TECHNICAL REPORT

IEC TR 62325-501

First edition
2005-02

Framework for energy market communications –

Part 501:
General guidelines for use of ebXML

Reference number
IEC/TR 62325-501:2005(E)

Publication numbering

As from 1 January 1997 all IEC publications are issued with a designation in the 60000 series. For example, IEC 34-1 is now referred to as IEC 60034-1.

Consolidated editions

The IEC is now publishing consolidated versions of its publications. For example, edition numbers 1.0, 1.1 and 1.2 refer, respectively, to the base publication, the base publication incorporating amendment 1 and the base publication incorporating amendments 1 and 2.

Further information on IEC publications

The technical content of IEC publications is kept under constant review by the IEC, thus ensuring that the content reflects current technology. Information relating to this publication, including its validity, is available in the IEC Catalogue of publications (see below) in addition to new editions, amendments and corrigenda. Information on the subjects under consideration and work in progress undertaken by the technical committee which has prepared this publication, as well as the list of publications issued, is also available from the following:

- **IEC Web Site (www.iec.ch)**

- **Catalogue of IEC publications**

  The on-line catalogue on the IEC web site (www.iec.ch/searchpub) enables you to search by a variety of criteria including text searches, technical committees and date of publication. On-line information is also available on recently issued publications, withdrawn and replaced publications, as well as corrigenda.

- **IEC Just Published**

  This summary of recently issued publications (www.iec.ch/online_news/ justpub) is also available by email. Please contact the Customer Service Centre (see below) for further information.

- **Customer Service Centre**

  If you have any questions regarding this publication or need further assistance, please contact the Customer Service Centre:

  Email: custserv@iec.ch
  Tel: +41 22 919 02 11
  Fax: +41 22 919 03 00

TECHNICAL REPORT

IEC TR 62325-501

First edition
2005-02

Framework for energy market communications –

Part 501:
General guidelines for use of ebXML

International Electrotechnical Commission, 3, rue de Varembé, PO Box 131, CH-1211 Geneva 20, Switzerland
Telephone: +41 22 919 02 11 Telefax: +41 22 919 03 00 E-mail: inmail@iec.ch Web: www.iec.ch

Commission Electrotechnique Internationale
International Electrotechnical Commission
Международная Электротехническая Комиссия

PRICE CODE X

For price, see current catalogue

CONTENTS

FOREWORD....4
INTRODUCTION....6

1 Scope....7
2 Normative references....7
3 Terms, definitions and abbreviations....8
3.1 Terms and definitions....8
3.2 Abbreviations....8
4 Generic technical architecture....9
4.1 General....9
4.2 Architecture....9
5 Comparison with other solutions....21
6 ebXML configuration....22
6.1 General....22
6.2 Business Process Specification....22
6.3 Collaboration Protocol Profile and Agreement (CPA/CPP)....23
6.4 Mapping to the UN/CEFACT Modelling Mythology UMM....24
6.5 Use of registries....24
6.6 Use of intermediates....24
7 Profile of the architecture....25
8 Migration scenarios....25
8.1 General....25
8.2 Recommended scenarios....26
8.3 ebXML and EDIFACT (X12) in parallel....26
8.4 ebXML with EDIFACT (X12) payload....27
8.5 EDIFACT (X12)/XML conversion....27

Annex A (informative) Scheduling example....28

Figure 1 – Process specification....11
Figure 2 – Business transaction....12
Figure 3 – Binary collaboration....13
Figure 4 – Multiparty collaboration....13
Figure 5 – Business document....14
Figure 6 – Collaboration protocol profile....15
Figure 7 – Collaboration rule....15
Figure 8 – Delivery channel....16
Figure 9 – Transport....16
Figure 10 – Document exchange....16
Figure 11 – ebXML binding of reliability and security....17
Figure 12 – SOAP envelope header and body....17
Figure 13 – Message header....18
Figure 14 – ebXML configuration interfaces....21

Figure 15 – Migration from EDIFACT to ebXML and use of data networks 26
Figure 16 – Migration from EDIFACT to ebXML 26
Figure A.1 – Schedule planning transmission process 29
Figure A.2 – Sequence diagram for the document flow without business signals 30
Figure A.3 – ebXML registry scenario for scheduling 31
Figure A.4 – ebXML client where the messages are sent and received 32
Figure A.5 – Visualization of a message 32
Figure A.6 – Interface used to compose messages 33
Figure A.7 – ebXML BPSS DTD 38
Figure A.8 – ebXML BPSS of scheduling phase 1 – 3 in XML 39
Figure A.9 – CPP of the Transmission System Operator for scheduling 41
Figure A.10 – CPP of the Balance Responsible Party for scheduling 42
Figure A.11 – CPA for scheduling 44
Figure A.12 – XML message document example for schedules 45

Table 1 – Relation of UMM with ebXML 19
Table 2 – Mapping between UML classes and XML elements 23

INTERNATIONAL ELECTROTECHNICAL COMMISSION

FRAMEWORK FOR ENERGY MARKET COMMUNICATIONS –

Part 501: General guidelines for use of ebXML

FOREWORD

1) The International Electrotechnical Commission (IEC) is a worldwide organization for standardization comprising all national electrotechnical committees (IEC National Committees). The object of IEC is to promote international co-operation on all questions concerning standardization in the electrical and electronic fields. To this end and in addition to other activities, IEC publishes International Standards, Technical Specifications, Technical Reports, Publicly Available Specifications (PAS) and Guides (hereafter referred to as "IEC Publication(s)"). Their preparation is entrusted to technical committees; any IEC National Committee interested in the subject dealt with may participate in this preparatory work. International, governmental and non-governmental organizations liaising with the IEC also participate in this preparation. IEC collaborates closely with the International Organization for Standardization (ISO) in accordance with conditions determined by agreement between the two organizations.
2) The formal decisions or agreements of IEC on technical matters express, as nearly as possible, an international consensus of opinion on the relevant subjects since each technical committee has representation from all interested IEC National Committees.
3) IEC Publications have the form of recommendations for international use and are accepted by IEC National Committees in that sense. While all reasonable efforts are made to ensure that the technical content of IEC Publications is accurate, IEC cannot be held responsible for the way in which they are used or for any misinterpretation by any end user.
4) In order to promote international uniformity, IEC National Committees undertake to apply IEC Publications transparently to the maximum extent possible in their national and regional publications. Any divergence between any IEC Publication and the corresponding national or regional publication shall be clearly indicated in the latter.
5) IEC provides no marking procedure to indicate its approval and cannot be rendered responsible for any equipment declared to be in conformity with an IEC Publication.
6) All users should ensure that they have the latest edition of this publication.
7) No liability shall attach to IEC or its directors, employees, servants or agents including individual experts and members of its technical committees and IEC National Committees for any personal injury, property damage or other damage of any nature whatsoever, whether direct or indirect, or for costs (including legal fees) and expenses arising out of the publication, use of, or reliance upon, this IEC Publication or any other IEC Publications.
8) Attention is drawn to the Normative references cited in this publication. Use of the referenced publications is indispensable for the correct application of this publication.
9) Attention is drawn to the possibility that some of the elements of this IEC Publication may be the subject of patent rights. IEC shall not be held responsible for identifying any or all such patent rights.

The main task of IEC technical committees is to prepare International Standards. However, a technical committee may propose the publication of a technical report when it has collected data of a different kind from that which is normally published as an International Standard, for example "state of the art".

IEC 62325-501, which is a technical report, has been prepared by IEC technical committee 57: Power systems management and associated information exchange.

The IEC 62325 series cancels and replaces IEC 62195 (2000) and its amendment (2002). It constitutes a technical revision.

IEC 62195 (2000) dealt with deregulated energy market communications at an early stage. Its amendment 1 (2002) points out important technological advancements which make it possible to use modern internet technologies based on XML for e-business in energy markets as an alternative to traditional EDI with EDIFACT and X12. The new IEC 62325 framework series for energy market communications currently consisting of IEC 62325-101, IEC 62325-102, IEC 62325-501, and IEC 62325-502 follows this direction and replaces IEC 62195 together with its amendment.

The text of this technical report is based on the following documents:

Enquiry draft	Report on voting
57/706/DTR	57/723/RVC

Full information on the voting for the approval of this technical report can be found in the report on voting indicated in the above table.

This publication has been drafted in accordance with the ISO/IEC Directives, Part 2.

IEC 62325 consists of the following parts, under the general title *Framework for energy market communications*:

Part 101: General guidelines
Part 102: Energy market model example
Part 201: Glossary [1]
Part 3XX: (Titles are still to be determined) [2]
Part 401: Abstract service model [3]
Part 501: General guidelines for use of ebXML
Part 502: Profile of ebXML
Part 503: Abstract service mapping to ebXML [3]
Part 601: General guidelines for use of web services [3]
Part 602: Profile of Web Services [3]
Part 603: Abstract service mapping to web services [3]

The committee has decided that the contents of this publication will remain unchanged until the maintenance result date indicated on the IEC web site under "http://webstore.iec.ch" in the data related to the specific publication. At this date, the publication will be

- reconfirmed,
- withdrawn,
- replaced by a revised edition, or
- amended.

A bilingual edition of this document may be issued at a later date.

---

1 Under consideration. Because the technologies have an inherent own glossary within their standard definitions, this glossary is a placeholder for a glossary for future parts indicated with [2)] including energy market specific terms and definitions.

2 Under consideration. These parts for business content are mentioned for completeness only with a number space as placeholder. They extend the original scope and require an agreed new work item proposal for further work based on an overall strategy how to proceed.

3 Under consideration. These technical parts are mentioned for completeness with provisional title. They extend the original scope and require an agreed new work item proposal for further work.

INTRODUCTION

With the transition of monopoly energy supply structures to deregulated energy markets, the function of the markets depends heavily on seamless e-business communication between market participants. Compared with global e-business, e-business in the energy market is only a small niche. Today EDIFACT or X12 messages, or proprietary HTML and XML solutions based on Internet technologies are being used.

The ‘electronic business Extensible Markup Language’ (ebXML) specification and architecture stems from UN/CEFACT and OASIS (see www.ebXML.org). The technical parts regarding the technical e-business infrastructure have now become the multipart ISO 15000 series “Electronic business eXtensible Markup Language (ebXML)” being complemented in future to cover all technical aspects of ebXML. ebXML is a complete set of specifications and standards to enable secure electronic business using proven, open standards such as TCP/IP, HTTP, SOAP, XML, and SOAP signature and encryption. ebXML is also evolutionary in nature, built on 25 years of EDI experience, designed to work with existing EDI solutions, or be used to develop an emerging class of internet-based electronic business applications based on XML. This means that existing EDI messages (EDIFACT, and X12) as well as XML messages can be exchanged.

FRAMEWORK FOR ENERGY MARKET COMMUNICATIONS –

Part 501: General guidelines for use of ebXML

1 Scope

This part of IEC 62325 provides general guidelines how to use the ebXML technology and architecture in energy markets based on the ISO 15000 ISO series “Electronic business eXtensible Markup Language (ebXML)” together with migration scenarios and an implementation example. For recommended profiles, see IEC 62325-502.

2 Normative references

The following referenced documents are indispensable for the application of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ISO/IEC 14662, *Information Technology – Open-edi reference model*

ISO 7372, *Trade data interchange – Trade data elements directory*

ISO 9735 (all parts), *Electronic data interchange for administration, commerce and transport (EDIFACT)*

ISO/TS 15000-1:2004, *Electronic business eXtensible Markup Language (ebXML) – Part 1: Collaboration-protocol profile and agreement specification (ebCPP)*

ISO/TS 15000-2:2004, *Electronic business eXtensible Markup Language (ebXML) – Part 2: Message service specification (ebMS)*

ISO/TS 15000-3:2004, *Electronic business eXtensible Markup Language (ebXML) – Part 3: Registry information model specification (ebRIM)*

ISO/TS 15000-4:2004, *Electronic business eXtensible Markup Language (ebXML) – Part 4: Registry services specification (ebRS)*

ANSI ASC X12, Release 4040, December 2000

UN/EDIFACT, D.01A Directory, January 2001

UN/CEFACT *Modelling Methodology (UMM)*, NO90 R12 or higher

UN/CEFACT *ebXML Technical Architecture Specification*, v1.04 or higher

UN/CEFACT *ebXML Business Process Specification Schema*, v1.10 or higher

In this part of IEC 62325, RFCs (Request for comments) from the Internet Engineering Task Force (IETF) and recommendations from other Organisations such as the Word Wide Web Consortium (W3C) and the Organization for the Advancement of Structured Information Standards (OASIS) are mentioned which are not included here because these documents are referenced in the references above.

3 Terms, definitions and abbreviations

3.1 Terms and definitions

None.

