segment Group(s)...UIT  
Responder UIH...Segment(s) and/or Segment Group(s)...UIT  
Initiator UIH...Segment(s) and/or Segment Group(s)...UIT  
Responder UIR...Report function, coded = 'n' (Abort dialogue)  
Reason code indicates problem area  
No further exchanges in this dialogue.

**Example g)** Dialogue unable to start. UIR used by Responder to report Start Dialogue Reject:

Initiator UNA...UIB  
Responder UIR...Report function, coded = 'n' (Start dialogue reject)  
Reason code indicates problem area  
No further exchanges in this dialogue.

**Example h)** Message pairs with first and final message combined with interchange header and trailer, and using pause and continue:

Initiator UIB...UIH...Segment(s) and/or Segment Group(s)...UIT  
Responder UIB...UIH...Segment(s) and/or Segment Group(s)...UIT

Initiator UIH...Segment(s) and/or Segment Group(s)...UIT  
Responder UIH...Segment(s) and/or Segment Group(s)...UIT  
Responder UIR...Report function, coded = 'n' (Pause dialogue)  
Reason code indicates reason for pause; e.g. low resources  
No more data flows in dialogue until:-  
Some time later...  
Responder UIR...Report function, coded = 'n' (Continue dialogue)  
Initiator UIH...Segment(s) and/or Segment Group(s)...UIT  
Responder UIH...Segment(s) and/or Segment Group(s)...UIT  
etc.  
Initiator UIH...Segment(s) and/or Segment Group(s)...UIT...UIZ  
Responder UIH...Segment(s) and/or Segment Group(s)...UIT...UIZ

## **Annex B** (informative)

### **I-EDI functions, states and events**

#### **B.1 I-EDI functions**

In the following sections, the word “application” can mean either the main application program, or that part of the I-EDI handler which manages the I-EDI dialogue, depending upon the implementation. The word “association” here refers to a logical relationship between two applications, not to any other meaning which may be used in other standards. Note that the following function points do not necessarily map to a single service segment or message.

##### ***Start dialogue request***

Allows an application to pass sufficient information to a remote application to enable an association between the two applications to be initiated.

##### ***Start dialogue confirm***

Allows the remote application to pass sufficient information to an initiating application to inform it that the association has been accepted.

##### ***Start dialogue reject***

Allows the remote application to pass sufficient information to an initiating application to inform it that the association cannot be initiated.

##### ***Transfer data***

Allows an application to pass business information to another application.

##### ***Request status***

Allows an application to request status or control information from the other application, in the association.

##### ***Report status (reply)***

Allows an application to send status or control information to the other application in the association. This can be sent as a reply to a request status, or as an unsolicited incident report.

##### ***Report error (reply)***

Allows an application to report a syntax error to the other application. It also allows an application error together with a start dialogue reject to be reported.

##### ***Pause dialogue***

Allows an application to request that the dialogue be paused until the same application issues a continue dialogue.

##### ***Continue dialogue***

Allows an application to request that the dialogue that it has previously paused be continued.

##### ***Abort dialogue***

Allows an application unconditionally to end an association when it is unable to continue with that association.

##### ***End dialogue request***

Allows an application to request the other application in the association to close the association, typically at the normal end of a business transaction.

**End dialogue confirm**

Allows a responding application to confirm to the requesting application that the association is terminated.

**Complete dialogue request**

Allows an application to pass sufficient information to a remote application to enable an association between the two applications to be initiated, data to be sent, and the association termination requested in a single transfer.

**Complete dialogue confirm**

Allows the remote application to pass sufficient information to an initiating application to inform it that the association has been accepted, data has been returned, and the association has been terminated in a single transfer.

**B.2 Data requirements**

Table B.1 indicates how the abstract I-EDI functions can be mapped to I-EDI service segments and messages. The S (Status) field indicates whether a segment is mandatory or conditional within an I-EDI function. The R field indicates the number of repetitions

**Table B.1 — Functions mapped to service segments**

Functions	Segments	S	R
Start Dialogue Request	UNA	C	1
	UIB	M	1
	(UIH <data> UIT)	C	n
Start Dialogue Confirm	UIB	M	1
	(UIH <data> UIT)	C	n
Start Dialogue Reject	UIR	M	1
Transfer Data	UIH <data> UIT	M	n
Request Status	UIR	M	1
Report Status	UIR	M	1
Report Error	UIR	M	1
Abort	UIR	M	1
End Dialogue Request	(UIH <data> UIT)	C	n
	UIZ	M	1
End Dialogue Confirm	(UIH <data> UIT)	C	n
	UIZ	M	1
Complete Dialogue Request	UNA	C	1
	UIB	M	1
	(UIH <data> UIT)	M	n
	UIZ	M	1
Complete Dialogue Confirm	UIB	M	1
	(UIH <data> UIT)	M	n
	UIZ	M	1

