



UN/CEFACT

United Nations Centre for Trade Facilitation and Electronic Business

1

2

3

4

UN/CEFACT – Business Process Specification Schema Technical Specification

5

Version 1.11

6

30 Sep, 2005

Table of Contents

| | | |
|----|--|-----------|
| 7 | | |
| 8 | | |
| 9 | 1.0 Status of this Document..... | 4 |
| 10 | 2.0 UN/CEFACT - BPSS Technical Specification Project Team Participants..... | 5 |
| 11 | 2.1 Disclaimer | 5 |
| 12 | 2.2 Contact Information | 5 |
| 13 | 3.0 Introduction..... | 6 |
| 14 | 3.1 Summary of Contents of Document | 7 |
| 15 | 3.2 Audience | 7 |
| 16 | 3.3 Related Documents | 7 |
| 17 | 3.4 Prerequisites | 8 |
| 18 | 4.0 Design Objectives | 9 |
| 19 | 4.1 Goals/Objectives/Requirements/Problem Description of the UN/CEFACT Business | |
| 20 | Process Specification Schema..... | 9 |
| 21 | 4.2 Caveats and Assumptions | 9 |
| 22 | 5.0 Language Overview | 14 |
| 23 | 5.1 UML Representation of Business Process Specification Schema | 16 |
| 24 | 5.2 XML Schema representation of Business Process Specification Schema..... | 16 |
| 25 | 5.3 UMM Business Process Interaction Patterns | 16 |
| 26 | 5.4 Business Signal Definitions | 17 |
| 27 | 5.5 Production Rules | 17 |
| 28 | 5.6 Relationship to CPP/CPA | 17 |
| 29 | 5.7 Relationship to Business Documents..... | 18 |
| 30 | 5.8 Relationship to ebXML Message Service Specification | 18 |
| 31 | 5.9 Key Concepts of the ebXML Business Process Specification Schema..... | 18 |
| 32 | 5.10 How to use the ebXML Business Process Specification Schema..... | 23 |
| 33 | 5.11 How ebXML Business Process Specification Schema is used with other ebXML | |
| 34 | specifications..... | 23 |
| 35 | 5.12 How to design collaborations and transactions, re-using at design time | 25 |
| 36 | 5.13 Core Business Transaction Semantics | 55 |
| 37 | 5.14 Run time Business Transaction semantics | 63 |
| 38 | 5.15 Runtime Collaboration Semantics | 69 |
| 39 | 5.16 Where the ebXML Business Process Specification Schema May Be Implemented | |
| 40 | | 70 |
| 41 | 5.17 Guidelines for Business Service Interface Interoperability | 70 |
| 42 | 5.18 Collaboration and transaction well-formedness rules..... | 71 |
| 43 | 6.0 ebXML Business Process Specification Schema — | 72 |
| 44 | 6.1 Introduction..... | 72 |
| 45 | 6.2 XML to UML cross-reference | 124 |
| 46 | 6.3 Scoped Name Reference | 125 |
| 47 | 6.4 Sample XML document against above Schema | 126 |

| | | |
|----|--|------------|
| 48 | 7.0 Business signal structures..... | 127 |
| 49 | 7.1 Signal Schema..... | 127 |
| 50 | 7.2 Documentation for Signal Schema | 130 |
| 51 | 8.0 Production Rules..... | 142 |
| 52 | Appendix A: Sample XML Business Process Specification Schema Instance | 144 |
| 53 | Appendix B: Sample XML Signals..... | 148 |
| 54 | Copyright Statement..... | 150 |
| 55 | | |

1.0 Status of this Document

This Technical Specification is being developed in accordance with the UN/CEFACT/TRADE/22 Open Development Process for Technical Specifications. This document has been approved by the the United Nations Centre for Trade Facilitation and Electronic Business (UN/CEFACT) Techniques and Methodology Group (TMG) for promulgation as a UN/CEFACT Technical Specification in accordance with the ODP.