3.2 Abbreviations

A2A	Application-to-Application
AES	Advanced Encryption Standard
B2B	Business-to-Business
BDS	Business Document Specification (instance)
BDSS	Business Document Specification Schema
BIE	Business Information Entity
BOV	Business Operational View
BPMS	Business Process Management System
BPSS	Business Process Specification Schema (or instance)
BSI	Business Service Interface
CC	Core Component (based on BIE)
CIM	Common Information Model
CPA	Collaboration Protocol Agreement
CPP	Collaboration Protocol Profile
DSO	Distribution System Operator (of power system
DUNS	Data Universal Numbering System (North America)
EAN	European Article Number (Europe)
ebMS	ebXML Messaging Service
ebXML	electronic business XML
EDI	Electronic Data Exchange
EIA	Enterprise Application Integration
EMS	Energy Management Systems
ERP	Enterprise Resource Planning
FOV	Functional Service View
FTP	File Transfer Protocol
HTTP	Hypertext Transport Protocol
ICT	Information and Communication Technology
ISO	Independent System Operator
IT	Information Technology
MIME	Secure/Multipurpose Internet Mail Extensions
MIS	Market Identification Schema
MOM	Message-oriented middleware
MSH	Message Service Handler
PKI	Public Key Infrastructure
QoS	Quality of Service
RPC	Remote Procedure Call
RR	Registry/Repository

SAML	Security Assertion Mark-up Language
SCADA	Supervision, Control, and Data Acquisition
SMTP	Simple Mail Transfer Protocol
SO	System Operator (of power system)
SOAP	Simple Object Access Protocol
TLS	Transport Layer Security
TSO	Transmission System Operator (of power system)
UML	Unified Modelling Language
UMM	UN/CEFACT Modelling Methodology
VPN	Virtual Private Network
WS	Web Services
WSDL	Web Services Definition Language
XML	eXtensible Markup Language
XKMS	XML Key Management Specification

4 Generic technical architecture

4.1 General

The following text is mainly based on the public description of the ebXML initiative (http://www.ebxml.org/) and is intended to provide a basic understanding of the technology. For details, refer to the ebXML implementation framework specification and the ebXML architecture document of the initiative.

4.2 Architecture

4.2.1 General

The vision of ebXML is to create a single global electronic marketplace where enterprises of any size and in any geographical location can meet and conduct business with each other through the exchange of XML based messages. ebXML is a complete set of specifications to enable secure, global, electronic business using proven, open standards such as TCP/IP, HTTP, SOAP, and XML. ebXML is also evolutionary in nature, built on 25 years of EDI experience, designed to work with existing EDI solutions, or be used to develop an emerging class of internet based electronic business applications based on XML.

Since systems integration and software interoperability are the cornerstones of any successful IT infrastructure, ebXML is built on an infrastructure that ensures electronic interoperability. This is accomplished by providing an open semantics framework that allows enterprises to find each other, agree to become trading partners, and conduct business. The evolution of many new business models will be enabled by ebXML, through business process patterns and the 'commoditization' of such business processes.

The electronic business infrastructure provided by ebXML is broad in scope and well integrated. And perhaps most importantly, ebXML is platform and vendor neutral, providing an industry solution based on open standards, designed through a collaborative and open process.

ebXML is a set of specifications that together enable a modular, yet complete electronic business framework for using the Internet. The ebXML architecture provides:

- A way to define business processes and their associated messages and content.
- A way to register and discover business process sequences with related message exchanges.

- A way to define company profiles.
- A way to define trading partner agreements.
- A uniform message transport layer.

The ebXML framework is designed for electronic interoperability, allowing businesses to find each other, agree to become trading partners and conduct business. All of these operations can be performed automatically, minimising, and in most cases completely eliminating the need for human intervention. This streamlines electronic business through a low cost, open, standard mechanism.

In order for enterprises to conduct electronic business with each other, they should:

- Discover each other and the products and services they have to offer.
- Determine which shared business processes, and associated document exchanges, to use for obtaining products or services from each other.
- Determine the contact points and form of communication for the exchange of information.
- Agree on the contractual terms on the above chosen processes and associated information.
- They can then: exchange information and services in an automated fashion in accordance with these agreements.

ebXML is designed to meet these needs and is built on three basic concepts: provide an infrastructure that ensures data communication interoperability; provide a semantics framework that ensures commercial interoperability; and provide a mechanism that allows enterprises to find each other, agree to become trading partners and conduct business with each other. The infrastructure to ensure data communication interoperability is provided through:

- A standard message transport mechanism with a well defined interface, packaging rules, and a predictable delivery and security model.
- A 'business service interface' that handles incoming and outgoing messages at either end of the transport.
- A semantic framework to ensure commercial interoperability is provided through a meta model for defining business process and information models.
- A set of re-useable business logic based on core components that reflect common business processes and XML vocabularies.
- A process for defining actual message structures and definitions as they relate to the activities in the Business Process model.
- A mechanism to allow enterprises to find each other, agree to establish business relationships, and conduct business, is provided through shared repository where enterprises can register and discover each other's business services via partner profile information.
- A process for defining and agreeing to a formal Collaboration Protocol Agreement (CPA), if so desired or where required.
- A shared repository for company profiles, business process models and related message structures.

The ebXML implementation framework defines the ebXML Technical Architecture. The technical architecture is composed of five main area of emphasis: Business Process and Information Model, Collaboration Protocol Profiles Company Profiles, Messaging Services, Registry and Repository, Collaborative Partner Agreements.

4.2.2 Business process description

The Business Process models define how business processes are described. Business Processes represent the "verbs" of electronic business and can be represented using modelling tools. The specification for business process definition enables an organisation to express its business processes so that they are understandable to other organisations. This enables the integration of business processes within a company, or between companies.

Figure 1 shows the graphical presentation of the BPSS (Business Process Specification Schema) process specification to provide a basic understanding of the technology. The main elements are multiparty collaborations and binary collaborations. Both include (reference) business transactions, which govern the business document flow.

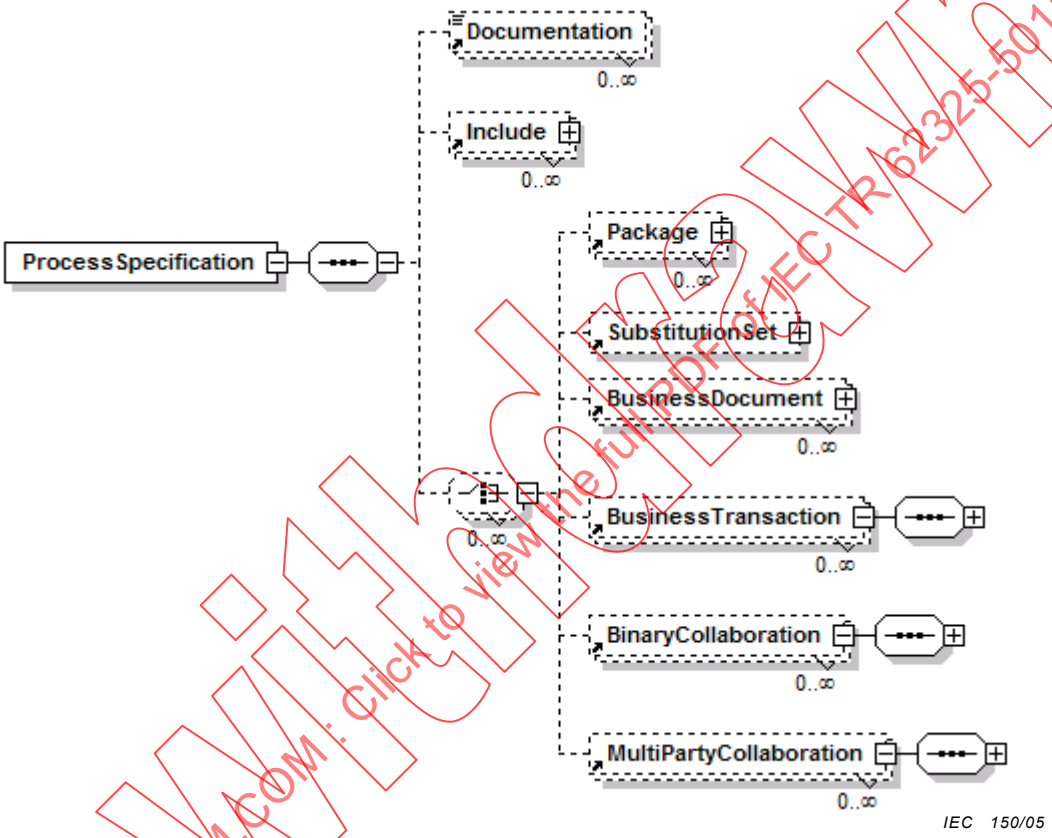


Figure 1 – Process specification

Figure 2 shows the graphical presentation of the business transaction from the Business Process Specification Schema (BPSS).

The business transaction consists of a requesting business activity and a responding business activity each associated with a document envelope (which includes the business documents and attachments).

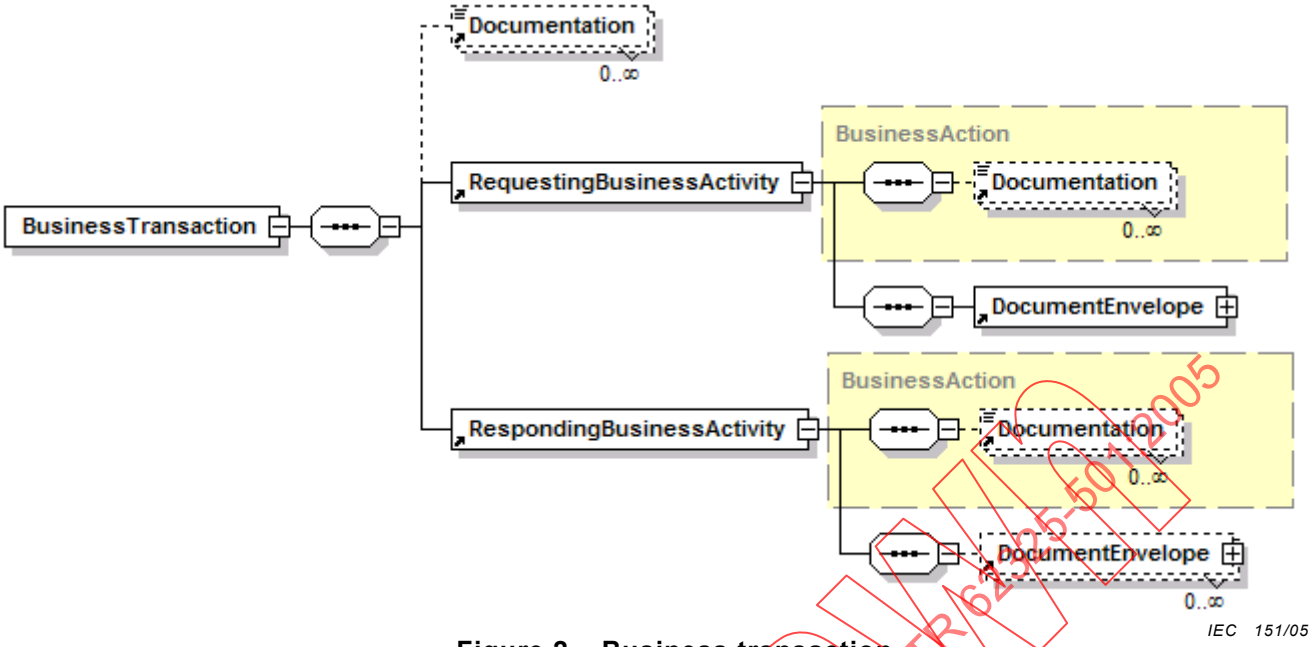


Figure 2 – Business transaction

Figure 3 shows the graphical presentation of the binary collaboration from the Business Process Specification Schema (BPSS). Specific business process specifications are derived from the UML activity diagram (see IEC 62325-502). The Business Process Specification Instance uses the BPSS and is, besides the CPP/CPA, the most innovative part of ebXML. BPSS, CPP/CPP are shared between business partners and are used for both documentation and system configuration.

The binary collaboration is between two business partners with one partner in the initiating role and the other partner in the responding role. It includes business transaction activities and (nested) collaboration activities. These activities reference business transactions respective collaborations. Collaborations have a start state (Start) and a completion state (Success or Failure). From start to end there can be pseudo-states fork (Fork) and join (Join) for parallel transactions. The transition state specifies that a given activity be followed by another activity.