### B.3 Sequencing of I-EDI functions

#### B.3.1 General

The I-EDI protocol is described in the following diagram and tables in terms of the states the protocol can be in, and the events which cause a transition from one state to another. As each event occurs the protocol "machine" moves automatically from state to state. The number of valid states the I-EDI protocol can be in is finite.

The dialogue state diagram (Figure B.1) shows the states of the I-EDI protocol, the events affecting the I-EDI protocol, and the transitions from state to state. This is further formalized as a state-event matrix (Table B.4) which is a two dimensional representation of the I-EDI protocol machine. The two dimensions are states and events, and the intersection of state and event gives the transition to the next state for that particular event; all other events are error conditions.

#### B.3.2 State

At any instant, the I-EDI protocol can be said to be in one of a finite number of states. Table B.2 lists the valid states for the I-EDI protocol and describes the purpose of the state.

**Table B.2 — States**

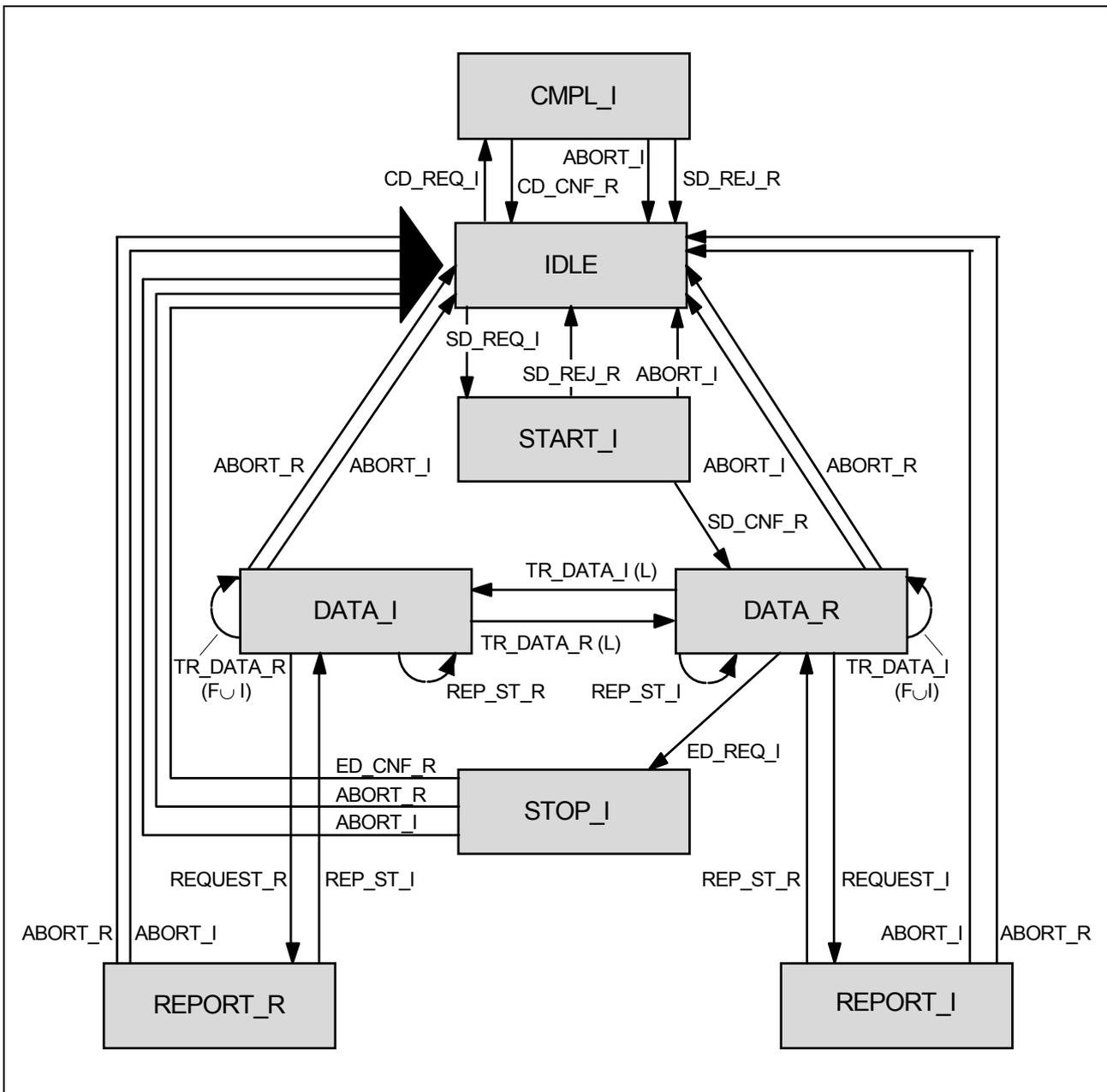
State	Description
IDLE	No association exists and no responses are outstanding
START_I	Waiting for 'Start Dialogue Confirm' from responder to initiator
DATA_I	Waiting for 'Transfer Data' from responder to initiator
DATA_R	Waiting for 'Transfer Data' from initiator to responder
REPORT_I	Waiting for 'Report Status' from responder to initiator
REPORT_R	Waiting for 'Report Status' from initiator to responder
STOP_I	Waiting for 'End Dialogue Confirm' from responder to initiator
CMPL_I	Waiting for 'Complete Dialogue Confirm' from responder to initiator