This document contains information to guide in the interpretation or implementation.

Distribution of this document is unlimited.

This “Approved” (For Implementation) version: *UN/CEFACT Business Process Specification Schema Technical Specification*, Version 1.11 of 30 September 2005

Previous “Approved” (For Implementation) version: *UN/CEFACT – ebXML Business Process Specification Schema*, Version 1.10 of 18 October 2003

Previous Draft (For Review) version: *UN/CEFACT – ebXML Business Process Specification Schema* Version 1.09 of 25 August 2003

2.0 UN/CEFACT - BPSS Technical Specification Project Team Participants

We would like to recognize the following for their significant participation to the development of this document.

Project Team Chair: Karlson Hsia, Institute for Information Industry, Taiwan

Lead Editor: Chao-Hsiang Cheng, Institute for Information Industry, Taiwan

Editing Team Members: Karlson Hsia, Institute for Information Industry, Taiwan
Chao-Hsiang Cheng, Institute for Information Industry, Taiwan
Sheng-Chi Chen, Institute for Information Industry, Taiwan

Contributors:

- Gunther Stuhec, SAP AG, Germany
- Christian Huemer, University of Vienna, Austria
- Anders W. Tell, Toolsmith, Sweden
- Mary Kay Blantz, AIAG, USA
- James Allen Wilson, KCX, USA
- Jens Dietrich, OSCI, Germany
- Ed Buchinski, Treasury Board of Canada Secretariat, Canada
- Steve Capell, RedWahoo, Australia
- Mark Crawford, LMI, USA
- Michael Dill, Gefeg, Germany
- Puala Heilig, Worldspan, USA
- William McCarthy, Michigan State University, USA
- Sue Probert, United Kingdom

2.1 Disclaimer

The views and specification expressed in this document are those of the authors and are not necessarily those of their employers. The authors and their employers specifically disclaim responsibility for any problems arising from correct or incorrect implementation or use of this technical specification.

2.2 Contact Information

TMG BPSS Project Chair: Karlson Hsia, Institute for Information Industry, Taiwan
E-mail address: khhsia@iii.org.tw

TMG BPSS Lead Editor: Chao-Hsiang Cheng, Institute for Information Industry, Taiwan
E-mail address: cch1002@iii.org.tw

3.0 Introduction

Executive Summary

The ebXML Business Process technical specification defines a standard language by which business systems MAY be configured to support execution of business collaborations consisting of business transactions. It is based upon prior UN/CEFACT work, specifically the metamodel behind the UN/CEFACT Modeling Methodology (UMM) defined in the “UN/CEFACT Modeling Methodology - Meta Model - Revision 12 (2003-01-17)” specification.

The BPSS technical specification supports the specification of Business Transactions and the choreography of Business Transactions into Business Collaborations. All Business Transactions are implemented using one of many available standard patterns. These patterns are defined in the UMM specification. A pattern is not executable, it rather specifies the type of the message exchange (request, response and signals) that applies for a given business transaction definition. It is a way to define classes of business transaction definitions. These patterns could potentially be related to different classes of electronic commerce transactions.

The current version of the BPSS technical specification addresses collaborations between two parties (Binary Collaborations). Collaborations involving more than two business partners (Multiparty Collaborations) have been deprecated.

3.1 Summary of Contents of Document

This document describes the ebXML Business Process Specification Schema.

This document describes it in its UML form and provides the corresponding XML Schema which every BPSS instance must conform to.

The document first introduces general concepts and semantics, then applies these semantics in a detail discussion of each part of the model. The document then specifies all elements in the UML form, and then in the XML form.

The keywords MUST, MUST NOT, REQUIRED, SHALL, SHALL NOT, SHOULD, SHOULD NOT, RECOMMENDED, MAY, and OPTIONAL, when they appear in this document, are to be interpreted as described in RFC 2119 [Bra97].