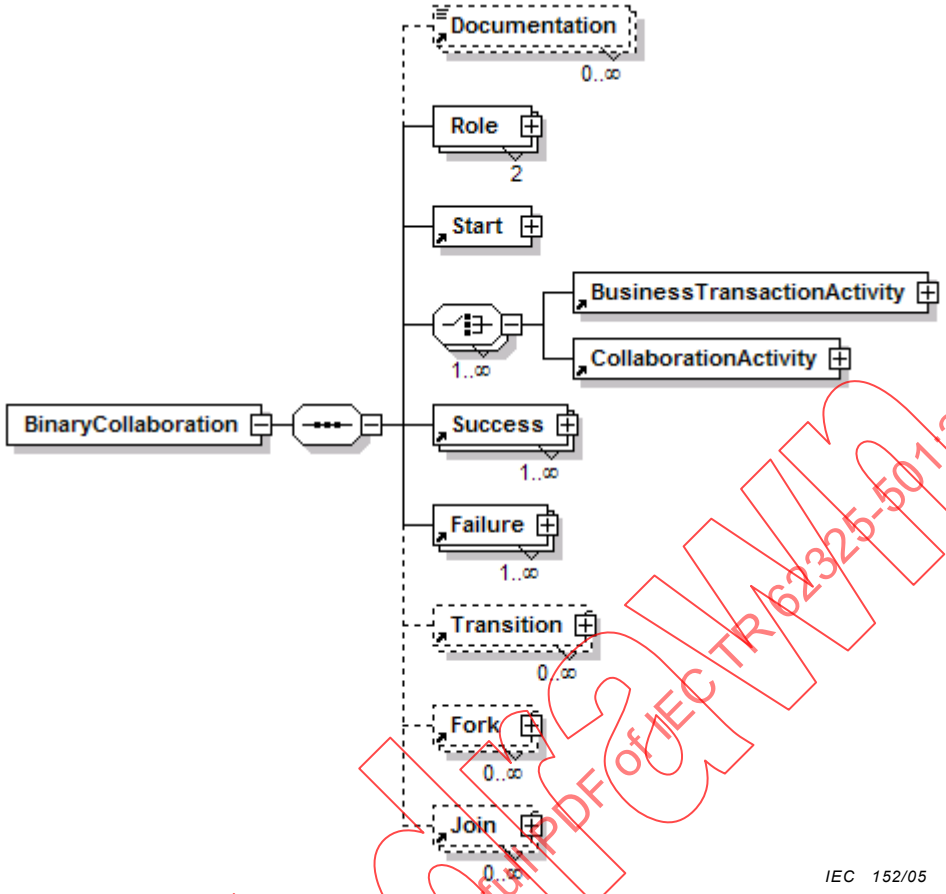


Figure 3 – Binary collaboration

Figure 4 shows the graphical presentation of the multiparty collaboration from the Business Process Specification Schema (BPSS). The multiparty collaboration is between more that two partners and consists of multiple binary collaborations. It is defined by the business partner's role, which performs (Performs) one or more roles in various binary collaborations. Transitions (Transition) can be added to define the transition from one business transaction activity to the other.

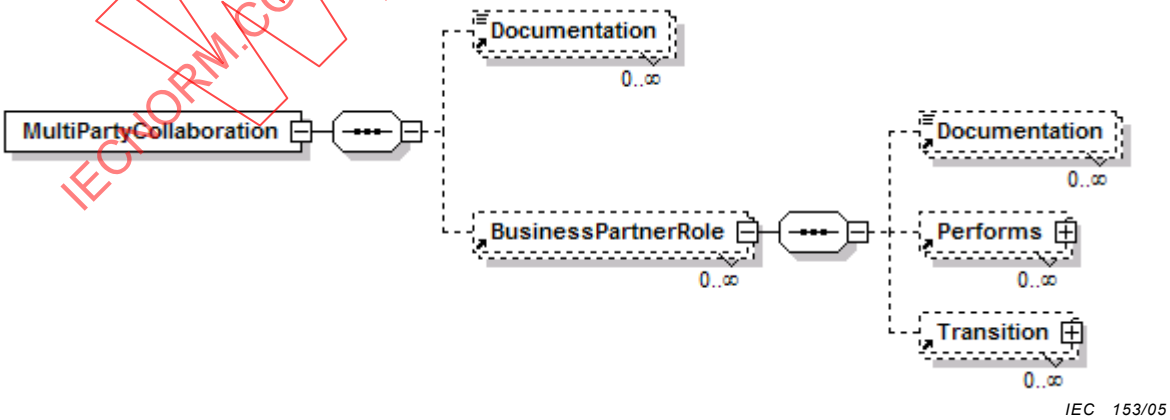
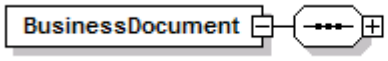


Figure 4 – Multiparty collaboration

4.2.3 Business documents

The Information models define reusable components that can be applied in a standard way within a business context. These Core Components represent the "nouns and adjectives" of electronic business. They are defined using identity items that are common across all businesses. This enables users to define data that is meaningful to their business while also maintaining interoperability with other business applications.

Figure 5 shows the graphical presentation of the business document from the Business Process Specification Schema (BPSS). Business documents are derived in the UMM design workflow from UML class diagrams of messages and incorporate information entities (re-usable core components). Business documents can be of any syntax (also EDIFACT or X12) but XML is the preferred syntax within ebXML.



IEC 154/05

Figure 5 – Business document

4.2.4 Business agreement

The Collaborative Partner Profiles (CPP) and Collaborative Partner Agreements (CPA, build from the intersection of two CPP) captures critical information for communications between applications and business processes and also records specific technical parameters for conducting electronic business. IEC 62325-502 specifies the profile.

Figure 6 shows the graphical presentation of the collaboration protocol profile from the Collaboration Protocol Profile (CPP) Schema. For collaboration role, delivery channel, transport, and document exchange, see the following Figures 7 to 10. Party Identification (PartyId) identifies and party reference (PartyRef) references the business partners.

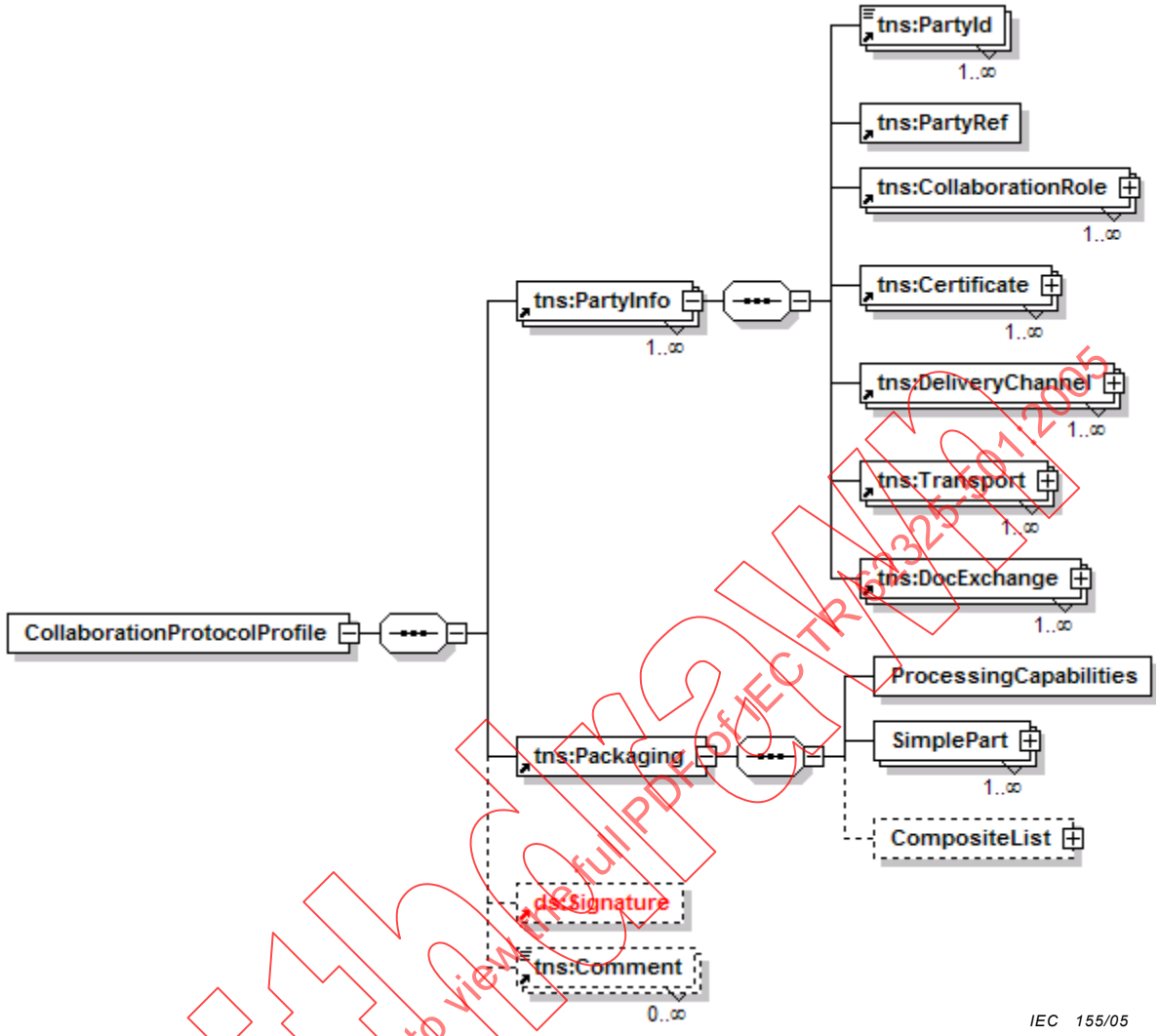


Figure 6 – Collaboration protocol profile

Figure 7 shows the graphical presentation of the collaboration role from the Collaboration Protocol Profile (CPP) Schema. It defines the collaboration role (Role) and includes a reference to the BPSS (ProcessSpecification)

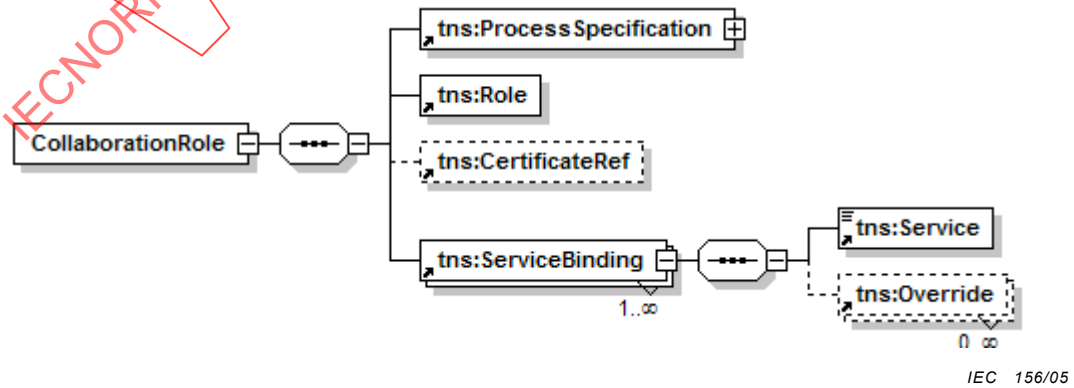


Figure 7 – Collaboration rule

Figure 8 shows the graphical presentation of the delivery channel from the Collaboration Protocol Profile Schema (CPP). It defines its characteristics with regard to reliability, non-repudiation and security.

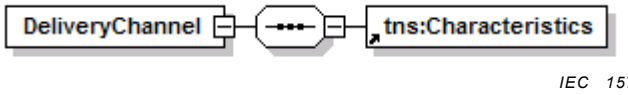


Figure 8 – Delivery channel

Figure 9 shows the graphical presentation of the transport from the Collaboration Protocol Profile (CPP) Schema. It defines the sending and receiving protocol (mostly the same) and its endpoints. Transport security is an option.

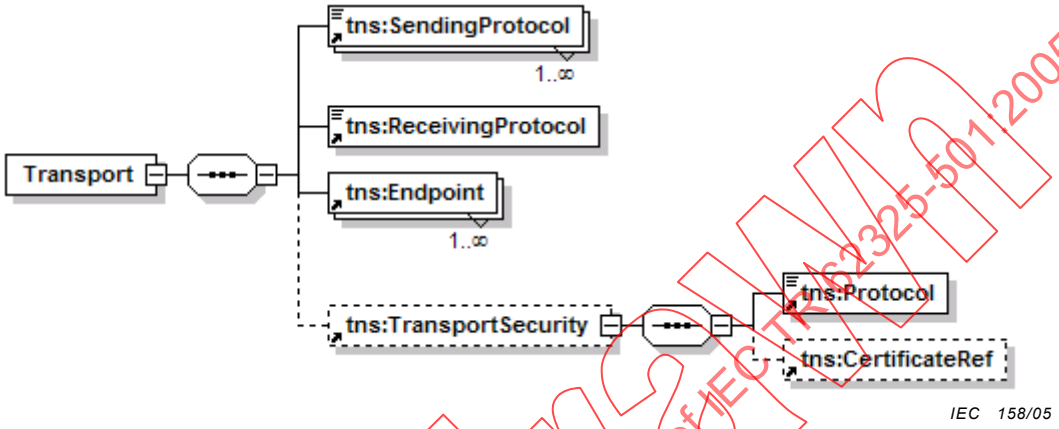


Figure 9 – Transport

Figure 10 shows the graphical presentation of the document exchange from the Collaboration Protocol Profile (CPP) Schema. It defines, with ebXML binding (see Figure 11), the reliability and security features and parameters of the ebXML messaging.