#### B.3.3 Event

Table B.3 lists the valid events for the I-EDI protocol and describes the conditions attached to those events. These events are usually caused by data objects or control objects being transferred through the protocol handler.

Table B.3 — Events

Event	Function	Direction
SD_REQ_I	Start Dialogue Request	From Initiator to Responder
SD_CNF_R	Start Dialogue Confirm	From Responder to Initiator
SD_REJ_R	Start Dialogue Reject	From Responder to Initiator
TR_DATA_I	Transfer Data	From Initiator to Responder
TR_DATA_R	Transfer Data	From Responder to Initiator
ED_REQ_I	End Dialogue Request	From Initiator to Responder
ED_CNF_R	End Dialogue Confirm	From Responder to Initiator
ABORT_I	Abort Dialogue	From Initiator to Responder
ABORT_R	Abort Dialogue	From Responder to Initiator
REQUEST_I	Request Status	From Initiator to Responder
REQUEST_R	Request Status	From Responder to Initiator
REP_ST_I	Report Status	From Initiator to Responder
REP_ST_R	Report Status	From Responder to Initiator
CD_REQ_I	Complete Dialogue Request	From Initiator to Responder
CD_CNF_R	Complete Dialogue Confirm	From Responder to Initiator



**Key**

- (F ∪ I) First or Intermediate message
- (L) Last message
- \_I suffix Initiator
- \_R suffix Responder

Figure B.1 — Dialogue state diagram

Table B.4 — State-event matrix

Event	State							
	IDLE	START_I	DATA_I	DATA_R	STOP_I	CMPL_I	REPORT_I	REPORT_R
SD_REQ_I	START_I							
SD_CNF_R		DATA_R						
SD_REJ_R		IDLE				IDLE		
TR_DATA_I(FUI)				DATA_R				
TR_DATA_I(L)				DATA_I				
TR_DATA_R(FUI)			DATA_I					
TR_DATA_R(L)			DATA_R					
ED_REQ_I				STOP_I				
ED_CNF_R					IDLE			
ABORT_I		IDLE	IDLE <sup>a</sup>	IDLE	IDLE <sup>a</sup>	IDLE	IDLE <sup>a</sup>	IDLE
ABORT_R			IDLE	IDLE <sup>a</sup>	IDLE		IDLE	IDLE <sup>a</sup>
REQUEST_I				REPORT_I				
REQUEST_R			REPORT_R					
REP_ST_I				DATA_R				DATA_I
REP_ST_R			DATA_I				DATA_R	
CD_REQ_I	CMPL_I							
CD_CNF_R						IDLE		

<sup>a</sup> Might not be possible if communication medium is half-duplex.

## Annex C (informative)

### A model of the I-EDI process

#### C.1 Summary of I-EDI

Interactive EDI is a series of exchanges of information between the applications of independent parties in order to accomplish a joint task, where subsequent exchanges may depend upon the results of previous exchanges. Strict timing constraints frequently apply. Applications which are inherently interactive include airline reservation systems; healthcare pharmacy, claims submission and eligibility verification; and remote automated teller machines for banks.