3.2 Audience

The primary audience is technical implementers of ebXML. We define a business process analyst as someone who applies the UN/CEFACT Modeling Methodology (UMM) which defines a process that centers around interviewing business people.

Additional audiences are designers of business process definition tools who need to specify the conversion of user input in the tool into the XML representation of the Specification Schema.

3.3 Related Documents

Documents:

As mentioned above, other documents provide detailed definitions of some of the components of the ebXML Business Process Specification Schema and of their inter-relationship. They include ebXML Specifications on the following topics:

- ebXML Technical Architecture Specification, version 1.04
- ebXML Core Components Dictionary, version 1.04
- ebXML Naming Convention for Core Components, version 1.04
- ebXML Collaboration-Protocol Profile and Agreement Specification V2.0
- ebXML Business Process and Business Information Analysis Overview, version 1.0
- ebXML Business Process Analysis Worksheets & Guidelines, version 1.0
- ebXML E-Commerce Patterns, version 1.0
- ebXML Catalog of Common Business Processes, version 1.0
- ebXML Message Service Specification V2.0

- 156 ● UN/CEFACT Modeling Methodology (UMM) as defined in the N090R10
157 specification

158 **Normative References:**

- 159 1. UN/CEFACT Modeling Methodology - Meta Model - Revision 12 (2003-01-17)
160 specification, http://www.untmg.org/artifacts/UMM_Metamodel_2003-01-17.pdf
- 161 2. UN/CEFACT Modeling Methodology N090 Revision 10 (2001-11-01)
162 http://www.untmg.org/artifacts/UMM_N090R10_2001-11_01.zip
- 163 3. UN/CEFACT Core Components Technical Specification, Version 2.01,
164 http://www.untmg.org/artifacts/CCTS_v2.01_2003-11-15.pdf
- 165 4. ebXML Technical Architecture Specification, version 1.04,
166 <http://www.ebxml.org/specs/ebTA.pdf>
- 167 5. Key Words for use in RFCs to Indicate Requirement Levels, Internet Engineering Task
168 Force RFC 2119, <http://www.ietf.org/rfc/rfc2119.txt>
- 169 6. Extensible Markup Language (XML), World Wide Web Consortium,
170 <http://www.w3.org/XML>
- 171 7. XML Schema Part 1: Structures, Worldwide Web Consortium,
172 <http://www.w3.org/TR/xmlschema-1/>
- 173 8. XML Schema Part 2: Datatypes, Worldwide Web Consortium,
174 <http://www.w3.org/TR/xmlschema-2/>
- 175 9. ebXML Message Service Specification, Version 2.0, [http://www.oasis-](http://www.oasis-open.org/committees/ebxml-msg/documents/ebMS_v2_0.pdf)
176 [open.org/committees/ebxml-msg/documents/ebMS_v2_0.pdf](http://www.oasis-open.org/committees/ebxml-msg/documents/ebMS_v2_0.pdf)
- 177 10. ebXML Registry Services Specification, Version 2.0, [http://www.oasis-](http://www.oasis-open.org/committees/regrep/documents/2.0/specs/ebrs.pdf)
178 [open.org/committees/regrep/documents/2.0/specs/ebrs.pdf](http://www.oasis-open.org/committees/regrep/documents/2.0/specs/ebrs.pdf)
- 179 11. ebXML Collaboration-Protocol Profile and Agreement Specification, Version 2.0,
180 <http://www.oasis-open.org/committees/ebxml-cppa/documents/ebcpp-2.0.pdf>
- 181 12. Multipurpose Internet Mail Extensions (MIME) Part One, IETF RFC 2958: Format of
182 Internet Message Bodies, N. Freed, N. Borenstein, Authors. Internet Engineering Task
183 Force, November 1996. Available at <http://www.ietf.org/rfc/rfc2045.txt>

184

185 **3.4 Prerequisites**

186 It is assumed that the audience will be familiar with or have knowledge of the following
187 technologies and techniques:

- 188 ● Business process modeling techniques and principles as defines in UN/CEFACT's
189 Modeling Methodology (UMM)
- 190 ● The UML syntax and semantics
- 191 ● The Extensible Markup Language (XML)

4.0 Design Objectives

4.1 Goals/Objectives/Requirements/Problem Description of the UN/CEFACT Business Process Specification Schema

BPSS Instances describe interoperable business processes that allow business partners to collaborate. These models must be executed by software components that collaborate on behalf of the business partners.