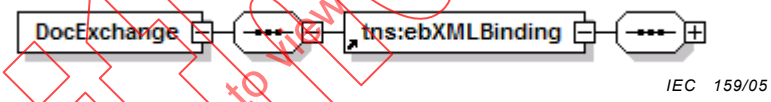


Figure 10 – Document exchange

Figure 11 shows the graphical presentation of the ebXML binding from the Collaboration Protocol Profile Schema (CPP). It defines reliable messaging (retry, retry interval, duration of persistent storage), non-repudiation (signature, etc.), and the digital envelope (encryption, etc.) including the associated certificates.

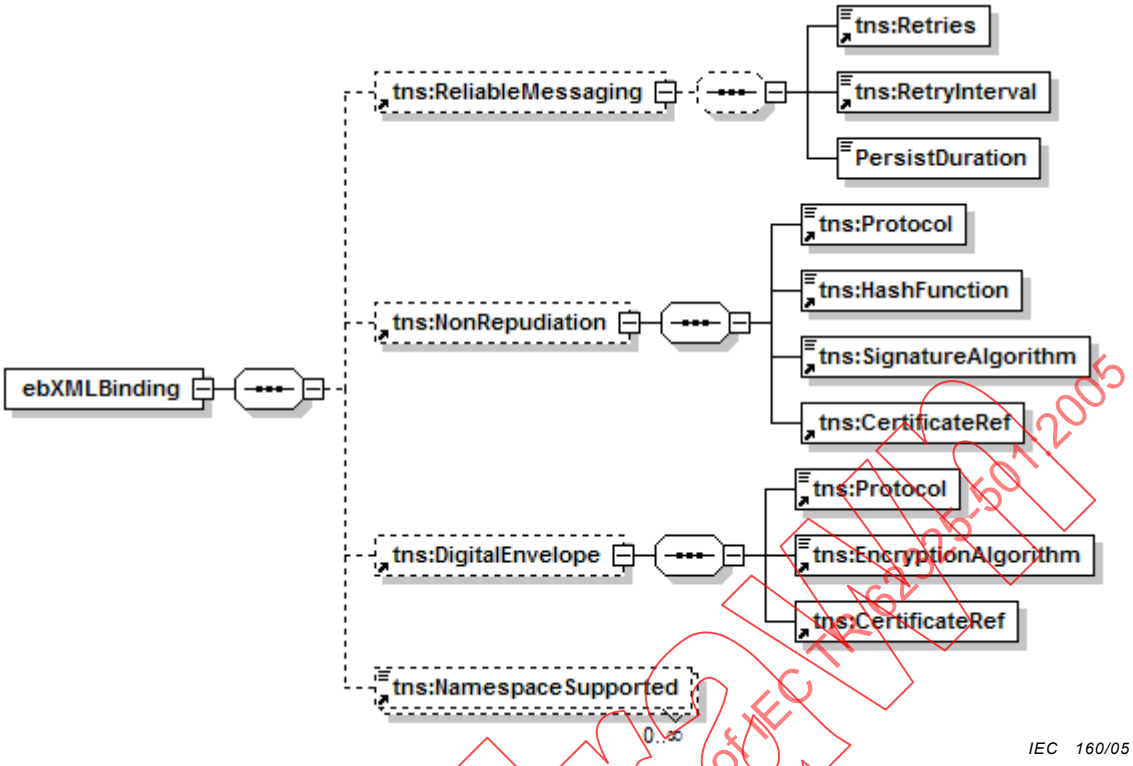


Figure 11 – ebXML binding of reliability and security

4.2.5 ebXML message

4.2.5.1 SOAP envelope header and body

The ebXML message consists of the envelope, header, and body wrapped into the SOAP with the Attachment container. A *signed* message is any message containing a Signature element.

Figure 12 shows the soap envelope header and body.

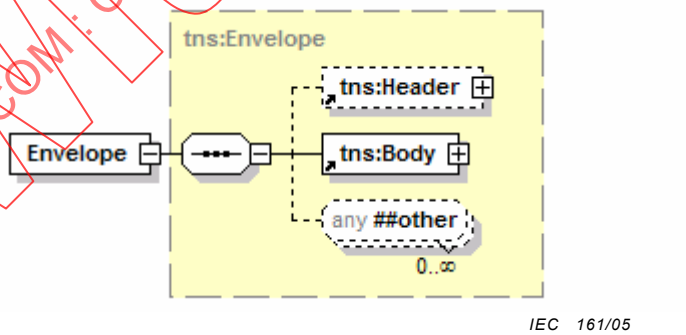


Figure 12 – SOAP envelope header and body

An ebMS message package consists of one or more MIME parts. The first MIME part contains the SOAP envelope. The SOAP envelope header contains, besides the MessageHeader element (see below) and other elements, the elements Signature and Acknowledgement. The element Signature conforms to the XML DSIG specification and includes, besides the methods and algorithms and the reference to the payload, the signature value (that is the signed message digest for authentication) and the message digest value. The public key can be optionally exchanged if not done by other means. It is configured by the CPA for all BPSS profiles which require *persistent* digital signatures.

The element Acknowledgement should be used in all BPSS profiles for the acknowledgement of messages on messaging level and to allow retry.

The SOAP envelope body contains, besides other elements, the element deliveryReceipt, which is used in the relevant BPSS profile (see IEC 62325-502) for persistent non-repudiation with signed receipt on application level. The Manifest element makes reference to the message payload and identifies some resource that describes the payload object or its purpose.

4.2.5.2 Message header and payload

Within the SOAP envelope, the SOAP Header element contains the MessageHeader element. The actual payload (XML documents or others, e.g. EDIFACT or X12 documents) is contained in the second (and maybe also in the following) MIME parts. Figure 13 shows the message header.

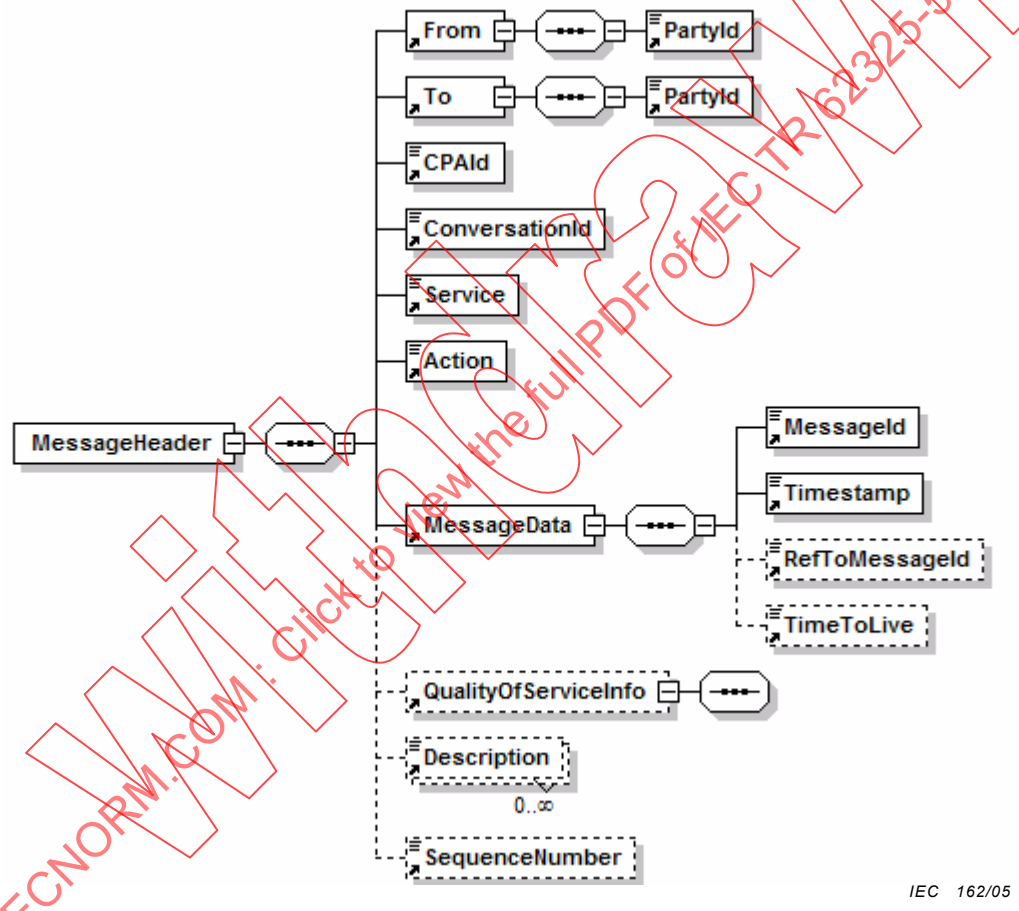


Figure 13 – Message header

The specific use of some elements in energy markets should be as follows (see also the BPSS profile, and the CPA profile in IEC 62325-502):

The required From/PartyID and To/PartyID elements contain the market participant identifier (e.g. market specific or EAN, DUNS) according to the Market Identification Schema (see IEC 62325-101, Subclause 6.4). The type attribute indicates the domain of the PartyID to differentiate, e.g. between more than one market identification schemas in use or to reference other registries. If no type is specified, the PartyID element should be an URI. It is recommended to specify the type (e.g. type value "DUNS", "EIC", or "EAN" according to the Market Identification Schema).

The required CPAId element references the CPA used in order to determine the reliability parameters for the message exchange. These parameters of the optional (but in this case, required) QualityOfServiceInfo element are configured with the CPA profile (see IEC 62325-502). The deliverySemantics attribute of the CPA is identical with the QualityOfServiceInfo element. The idempotency attribute of the CPA determines if the MSH checks for duplicated messages and processes it. The messageOrderSemantics attribute of the CPA is identical with the QualityOf ServiceInfo element's attribute of same name. The retry attribute of the CPA determines the maximum number of retries of an unacknowledged message. The RetryInterval element contains a time value for the duration that a sender should wait between retries. The PersistDuration element of the CPA determines the duration a message is kept persistently in store by the receiving MSH. The deliveryReceiptRequested attribute of the QualityOf ServiceInfo element is determined by the profile of the BPSS (see BPSS profile in IEC 62325-502) referenced by the CPA.

The required Service element defines the service that acts on the message and is specified in a project or market. The service is further specified by the type attribute. The required Action element identifies the associated process activity within the service conforming to the BPSS. The acknowledgement service should be used for reliable message transfer in all CPA profiles.

The optional SequenceNumber element defines the sequence number of messages and should be present, because in all CPA profiles, the messageOrderSemantics attribute of the QualityOfServiceInfo element is set to the value "Guaranteed".

The optional Description element can have a number of child elements, which can be optionally used to describe the use of the message. For example, for a human readable description of the message content, the supported process can be described.

4.2.6 Mapping of the UMM model to ebXML

The mapping of the UMM (UN/CEFACT Modelling Methodology) business process model to ebXML is straightforward. Table 1 shows the relation of UMM with ebXML.

Table 1 – Relation of UMM with ebXML

UMM workflow	ebXML component	Elements	Links
Business modelling	BPSS	MultiPartyCollaboration	BusinessPartnerRole Transition
Requirements		BinaryCollaboration	BusinessTransactionActivity Fork, Join
Analysis		BusinessTransaction	
Design	CPP/CPA	CollaborationRole	Role

The business process modelling goes over three workflows defining multiparty collaborations, binary collaborations and business transactions. Its result is mapped to the Business Process Specification (BPSS instance) based on the BPSS. The result of the design (business service view) workflow is mapped to the Collaboration Protocol Profile (CPP). In the implementation workflow, the negotiated CPP, called Collaboration Protocol Agreement (CPA) is used for both documentation and configuration of the Business Service Interface (BSI).

4.2.7 Registry and repository

The Registry and Repository (RR) provides a number of key functions. For the user (application), it stores company profiles and trading partner specifications. These give access to specific business processes and information models to allow updates and additions over time. For the application developer, it will store not only the final business process definitions, but also a library of core components. It allows to describe and search business partners, and to update and retrieval the ebXML key artefacts BPSS, CPP, CPA besides UML models and other textual descriptions.