Initially, Interactive EDI is aimed at those applications where the initiating party, sends data to the responder, and the responder sends data back in reply. This alternate exchange of data controlled by the initiator is by far the most common way of working among existing interactive applications, but the I-EDI syntax does not exclude other modes of working.

The definition of interactive EDI depends upon the definition of EDI in general. The approach taken towards EDI in this document has been based on the "Report on the Open-edi Conceptual Model" prepared by the EDI Special Working Group of ISO/IEC JTC 1. Characteristics of the "Open-edi Conceptual Model" include:

- generalizing EDI beyond trade;
- defining EDI as "open" (available to all parties, according to standards and without requiring special bipartite agreements);
- co-ordinating EDI with other International Standards in communications, modelling and open environments.

Two major elements of the business context of EDI have made the development of interactive EDI necessary. The first is pressure from the market on many organizations (not just in the private sector) for more competitive, more responsive performance. Many fundamental processes must, in fact, be "remodelled" to respond to these pressures. The second element is the desire for standard solutions, in contrast to the current proprietary (and therefore "non-Open-edi") situations.

The following guiding principles were adopted in defining I-EDI requirements:

- Ease of user implementation is paramount and standards should define their elements accordingly.
- Interactive EDI mechanisms should be fully compatible with and where possible identical to those for other forms of EDI.
- The required functions should be available no matter what communications methods are used.
- Wherever equivalent functions are available in the underlying communications protocols (e.g. X.25, OSI Transaction Processing) they may be used.
- EDI standards should be fully harmonized with all other relevant International Standards.

The business and functional models, and the contents of the information required in interactive EDI service segments, have been described below, to present the characteristics and requirements of interactive EDI independently of an underlying architecture. It is recommended though, but not mandatory, that the relevant ISO protocols be used to carry I-EDI data.

## **C.2 Business requirements of Interactive EDI**

- Enable consistent completion of a single business transaction between two or more business partners.
- Interactive conversational activities must be supported.
- Provide for the handling of high volumes of business information, in a timely manner.
- Provide the means for business information to be passed securely between business partners.

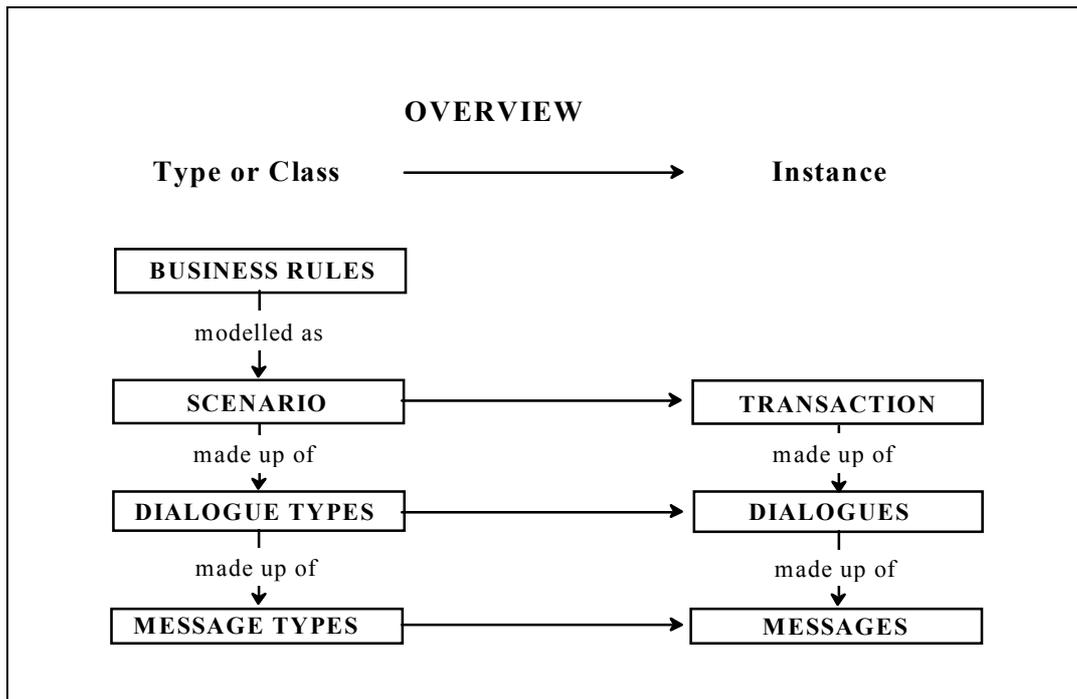
## **C.3 Functional requirements to support business requirements**

Within a business transaction:

- Enable co-operation between applications.
- Enable multiple bilateral conversations.
- Enable the co-ordination of bilateral conversations.
- Enable cascading of bilateral conversations.
- Enable the two way exchange of I-EDI messages within a bilateral conversation.
- Provide efficient mechanisms to allow for sub-second response times.
- Support high transaction volumes through reduced overhead.
- Security shall be provided by common UN/EDIFACT security, or other standards.