The goal of the ebXML Business Process Specification Schema is to provide the bridge between e-business process modeling and specification of e-business software components.

The ebXML Business Process Specification Schema technical specification provides for the nominal set of specification elements necessary to specify a collaboration between business partners, and to provide configuration parameters for the partners' runtime systems in order to execute that collaboration between a set of e-business software components.

A business process specification created with the ebXML Business Process Specification Schema is referred to as a BPSS instance. The *ebXML Specification Schema* is available as an XML Schema (<http://www.w3.org/2001/XMLSchema>) format at this location: <http://www.untmg.org/downloads/General/approved/BPSS-v1pt11.xsd>. A UML description of elements of the schema is found in relevant sections of this document.

The UML version of the *ebXML Business Process Specification Schema* is merely a UML Class Diagram. It is not intended for the direct creation BPSS instances. Rather, it is a self-contained statement of all the specification elements and relationships required to be able to create an ebXML compliant Business Process Specification. Any methodologies and/or metamodels used for the creation of ebXML compliant Business Process Specifications must at minimum support these elements and relationships.

The XML Schema provides the specification for XML based BPSS instances.

The UML and XML based representations of the *ebXML Business Process Specification Schema* are unambiguously mapped to each other.

4.2 Caveats and Assumptions

This technical specification is designed to specify the run time aspects of a business collaboration.

It is recommended that the preferred methodology for creating an ebXML BPS shall be UN/CEFACT Modeling Methodology (UMM).

224 The *ebXML Business Process Specification Schema* does not by itself define Business
 225 Documents Structures. It is intended to work in conjunction with already existing Business
 226 Document definitions, and/or the document metamodel defined by the UN/CEFACT Core
 227 Components specifications.

228 **4.2.1 Relationship between ebXML Business Process Specification Schema and** 229 **UMM**

230 The UN/CEFACT Modeling Methodology (UMM) is a set of architectures, methodologies,
 231 business semantics, ontologies and reference models. The UMM offers a formal
 232 methodology for describing any Open-edi scenario as defined in ISO/IEC 14662, Open-edi
 233 Reference Model. Examples of an Open-edi scenario are purchasing and inventory
 234 management. The primary scope of the UMM is to provide "a perspective of business
 235 transactions limited to those aspects regarding the making of business decisions and
 236 commitments among organizations, which are needed for the description of a business
 237 transaction". The UMM provides a procedure for specifying (modeling) business processes
 238 involving information exchange in a technology neutral, implementation-independent
 239 manner.

240 This section describes the relationship between UMM and the ebXML Business Process
 241 Specification Schema.

242 The UMM Meta Model is a description of business semantics that allows Trading Partners
 243 to capture the details for a specific business scenario (a Business Process) using a consistent
 244 modeling methodology. A Business Process specification describes in detail how Trading
 245 Partners take on shared roles, relationships and responsibilities to facilitate interaction with
 246 other Trading Partners. The interaction between roles takes place as a choreographed set of
 247 Business Transactions. Each Business Transaction is expressed as an exchange of electronic
 248 Business Documents. The sequence of the exchange is determined by the Business Process,
 249 and by messaging and security considerations. Business Documents are composed from re-
 250 useable Business Information Entities, expressed in an appropriate format (XML, EDI,
 251 UBL, ...). At a lower level, Business Processes can be composed of re-useable Common
 252 Business Processes, and Business Information Entities can be composed of re-useable Core
 253 Components. Common Business Processes and Business Information Entities reside in a
 254 UMM Business Library.