4.2.8 Messaging services

The ebXML Messaging Service specification defines the set of services and protocols that enables electronic business applications to exchange data. The specification allows any application-level protocol to be used. These can include common protocols such as SMTP, HTTP, and FTP, but at the moment, only HTTP and SMTP are included.

Well-established cryptographic techniques can be used to implement strong security. XML Encryption is the preferred default encryption to guarantee persistent confidentiality. Other secure protocols such as TLS or IPsec are optional. In addition, persistent digital signatures can be applied with XML Digital Signature to individual messages or a group of related messages to guarantee authenticity.

For profiles, see IEC 62325-502.

4.2.9 Transport of messages

The transport is over TCP/IP.

4.2.10 Configuration of agreed messaging services

The business service interface (BSI) provides the means to communicate with business partners and is configured with the CPA.

4.2.11 Abstraction, binding and mapping

Figure 14 shows the ebXML configuration, abstraction, binding and mapping. The ebXML Messaging Service (ebMS) is a service logically positioned between one or more business applications and a communication service (transport protocols). This requires the definition of an *abstract business service interface* of the BSI (Business Service Interface) between the business applications and the Message Service Handler (MSH) to hide the ebMS technology from the application. ebXML does not yet define this abstract business service interface, but it may be included in future versions of the ebMS with the following functions:

- Send () – send an ebXML Message (values for the parameters are derived from the ebXML message headers).
- Receive () – indicates willingness to receive an ebXML message.
- Notify () – provides notification of expected and unexpected events.
- Inquire () – provides a method of querying the status of the particular ebXML message interchange.

The ebXML architecture uses the ebMS, but is not restricted to it. The abstract service interface (to be specified) and the descriptive nature of ebXML (with BPSS, CPA) will allow alternative reliable and secure messaging services in future, which may come for example from W3C.

Bindings and mappings to two communication protocols (HTTP and SMTP over TCP/IP) are defined. However, the MSH is specified independent of any communication protocols. While HTTP is the preferred solution, no preference is being provided to this protocol. Other protocols may be used in future versions of the ebMS.

The ebMS relies on *external* configuration information. The CPA referencing the BPSS provides this information. But this is not a requirement. If the CPA and BPSS are not used, the ebMS should be configured manually.

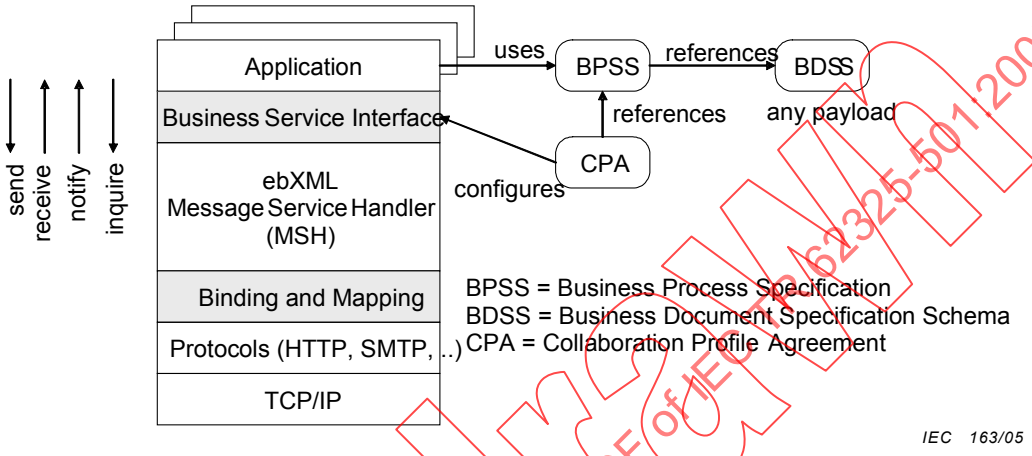


Figure 14 – ebXML configuration interfaces

The relative independence of the various ebXML specifications such as ebMS, BPSS, CPP/A and the fact that the message payload can be of any format (but XML is preferred) ensures their interoperability with alternative or future implementations, flexibility in choosing software components to implement ebXML systems and protection of investment in ebXML solutions.

5 Comparison with other solutions

The ebXML e-business technology goes far beyond EDIFACT and X12: EDIFACT and X12 are message-centric and do not define transactions. Trading partner agreements (TPA) and implementation guides are necessary to implement systems. With ebXML, these agreements and guides become machine-readable XML configuration files for the deployment of systems.

EDIFACT and X12 use fixed messages that do not fit all business cases that require certain variations in the messages. With ebXML, flexible messages can be generated and extended by Business Information Entities (BIE's) based on re-usable core components. However, it is not the message that is the first priority of standardization, but rather the information entities based on core components (vocabulary) from which the messages are built. Core components are context-driven (industry, geopolitical, business process, product, official constraints) to give them a specific meaning in various business processes.

In the future, business processes could be based on core business processes in the same way that messages are based on core components.

Web Services from W3C and OASIS may be an alternative to ebXML in the future, if the current simple OO (object oriented) RPC services possible with UDDI registry, WSDL service definition, and SOAP messaging and encoding are enhanced (e.g. reliable and secure messaging with literal/document encoding using XM schemas, choreography of services). This will be covered by the future IEC 62325-6XX.

6 ebXML configuration

6.1 General

Before any electronic business document interchange using ebXML, a BPSS (Business Process Specification Schema) as a UML class diagram, the XML version derived from it and a BPSS which is the XML version of a BPSS instance have to be produced. For each complete process a CPP is then created, which is again a XML document. In the end, when two market participants agree to engage in e-business interactions, they create a CPA. The CPA specifies how the market participants are going to work together on a business collaboration level and how the messages will be transported (selected features of reliable and secure messaging) using the ebXML messaging service.

6.2 Business Process Specification

The ebXML Business Process Specification is a XML document (BPSS instance) that conforms to the BPSS Schema and that is used to configure the Business Service Interface (BSI).

Since the ebXML Business Process Specification Schema (in UML or XML) is a semantic subset of the UMM meta model, the user may extract in an automated or manual fashion from the UMM model the required set of elements and relationships, and transform them into an ebXML Business Process Specification conforming to the ebXML Business Process Specification Schema. This can be a Business Process Specification in UML that is then transformed into a XML format or the Business Process Specification is directly created in XML from the UMM model. However, the final result should be the ebXML Business Process Specification in XML that conforms to the BPSS and is a BPSS instance.

If the XML Business Process Specification is derived from the corresponding UML class diagram, all the classes, except the abstract ones, become elements of the XML Business Process Specification (see Table 2). For this transformation, production rules should be followed. The primary purpose of these production rules is to govern the one-time generation of the BPSS instance from the UML class diagram version.

The production rules from UML to XML are defined for concrete classes, abstract classes, aggregate associations, specialisation associations and unidirectional associations. For this, see the ebXML Business Process Specification Schema specification.

Annex A shows the DTD and the XML files used for the ebXML scheduling example.

Table 2 – Mapping between UML classes and XML elements

XML element	UML class
Attachment	Attachment
InitiatingRole	AuthorizedRole
RespondingRole	AuthorizedRole
Binary Collaboration	Binary Collaboration
BusinessPartner Role	BusinessPartner Role
Business Transaction Activity	Business Transaction Activity
Business Transaction	Business Transaction
Responding BusinessActivity	Responding BusinessActivity
Requesting BusinessActivity	Requesting BusinessActivity
Collaboration Activity	Collaboration Activity
DocumentEnvelope	DocumentEnvelope
Documentation	None (Should be added)
ebXML Process Specification	(From Package model: ebXML Process Specification)
Failure	Failure
Include	(From Package model: Include)
MultiParty Collaboration	MultiParty Collaboration
Package	(From Package model: Package)
Performs	Performs
Schema	Schema
Fork	Fork
Start	Start
Success	Success
Join	Join
Transition	Transition

6.3 Collaboration Protocol Profile and Agreement (CPA/CPP)

The two CPP in the scheduling example (one for the Balance-Responsible Party (BRP) and one for the Transmission System Operator (TSO)) are done by each one of the actor's role. These CPP are published in the registry. It is assumed that another party maintains this registry. The TSO's CPP describes the business capabilities of the TSO, the role it plays, the services it offers and the technical details on how those services can be accessed. The BRP then browses the registry (in the right table) and retrieves the TSO's CPP and decides to engage in an e-business partnership with this TSO.

The BRP then unilaterally creates a CPA using its own and the one of the TSO. The BRP proposes this CPA to the TSO. When the TSO accepts this CPA, the e-business relationship is established. Once the TSO accepts the CPA, the parties may begin to conduct B2B transactions as configured in the MSH (Message Service Handler).

6.4 Mapping to the UN/CEFACT Modelling Mythology UMM

UMM has strongly influenced the design of the ebXML Business Process Specification Schema (BPSS) and therefore the mapping from UMM to BPSS is straightforward. For modelling purposes, the four main workflows of UMM are used, which are: (1) Business Modelling, (2) Requirements, (3) Analysis and (4) Design. Each workflow can go through the four phases: Interception, Elaboration, Construction, and Transition. The third dimension is Iteration, which is not defined by UMM.

An alternative of UMM is the ebXML worksheets that are based on templates to capture the modelling results in a predefined table structure. This alternative may be of more interest if specialised e-business editors with GUIs are available. It is also possible to use UML tools with stereotyped business patterns.

6.5 Use of registries

The ebXML Registry/Repository (RR) should be used to enable market participants to facilitate the establishment of ebXML data exchange. It is used to describe and discover market participants, and to store and retrieve relevant e-business artefacts for configuration and description. It is recommended that the RR be operated within a market or market region by only one organisation in consensus with the market participants. This RR operator can be totally independent from the parties exchanging ebXML documents. It is further recommended that its content and structure be within an energy market harmonised over all market segments (in the UMM sense business areas). This will afford the co-operation of all market participants.

Besides ebXML artefacts and UMM market models, the Market Identification Schema (the ID's) and other useful items could also be published in the RR.

The configuration of ebXML systems may be simplified if the RR stores besides CPP's "pre-negotiated" default CPA templates for supported core business processes.

For the RR, the same specified BPSS reliability and security profiles apply as for B2B.

6.6 Use of intermediates

EbXML is based on SOAP with attachments capable of routing messages through so called SOAP intermediates which receive a message and forward it towards its final destination over a multi-hop message path. This is an important concept which allows SOAP intermediates to do some useful processing on the messages taking some burden from the original sender or receiver. For example, SOAP intermediates may verify, encrypt/decrypt and sign/authenticate messages. One possibility would be that messages are signed on application level to allow non-repudiation, but message encryption and system authentication takes place in the DMZ (demilitarised zone) using an intermediate. An intermediate could also act as a third party provider and do all the message processing and then forward the message to the final party. There are many possibilities and no general recommendation about the use of intermediates ca be given.

The referenced version of the ebXML messaging service does not define the use of intermediates possible with SOAP. Its use is therefore proprietary and project specific.

7 Profile of the architecture

Within the ebXML specification framework, two business partners agree on how to perform e-business using machine-readable Trading Partner Agreements based on XML syntax and named Collaboration Profile Agreements (CPA). In the general case of global e-business, the CPA is negotiated as the intersection of the Collaboration Protocol Profiles (CPP) of these two partners, who may have discovered each other using the registry partner-discovering feature. Energy markets normally exist in a specific geographical area or geopolitical region with known business partners, agreed market rules and communication infrastructure. In this environment, a simplification may be possible where alternatively pre-negotiated CPA's of each business process are stored pre-defined in the registry/repository and can be downloaded for use.

Within each market, a profile or a limited set of profiles of the ebXML architecture should be used to harmonise and simplify e-business. Since the ebXML specification framework does not define any market specific profiles, the profile for energy markets is defined in IEC 62325-502.

It is recommended that ebXML systems, which use digital signatures with certificates, a different certificate is used for test purposes to differentiate between production and test. For simple tests, the ebMS ping service can be used.

8 Migration scenarios

8.1 General

Figure 15 conceptually shows the migration from EDIFACT (X12) to ebXML and the data networks that can be used. In all scenarios, a choice can be made to use VAN's (Value-Added Networks) to provide secure and reliable messaging, Extranets (own network for market participants under control of the trusted market participants) or the insecure public Internet. Because the ebXML messaging service provides reliable and secure messaging, there is normally no need for a VAN or Extranet and the Internet can be used as is. That does not exclude leased lines or Extranets for specific market segments (e.g. trading).

It is important to note that EDIFACT (X12) is not restricted to VAN's and can, besides Extranets also use the public Internet (see in Figure 15, (1)) if reliable and secure messaging is used. The same is true for XML-messages (see in Figure 15, (2)), if ebXML is not used. For an environment with no planned migration to ebXML, the IETF Internet Draft (draft-ietf-ediint-as2-xx ("AS2")) can be used, which is still a work in progress (no RFC). AS2 describes how to exchange structured business data securely using HTTP transport for EDIFACT, X12, XML or other formats used for business-to-business data interchange. The data is packaged using standard MIME content-types. Authentication and privacy are obtained by using (preferably) S/MIME or OpenPGP security body parts. Authenticated acknowledgements make use of multipart/signed replies to the HTTP POST requests.