## **C.4 Business model**

The I-EDI dialogue is separate from and independent of, dialogue as a term used in other ISO documents.



**Figure C.1 — Overview of types and instances**

A scenario is a formal specification of a group of business activities that take place between parties to achieve a particular business objective. A scenario models the relationships and interactions among the parties.

A *transaction* is an instance of a *scenario*. When roles are played in a *scenario* to execute an actual business transaction, a *transaction* is created. *Transactions* are outlined here simply to clarify the context of the *dialogue*.

In order to carry out a *transaction*, the various parties involved in the business transaction communicate bilaterally using *dialogues* for the I-EDI part of the *transaction*. *Transactions* have the potential of grouping a number of *dialogues*. But many *scenarios* can be modelled which contain only a single *dialogue type* between two parties, an instance of which is a *transaction* containing only a single *dialogue* between two parties.

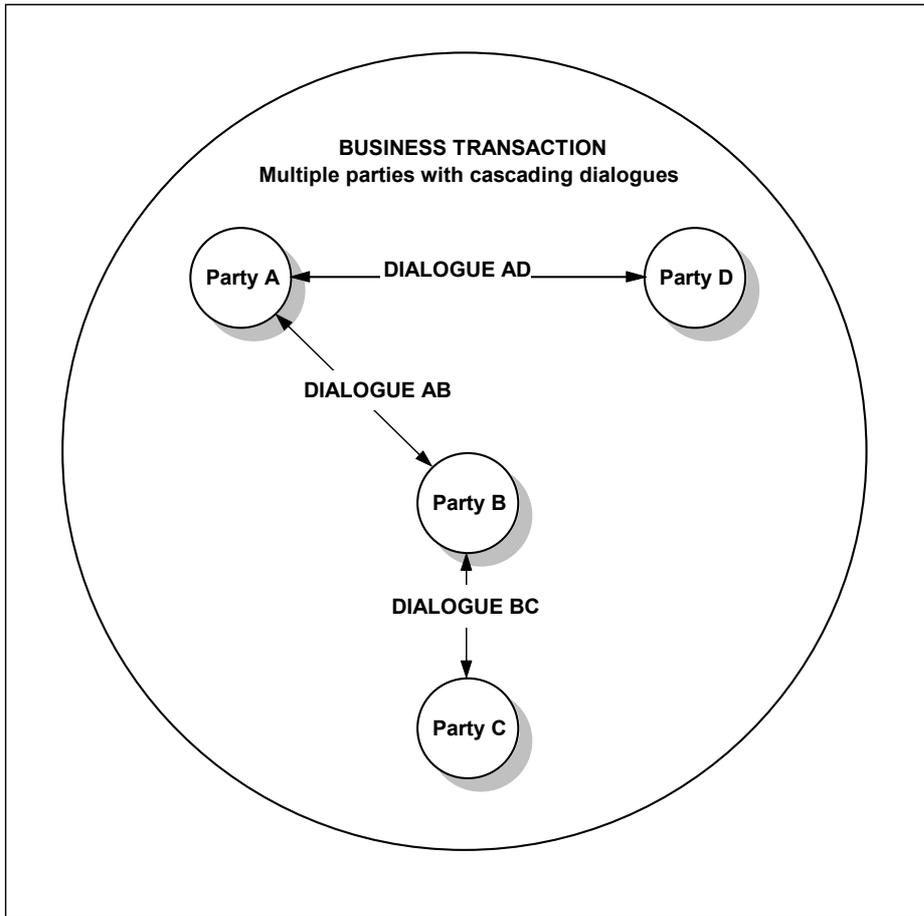


Figure C.2 — Illustration of a business transaction

Dialogues can be grouped together within the same transaction. Multiple dialogues can take place between the same or different pairs of parties.