255 The UMM Meta Model supports a set of Business Process viewpoints that provide a set of
 256 semantics (vocabulary) for each viewpoint and forms the basis for specification of the
 257 semantics and artifacts that are required to facilitate business process and information
 258 integration and interoperability. Using the UMM methodology and the UMM metamodel,
 259 the user may thus create a complete Business Process and Information Model. This model
 260 contains more information than what is required for configuring ebXML compliant software.
 261 Also the model is syntax independent and not directly interpretable by ebXML compliant
 262 software.

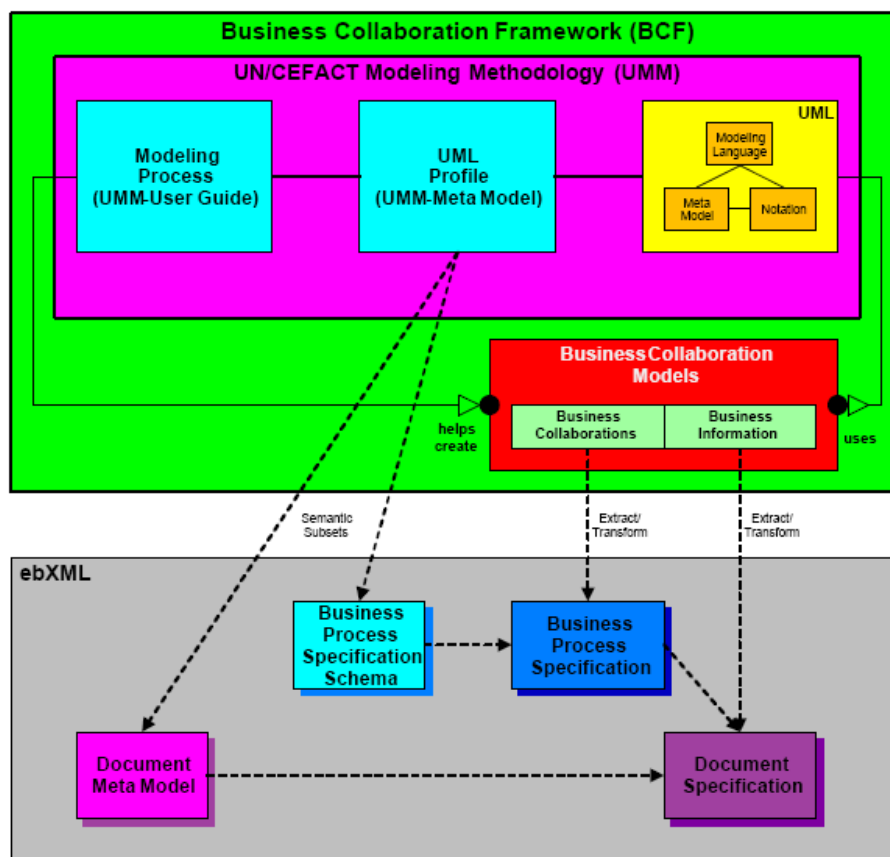
The ebXML Business Process Specification Schema provides an additional view of the UMM metamodel. This subset is provided to support the direct specification of the nominal set of elements necessary to configure a runtime system in order to execute a set of ebXML business transactions. By drawing out modeling elements from several of the other views, the ebXML Business Process Specification Schema forms a semantic subset of the UMM Meta Model. Using the ebXML Business Process Specification Schema the user may thus create a Business Process Specification that contains only the information required to configure ebXML compliant software, while other modeling elements of the UMM could be used to configure other software components such as a business process management system (BPMS).

It is expected that ebXML compliant software will be configured with XML instances conforming to the ebXML Business Process Specification Schema.