The various possible scenarios are described in Figure 15.

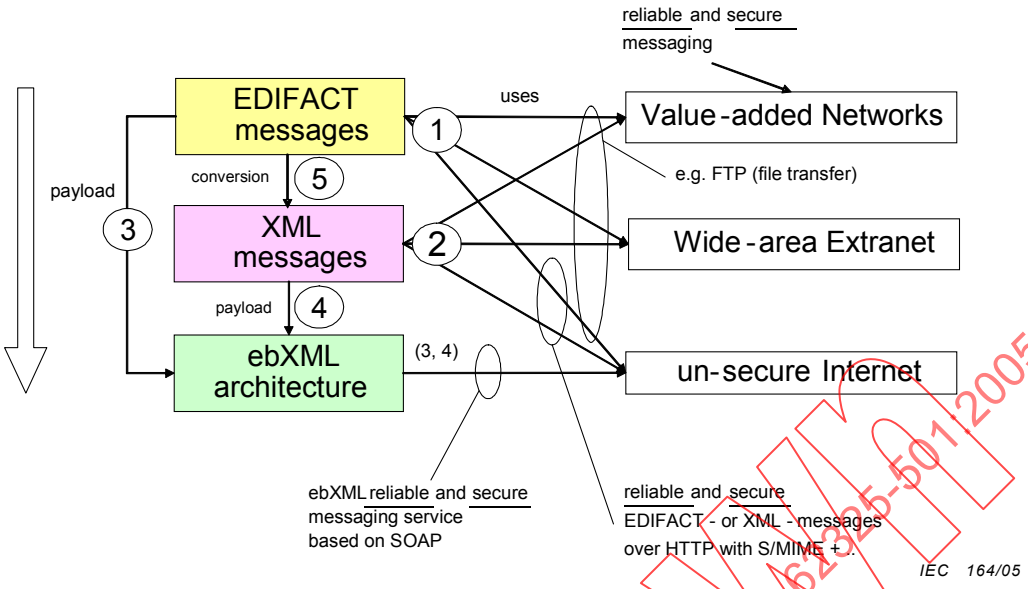


Figure 15 – Migration from EDIFACT to ebXML and use of data networks

The scenario (5) with conversion of EDIFACT messages to XML is not recommended. Instead (see Figure 16) EDIFACT messages should be used uncharged as payload of the ebXML messaging service (3).

8.2 Recommended scenarios

It is not recommended to use all possible scenarios in an ebXML environment and to use the recommended scenarios shown in Figure 16. Either existing EDIFACT messages or new XML messages based on core components are used as payload. Both are possible in parallel.

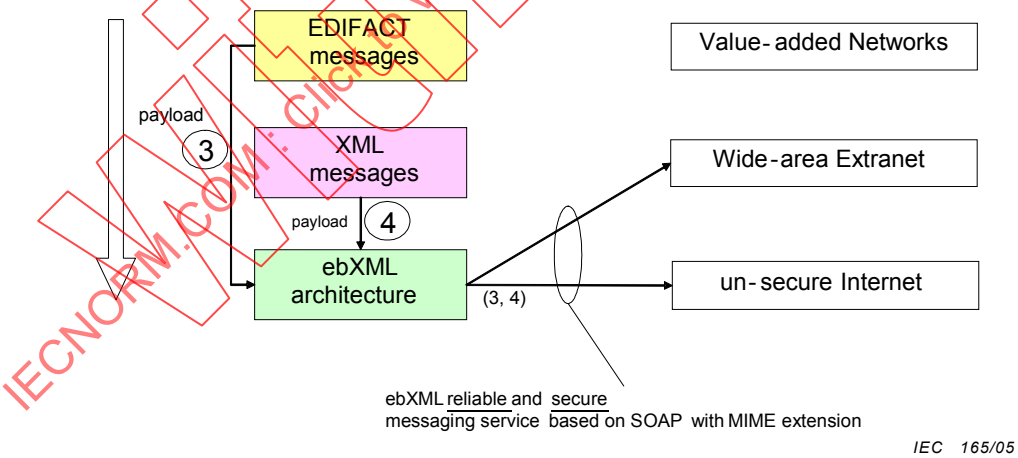


Figure 16 – Migration from EDIFACT to ebXML

8.3 ebXML and EDIFACT (X12) in parallel

In this scenario, an EDIFACT (X12) infrastructure exists, but additionally ebXML communication should be possible with some market participants. The obvious solution is to create, within a system, a new ebXML communication interface in parallel to the EDIFACT (X12) communication interface (see in Figure 15, (1) and (4)). This solution requires that the application maps to both communication interfaces.

8.4 ebXML with EDIFACT (X12) payload

In this scenario, an EDIFACT (X12) infrastructure exists, but ebXML communication should additionally be possible with market participants also using the reliable and secure ebXML messaging service for EDIFACT (X12) messages (documents). The ebXML messaging services based on SOAP with extensions and attachments, uses within the MIME envelope multiple MIME parts for the SOAP envelope (with header and body), and the payload container(s). The EDIFACT (X12) messages are carried as payload in these containers (see in Figure 15, (3) and(4)).

IETF RFC 1767, MIME Encapsulation of EDI Objects, provides a description of the applicable content types for EDI. The encoding scheme Base64 ensures the encoding of non-printable EDI delimiters.

8.5 EDIFACT (X12)/XML conversion

Conversion based on the XEDI or XML/EDI or other approaches to XML is not recommended. New XML messages should be designed from the begining based on an UMM market model and standardized core components. A design methodology and rules are necessary to archive a harmonised approach over all market segments. For XML messages, basic work is in progress at UN/CEFACT.

Annex A
(informative)

Scheduling example

A.1 General

The following example for scheduling with ebXML is based on the technology-independent modelling for the ETSO Scheduling System ESS (www.edi.etso-net.org/) and shows BPSS, CPP, CPA and BD instance examples. The example is not part of the ESS system.

A.2 Scheduling model

The example follows the UML model of scheduling given in IEC 62325-102.

In the following example, a BPSS XML file is created for the Schedule Planning Transmission Process. This BPSS can be stored in the Registry/Repository.

In this process (see Figure A.1), the Balance Responsible Party sends his electricity generation schedule in his balance area to his Transmission System Operator. The Transmission System Operator then has to validate this data and send the validated data to the Balance Responsible Party and to the Settlement Responsible Party.

The three phases of the schedule planning transmission process are implemented. Some points have been modified or are not done in an automated way:

1) **Phase one:** the question "Schedule Message correct" is answered in an automated way, so that all the errors are taken into account,
2) **Phase two:** the question "time series in balance" is answered by the user by "yes" or "no". If the answer is "no" then the model ends and the user has to begin again. If the answer is "yes", then phase 3 starts,
3) **Phase three:** a reference schedule file (which is arbitrary) has been made and the transmitted schedule for the example is then compared to this reference file. The confirmation report is sent in an automated way with or without differences. The accepted schedule is sent to the Imbalance Settlement Responsible.

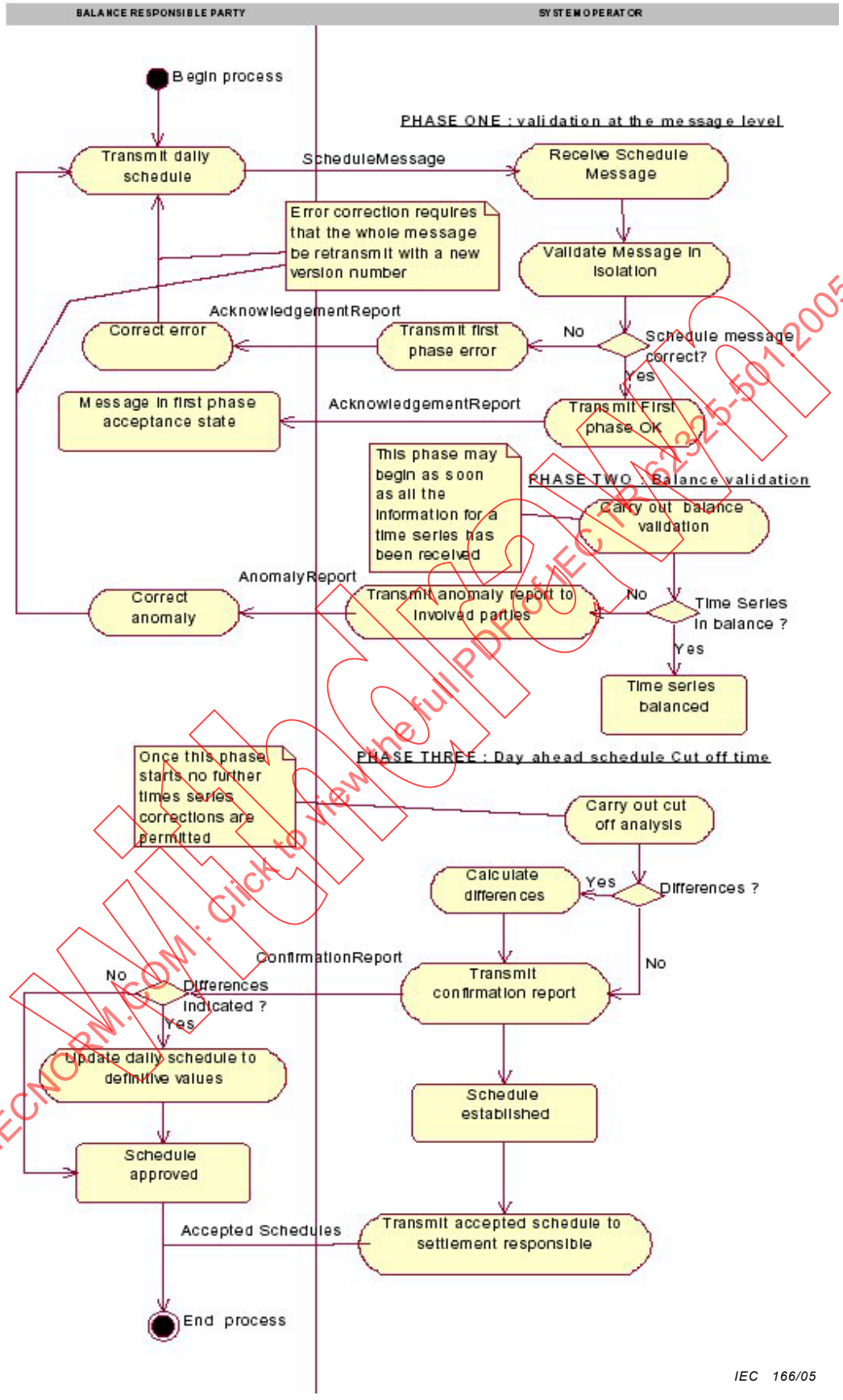
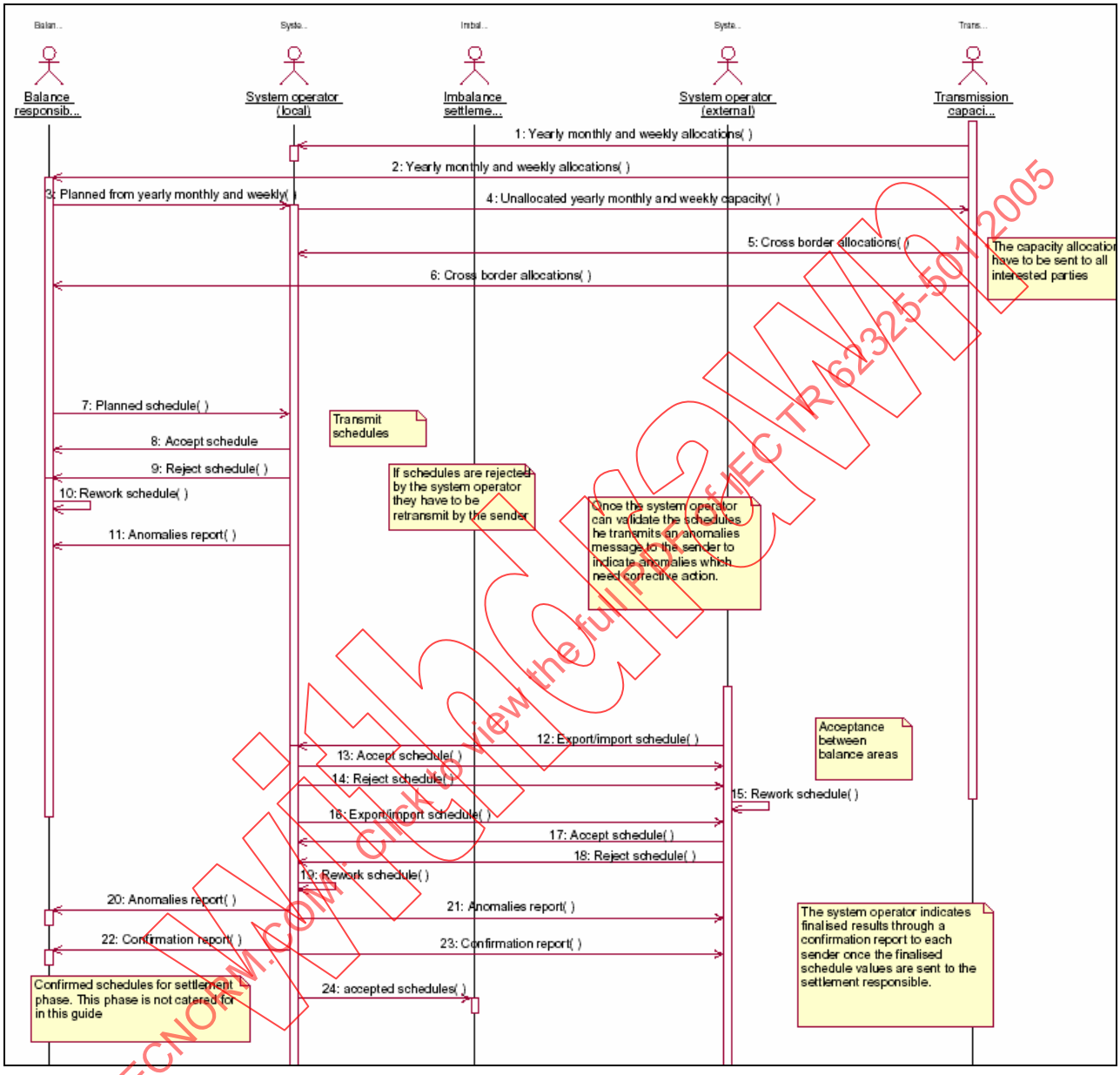