**C.5 Functional Model**

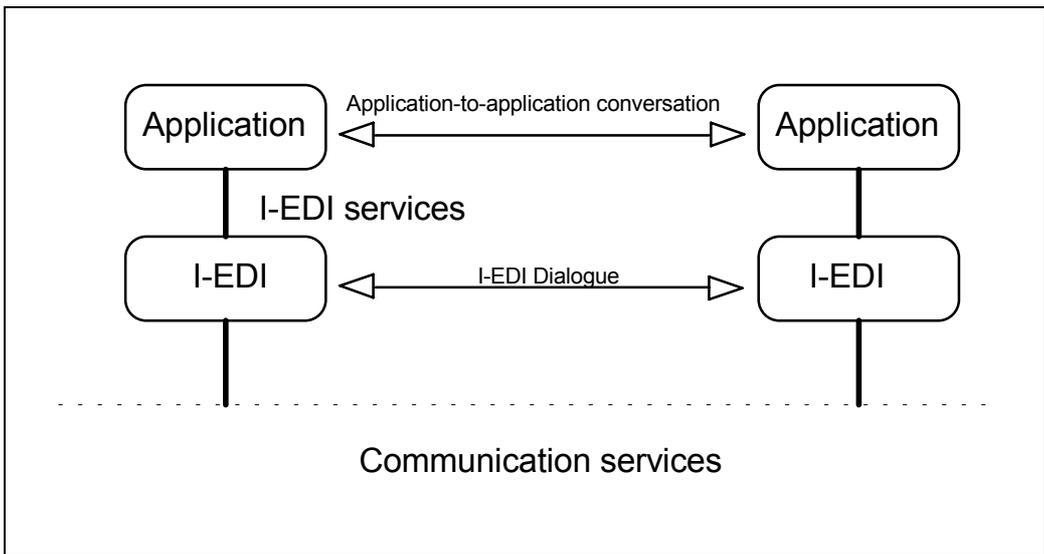


Figure C.3 — Dialogue

## C.6 Minimum communication requirements

The communications shall:

- be error free;
- deliver data in the order in which it was transmitted;
- allow bi-directional data flows;
- provide detection and reporting of lost logical associations;
- provide a persistent logical association between applications (e.g. session, conversation, etc.). Each I-EDI dialogue would then have its own unique logical association. If this requirement cannot be met, implementers will have to deal with problems associated with separators and character set recognition.

## C.7 Data requirements

The following list is an attempt to provide a list of the data which are needed to perform the named functions. The list was used for modelling the service segments but the presence of a function here does not necessarily guarantee the existence of a unique service segment, as some service segments perform multiple functions.

### **Start dialogue request; (UNA, UIB and optional message)**

- Separator characters
- Character set
- Syntax identifier
- Dialogue reference
- Business transaction reference
- Scenario identifier
- Dialogue identifier
- Sender identifier
- Recipient identifier
- Date and time
- Duplicate indicator
- Test indicator
- Security information

### **Start dialogue confirm; (UIB and optional message)**

- Syntax identifier
- Dialogue reference
- Business transaction reference
- Scenario identifier
- Dialogue identifier

- Sender identifier
- Recipient identifier
- Date and time
- Duplicate indicator
- Test indicator
- Response information
- Security information

**Send data; (Message = UIH, query or command, UIT)**

- Message identifier or type
- Message reference
- Dialogue reference
- Status of transfer
- Date and time
- Test Indicator

**Receive data; (Message = UIH, response, UIT)**

- Message identifier or type
- Message reference
- Dialogue reference
- Status of transfer
- Date and time
- Test Indicator

**Request status; (UIR)**

- Dialogue reference
- Function (= Query)
- Date and time

**Report status; (UIR)**

- Dialogue reference
- Function (= Report)
- Reason code
- Other information from message in error
- Date and time

**Start dialogue reject; (UIR)**

- Dialogue reference

## **ISO 9735-3:2002**

- Function (= Start dialogue reject)
- Reason code
- Other information from dialogue in error
- Date and time

### **Pause dialogue; (UIR)**

- Dialogue reference
- Function (= Paused)
- Reason code
- Date and time

### **Continue dialogue; (UIR)**

- Dialogue reference
- Function (= Continue)
- Date and time

### **Abort; (UIR)**

- Dialogue reference
- Function (= Abort dialogue)
- Reason code
- Other Information from message in error
- Date and time

### **End dialogue request; (optional message and UIZ)**

- Dialogue reference
- Control count of messages sent
- Duplicate indicator

### **End dialogue confirm; (optional message and UIZ)**

- Dialogue reference
- Control count of messages sent