The concepts in BPSS is primarily based on UMM-Meta Model - Revision 12 published on Jan 17, 2003. The reference document of UMM meta model in Reference 2 could be found in TMG website: http://www.untmg.org/artifacts/UMM_Metamodel_2003-01-17.pdf. And the normative textual specification of UMM-MM document could be found in TMG website, UMM N090R10, which is published on Nov 1, 2001. This document illustrates the whole picture and the concepts of UMM methodology completely. The chapter 8 of this specification specifies the abstract syntax and semantics of each view in UMM meta model as well as the model management packages.

In order to understand UMM meta model, we recommend reference to chapter 8 of UMM N090R10 in the BPSS reference section with http://www.untmg.org/artifacts/UMM_N090R10_2001-11_01.zip.

288 The relationship between the UMM Meta Model and the ebXML Business Process
 289 Specification Schema is shown in Figure 1.



290

291 Figure 1. UMM Metamodel and ebXML Business Process Specification Schema

292 Using the UMM methodology, and drawing on content from the UMM Business Library a
 293 user may create complete Business Process and Information Model conforming to the
 294 UMM metamodel.

295 Since the *ebXML Business Process Specification Schema* is a semantic subset of the UMM
 296 metamodel, the user may then in an automated fashion extract from the Business Process
 297 and Information Model the required set of elements and relationships, and transform them
 298 into a BPSS instance conforming to the *ebXML Business Process Specification Schema*.

299 Likewise, since the UN/CEFACT Core Component (CC) document metamodel is aligned
 300 with the UMM Metamodel, the user may then in an automated fashion extract from the
 301 Business Process and Information Model the required set of elements and relationships, and
 302 transform them into an ebXML document model conforming to UN/CEFACT Core
 303 Component specifications.

304 The UN/CEFACT UMM and CC Specification are not part of the formal set of ebXML
305 specifications.

306 The remainder of this document focuses on the *ebXML Business Process Specification*
307 *Schema* and Business Process Specifications created with it. It is recommended that proper
308 Business Process and Information Modeling using the UMM has taken place prior to
309 beginning the activity of creating a Business Process Specification.

5.0 Language Overview

The ebXML *Business Process Specification Schema* defines a standard language for business process specification. As such, it works with the ebXML Collaboration Protocol Profile (CPP) and Collaboration Protocol Agreement (CPA) specifications to bridge the gap between Business Process Modeling and the configuration of ebXML compliant e-commerce software.

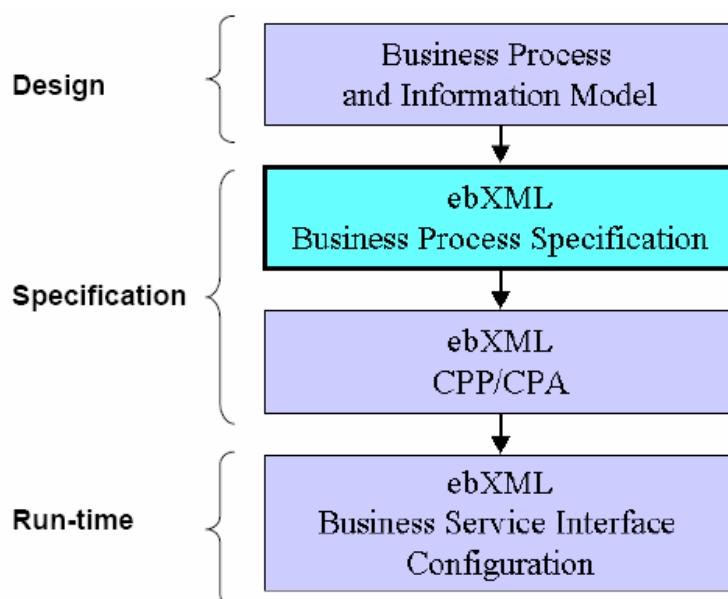


Figure 2. Business Process Specification and Business Service Interface Configuration

Using Business Process Modeling, a user may create a complete Business Process and Information Model.

Based on this Business Process and Information Model and using the ebXML *Business Process Specification Schema* the user will then extract and format the nominal set of elements necessary to configure an ebXML runtime system in order to execute a set of ebXML business transactions. The result is a *BPSS instance*.