Figure A.1 – Schedule planning transmission process

The ETSO process for scheduling (ETSO (European Transmission System Operator) Scheduling System, ESS version 1.0.) is used for the following example which implements only phases 7 through 11 and 24 which includes the document flow between the Balance Responsible and the Transmission System Operator (phases 7 to 11) and the transmission of the accepted schedule to the Imbalance Settlement Responsible (phase 24).



IEC 167/05

Figure A.2 – Sequence diagram for the document flow without business signals

Figure A.2 shows the sequence diagram for the document flow of the example.

This example uses only one BPSS instance. It is assumed that the Transmission System Operator has created its BPSS instance and has put this XML documents in the registry. It is further assumed that the Balance Responsible Party retrieved the BPSS instance from the registry and agreed on what was specified in it. Each partner has a CPP and agrees on a common CPA in order to conduct e-business. All the XML configuration files are given in Annex B. The two partners can then conduct e-business. The MSH, which is used for this example, performs reliable messaging using SSL (one of the optional security services among others).

A.3 Registry

Figure A.3 shows a typical ebXML scenario for schedule planning process.

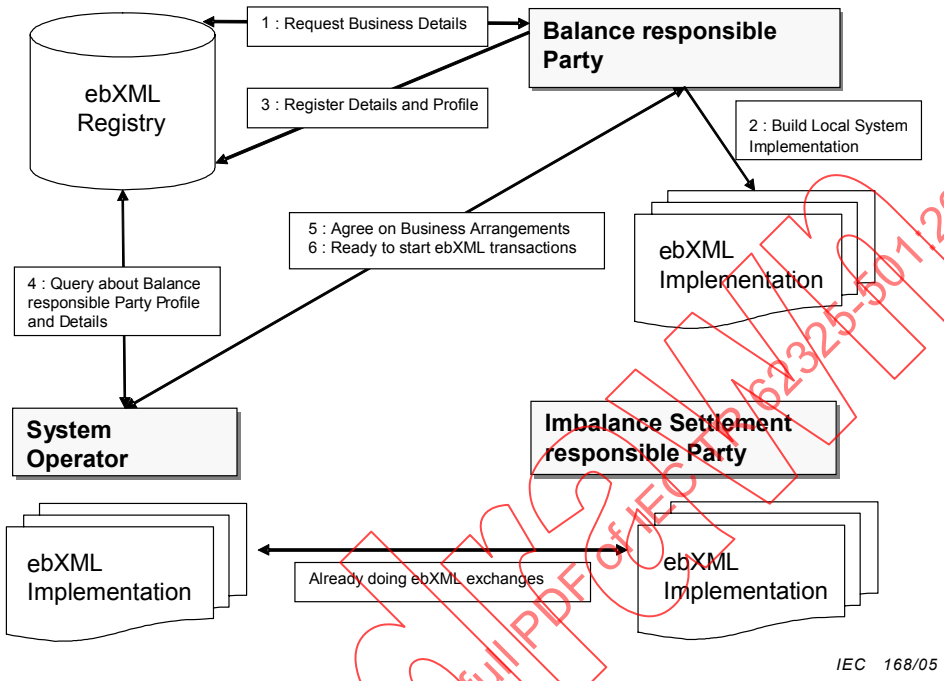


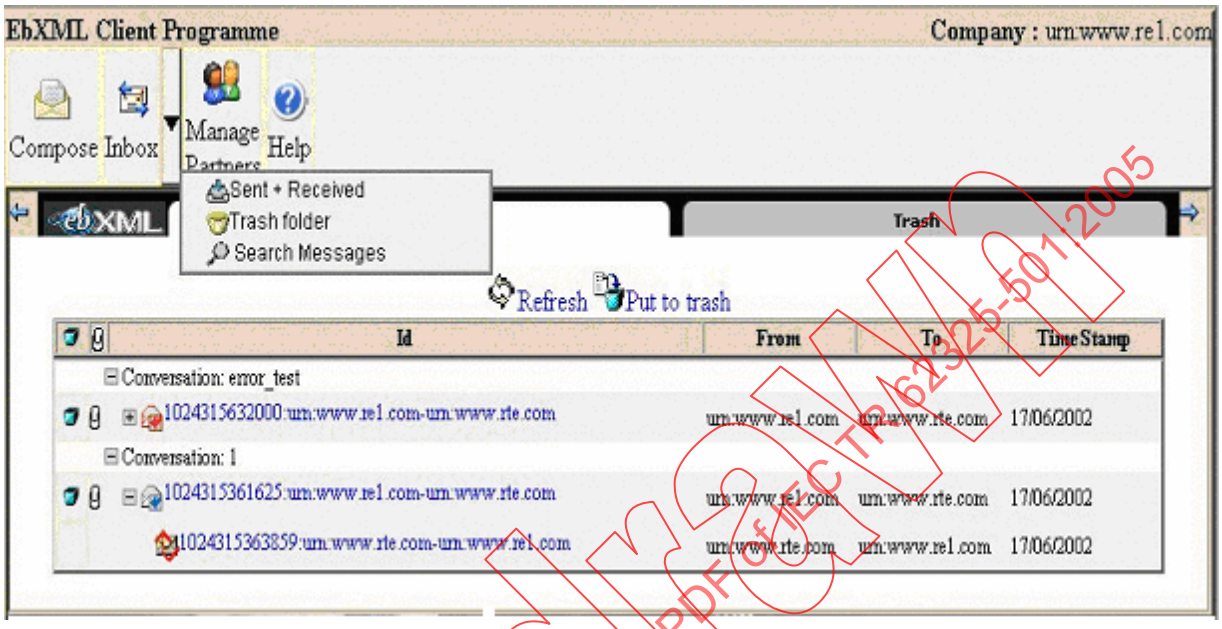
Figure A.3 – ebXML registry scenario for scheduling

In this scenario, the Transmission System Operator (TSO) and the Imbalance Settlement Responsible (ISR) already have a local ebXML implementation and are able to conduct ebXML transactions.

1) The Balance Responsible Party (BRP) wants to send a schedule to his TSO. As his TSO works with ebXML, the BRP decides to implement his own ebXML system. The BRP consults an ebXML registry to examine specifications and example use-case details. Therefore, he knows about TSO's ebXML profile, TSO's BPSS and TSO's CPP.
2) The BRP then implements an ebXML compliant application.
3) The BRP submits, to an ebXML registry, his business profile information, which describes his new ebXML related functionality.
4) The TSO queries BRP ebXML profile and details (BPSS, CPP).
5) The TSO then sends a request to the BRP requesting that they engage in a business scenario using ebXML and submits a proposed business arrangement which outlines the proposed business process, business documents, messaging arrangements, security requirements, etc. Upon negotiation of this proposal, a collaboration agreement is made between TSO and BRP which forms part of the trading partner agreement. It leads to the creation of the CPA by examining the BPSS and CPP stored in the registry.
6) TSO and BRP are now ready to engage in exchange of schedules using an ebXML conformant messaging. The CPA will enable the automation of ebXML data exchange.

A.4 The client

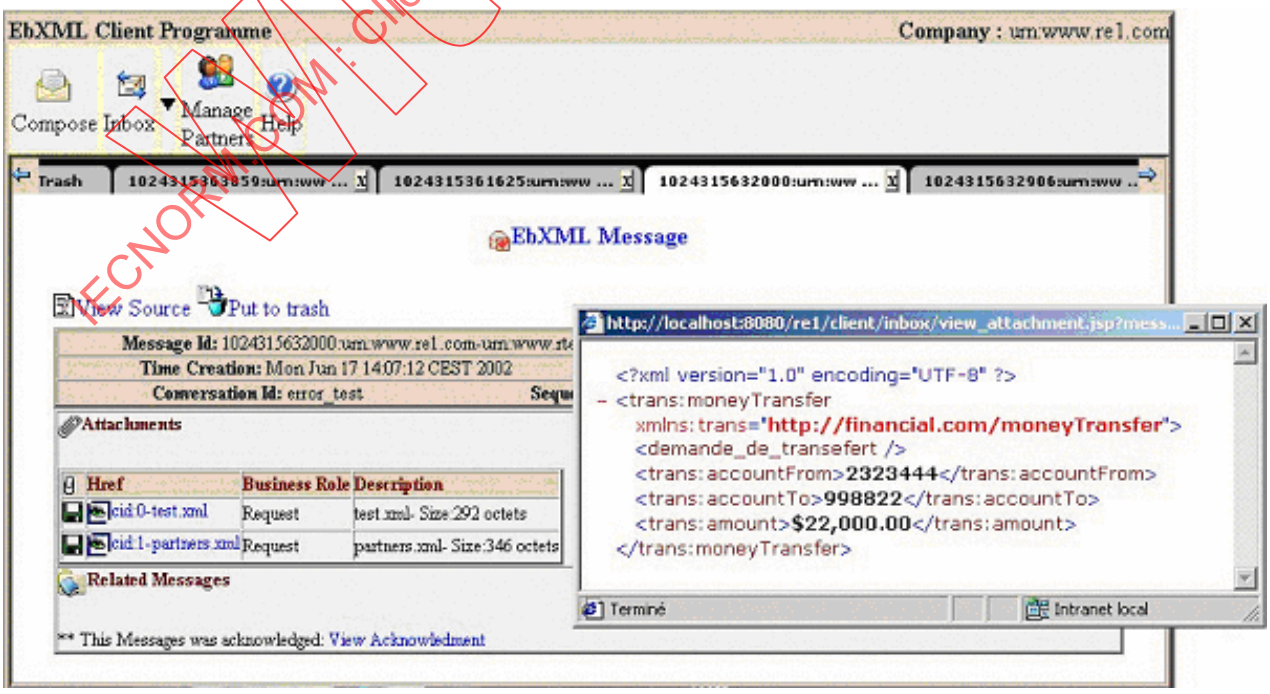
Figure A.4 shows a simple example client with which the ebXML documents are received and sent. In this example, it is assumed that the client is used by the Balance Responsible Party to send a generation schedule for the day ahead.



IEC 169/05

Figure A.4 – ebXML client where the messages are sent and received

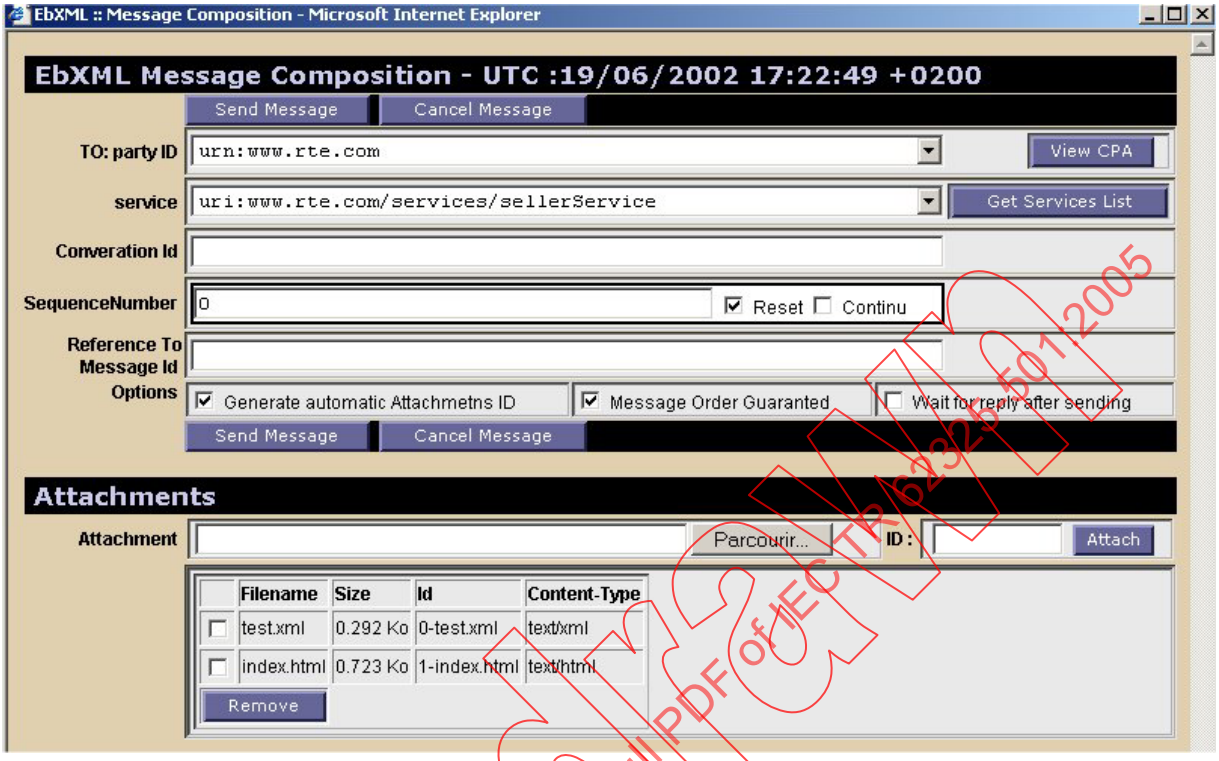
Figure A.4 shows the messages that where sent and received and the name of the conversation in which they are. Just under a message, we can see if the ebXML acknowledgement was received or not, (there is a cross on the left of the envelope icon). We can visualise the acknowledgement underneath the original message (the sequence of messages related to the same conversation has a tree representation).



IEC 170/05

Figure A.5 – Visualization of a message

Figure A.5 shows the visualisation of a message, if we click on the name of the message, we can see it. If there are one or more attachments within a message envelope, it is also possible to visualise them.



IEC 171/05

Figure A.6 – Interface used to compose messages

Figure A.6 shows the interface used to compose ebXML messages for transactions between the market participants.

A.5 Configuration

A.5.1 DTD and XML files

Figure A.7 below shows the ebXML BPSS DTD from the ebXML specification.

```
<!-- edited with XML Spy v3.5 NT (http://www.xmlspy.com) by Kurt Kanaskie (Lucent Technologies) -->
<!-- ================================================================= -->
<!-- Editor:   Kurt Kanaskie (Lucent Technologies)                     -->
<!-- Version:  Version 1.01                                            -->
<!-- Updated:  2001-05-24                                              -->
<!--                                                                   -->
<!-- Public Identifier:                                                -->
<!--          "-//ebXML//DTD Process Specification ver 1.0//EN"        -->
<!--                                                                   -->
<!-- Purpose:                                                          -->
<!--    The ebXML Specification DTD provides a standard                -->
<!--    framework by which business systems may be                     -->
<!--    configured to support execution of business                    -->
<!--    transactions.  It is based upon prior UN/CEFACT                -->
<!--    work, specifically the meta model behind the                   -->
<!--    UN/CEFACT Unified Modeling Methodology (UMM) defined           -->
<!--    in the N090R10 specification.                                  -->
<!--                                                                   -->
```

```
<!--    The Specification Schema supports the specification      -->
<!--    of Business Transactions and the choreography of         -->
<!--    Business Transactions into Business Collaborations.      -->
<!--                                                             -->
<!-- Notes:                                                      -->
<!--    time periods are represented using ISO 8601 format       -->
<!--    (e.g. P2D for 2 Days, P2H30M for 2 Hours 30 Minutes      -->
<!--                                                             -->
<!--    Naming and reference is based on convention that an      -->
<!--    Element with a name attribute (e.g. AuthorizedRole)      -->
<!--    is referenced by an attribute in another element with    -->
<!--    the name in lowerCamelCase (e.g. authorizedRole).        -->
<!--                                                             -->
<!--    fromBusinessState and toBusinessState refer to the       -->
<!--    the names of a BusinessTransactionActivity,              -->
<!--    CollaborationActivity, Fork, and Join, all are targets for -->
<!--    from/to in Transition. This deviates from the normal     -->
<!--    convention of lowerCamelCase attribute name              -->
<!--    BusinessState is used as a generic term for:             -->
<!--    Fork, Join, Success, Failure                             -->
<!--                                                             -->
<!-- Constraints:                                                -->
<!--    – attributes location, logicalModel, pattern, specification -->
<!--      uri, are of type xsd:anyURI                            -->
<!--    – attributes timeTo* are of type xsd:duration            -->
<!--                                                             -->
<!-- ========================================================== -->
<!ELEMENT ProcessSpecification (Documentation*, SubstitutionSet*, (Include | BusinessDocument |
ProcessSpecification | Package | BinaryCollaboration | BusinessTransaction |
MultiPartyCollaboration)*)>
<!ATTLIST ProcessSpecification
    name ID #REQUIRED
    uuid CDATA #REQUIRED
    version CDATA #REQUIRED
>
<!ELEMENT Documentation (#PCDATA)>
<!ATTLIST Documentation
    uri CDATA #IMPLIED
>
<!ELEMENT Include (Documentation*)>
<!ATTLIST Include
    name CDATA #REQUIRED
    uuid CDATA #REQUIRED
    uri CDATA #REQUIRED
    version CDATA #REQUIRED
>
<!ELEMENT BusinessDocument (ConditionExpression?, Documentation*)>
<!ATTLIST BusinessDocument
    name CDATA #REQUIRED
    nameID ID #IMPLIED
    specificationLocation CDATA #IMPLIED
    specificationElement CDATA #IMPLIED
>
<!ELEMENT ConditionExpression (Documentation*)>
<!ATTLIST ConditionExpression
    expressionLanguage CDATA #IMPLIED
    expression CDATA #IMPLIED
>
<!ELEMENT SubstitutionSet (DocumentSubstitution | AttributeSubstitution | Documentation)*>
<!ATTLIST SubstitutionSet
    name CDATA #IMPLIED
```

```
    nameId IDREF #IMPLIED
    applyToScope CDATA #IMPLIED
>
<!ELEMENT DocumentSubstitution (Documentation*)>
<!ATTLIST DocumentSubstitution
    originalBusinessDocument CDATA #IMPLIED
    originalBusinessDocumentID IDREF #IMPLIED
    substituteBusinessDocument CDATA #IMPLIED
    substituteBusinessDocumentId IDREF #IMPLIED
>
<!ELEMENT AttributeSubstitution (Documentation*)>
<!ATTLIST AttributeSubstitution
    attributeName CDATA #IMPLIED
    value CDATA #IMPLIED
>
<!ELEMENT Package (Documentation*, (Package | BinaryCollaboration | BusinessTransaction |
MultiPartyCollaboration)*)>
<!ATTLIST Package
    name CDATA #REQUIRED
    nameID ID #IMPLIED
>
<!ELEMENT BinaryCollaboration (Documentation*, InitiatingRole, RespondingRole, (Documentation |
Start | Transition | Success | Failure | BusinessTransactionActivity | CollaborationActivity | Fork |
Join)*)>
<!ATTLIST BinaryCollaboration
    name CDATA #REQUIRED
    nameID ID #IMPLIED
    pattern CDATA #IMPLIED
    beginsWhen CDATA #IMPLIED
    endsWhen CDATA #IMPLIED
    preCondition CDATA #IMPLIED
    postCondition CDATA #IMPLIED
    timeToPerform CDATA #IMPLIED
>
<!ELEMENT MultiPartyCollaboration (Documentation*, BusinessPartnerRole*)>
<!ATTLIST MultiPartyCollaboration
    name CDATA #REQUIRED
    nameID ID #IMPLIED
>
<!ELEMENT InitiatingRole (Documentation*)>
<!ATTLIST InitiatingRole
    name CDATA #REQUIRED
    nameID ID #IMPLIED
>
<!ELEMENT RespondingRole (Documentation*)>
<!ATTLIST RespondingRole
    name CDATA #REQUIRED
    nameID ID #IMPLIED
>
<!-- A BusinessState is one of Start, Success, Failure, Fork, Join, BusinessTransactionActivity or
CollaborationActivity -->
<!-- fromBusinessState and toBusinessState are fully qualified using XPath -->
<!ELEMENT Transition (ConditionExpression?, Documentation*)>
<!ATTLIST Transition
    onInitiation (true | false) "false"
    fromBusinessState CDATA #IMPLIED
    fromBusinessStateIDRef IDREF #IMPLIED
    toBusinessState CDATA #IMPLIED
    toBusinessStateIDRef IDREF #IMPLIED
    conditionGuard (Success | BusinessFailure | TechnicalFailure | AnyFailure) #IMPLIED
>
```

```
<!-- Start is a special type of Transition in that it only has a destination -->
<!ELEMENT Start (Documentation*)>
<!ATTLIST Start
    toBusinessState CDATA #REQUIRED
    toBusinessStateIDRef IDREF #IMPLIED
>
<!-- Success is a special type of Transition in that it only has a origination -->
<!ELEMENT Success (ConditionExpression?, Documentation*)>
<!ATTLIST Success
    fromBusinessState CDATA #REQUIRED
    fromBusinessStateIDRef IDREF #IMPLIED
    conditionGuard (Success | BusinessFailure | TechnicalFailure | AnyFailure) #IMPLIED
>
<!-- Failure is a special type of Transition in that it only has a origination -->
<!ELEMENT Failure (ConditionExpression?, Documentation*)>
<!ATTLIST Failure
    fromBusinessState CDATA #REQUIRED
    fromBusinessStateIDRef IDREF #IMPLIED
    conditionGuard (Success | BusinessFailure | TechnicalFailure | AnyFailure) #IMPLIED
>
<!-- Fork is a special type of BusinessState that can be transitioned to -->
<!ELEMENT Fork (Documentation*)>
<!ATTLIST Fork
    name CDATA #REQUIRED
    nameID ID #IMPLIED
>
<!-- Join is a special type of BusinessState that can be transitioned to -->
<!ELEMENT Join (Documentation*)>
<!ATTLIST Join
    name CDATA #REQUIRED
    nameID ID #IMPLIED
    waitForAll (true | false) "true"
>
<!-- fromAuthorizedRole and toAuthorizedRole are fully qualified using XPath -->
<!-- BusinessTransactionActivity is a BusinessState that can be transitioned to -->
<!ELEMENT BusinessTransactionActivity (Documentation*)>
<!ATTLIST BusinessTransactionActivity
    name CDATA #REQUIRED
    nameID ID #IMPLIED
    businessTransaction CDATA #REQUIRED
    businessTransactionIDRef IDREF #IMPLIED
    fromAuthorizedRole CDATA #REQUIRED
    fromAuthorizedRoleIDRef IDREF #IMPLIED
    toAuthorizedRole CDATA #REQUIRED
    toAuthorizedRoleIDRef IDREF #IMPLIED
    isConcurrent (true | false) "true"
    isLegallyBinding (true | false) "true"
    timeToPerform CDATA #IMPLIED
>
<!-- fromAuthorizedRole and toAuthorizedRole are fully qualified using XPath -->
<!-- CollaborationActivity is a BusinessState that can be transitioned to -->
<!ELEMENT CollaborationActivity (Documentation*)>
<!ATTLIST CollaborationActivity
    name CDATA #REQUIRED
    nameID ID #IMPLIED
    fromAuthorizedRole CDATA #REQUIRED
    fromAuthorizedRoleIDRef IDREF #IMPLIED
    toAuthorizedRole CDATA #REQUIRED
    toAuthorizedRoleIDRef IDREF #IMPLIED
    binaryCollaboration CDATA #REQUIRED
    binaryCollaborationIDRef IDREF #IMPLIED
```