Alternatively the ebXML *BPSS instance* may be created directly, without prior explicit business process modeling.

A *BPSS instance* contains the specification of Business Transactions and the choreography of these Business Transactions into Business Collaborations.

This *BPSS instance* is then the input to the formation of ebXML trading partner Collaboration Protocol Profiles and Collaboration Protocol Agreements.

These ebXML trading partner Collaboration Protocol Profiles and Collaboration Protocol Agreements in turn serve as configuration files for Business Service Interface (BSI) software component. The Business Service Interface Software represents any ebXML compliant component, which is able to be, configured from an ebXML BPSS instance and a CPA.

The architecture of the ebXML *Business Process Specification Schema* technical specification consists of the following functional components:

- UML representation of the *Business Process Specification Schema semantics*
- XML Schema definition of the *Business Process Specification Schema*. Each BPSS instance must conform to this schema definition.
- Production Rules defining the mapping from the UML representation of the *Business Process Specification Schema* to the XML Schema version
- Business Signal Definitions

Together these components allow you to specify the run time aspects of a business process model within the limitations of this current version of the BPSS. However, all the parameters of the ebXML *Business Process Specification Schema* are intended to be specified at design time. None of these parameters are specified or inferred at run-time.

These components are shown (inside the dotted box) in figure 3 below.

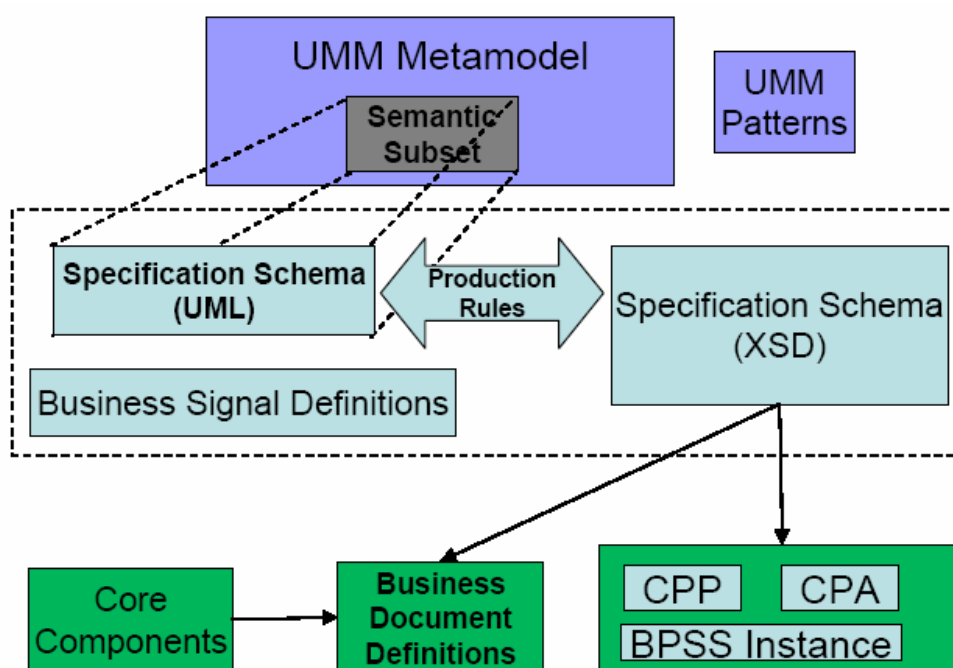


Figure 3. Relationship of *ebXML Business Process Specification Schema* to UMM, CPP/CPA and Core Components

The following provides a description of each of the components in the ebXML *Business Process Specification Schema* and their relationship to UMM, and UN/CEFACT Core Component and CPP/CPA:

5.1 UML Representation of Business Process Specification Schema

The UML representation of the ebXML Business Process Specification Schema is a semantic subset of the metamodel behind UMM as specified in UN/CEFACT Modeling Methodology - Meta Model - Revision 12 (2003-01-17). The UML representation of the ebXML Business Process Specification Schema is a UML Class Diagram.

5.2 XML Schema representation of Business Process Specification Schema

The corresponding XML Schema representation of the ebXML *Business Process Specification Schema* provides the specification for XML based instances of ebXML BPSS, and as a target for production rules from other representations. Thus, a user may either create a *BPSS instance* directly as an XML document, or may chose to use some other means of specification first and then apply production rules to arrive at the XML document version.

Any methodologies and/or metamodels used for the creation of ebXML BPSS instances must at a minimum support the production of the elements and relationships contained in the XML representation of the ebXML *Business Process Specification Schema technical specification*.

This XML Schema definition is isomorphic to the UML representation of the ebXML *Business Process Specification Schema*.

5.3 UMM Business Process Interaction Patterns

Any ebXML Business Service Interface software components should be able to be configured to execute the business processes specified in a *BPSS instance*. They do so by exchanging ebXML messages and business signals.

Each Business Transaction can be implemented using one of many available standard patterns. These patterns determine the actual exchange of messages and business signals between the partners to achieve the required electronic commerce transaction.

The Business Transaction Interaction Patterns set forth in the UN/CEFACT Modeling Methodology illustrate recommended permutations of message sequences as determined by

the type of business transaction defined and the timing policies specified in the transactions. While the UMM patterns themselves are not part of the ebXML specifications, all the security and timing parameters required to express the pattern properties are provided as attributes of elements in the ebXML *Business Process Specification Schema*.

5.4 Business Signal Definitions

A business signal is an object that is transmitted back to an activity that initiated the transfer of execution control. Business signals have specific business purpose and are separate from lower protocol and transport signals as specified in the ebXML Message Service Specification. The state of a given business transaction activity instance can be explicitly calculated at run-time by evaluating these signals. As such they are instrumental in establishing a business collaboration protocol that guarantees that the representation of the state of a business collaboration instance for each party, is the strictly identical for both parties. This is what we reference as “state alignment”.

The structures of ebXML business signals are ‘universal’ and do not vary from transaction to transaction. Thus, they can be defined once and for all. These schemas are included in the ebXML *BPSS technical specification* itself.

The Business Process Specification provides both the choreography of business signals, and the structure definition of the business payload of a business signal. The ebXML Message Service Specification provides a reliable messaging infrastructure upon which the ebXML BPSS technical specification builds its protocol for business state alignment via the use of business signals. The business signal payload structures provided herein are optional and normative and are intended to provide business and legal semantics to the business signals.

A Schema is provided for each of the possible business signals.

5.5 Production Rules

A set of production rules is provided, defining the mapping from the UML version of the ebXML *Business Process Specification Schema* to the XML version.

The primary purpose for these production rules is to govern the one-time generation of the Schema representation of the ebXML *Business Process Specification Schema* from the UML Class Diagram version of the ebXML *Business Process Specification Schema*.

5.6 Relationship to CPP/CPA

A *BPSS instance* is, along with protocol specifications, the object of the agreement between two parties. The BPSS instance is therefore incorporated with or referenced by ebXML

trading partner Collaboration Protocol Profiles (CPP) and Collaboration Protocol Agreements (CPA). Each CPP declares its support for one or more Roles within the *BPSS instance*. A BPSS instance is also a machine interpretable specification needed for an ebXML Business Service Interface, which will enforce its definition at run-time. The CPP profiles and CPA agreements contain further technical parameters resulting in a full specification of the run-time software at each trading partner.

5.7 Relationship to Business Documents

The *Business Process Specification Schema* does not by itself support the definition of Business Documents. Rather, a *BPSS instance* merely points to the definition of Business Documents. Such definitions may either be XML based, or – as attachments – may be any other structure, or completely unstructured.

5.8 Relationship to ebXML Message Service Specification

The Business Process Specification Schema will provide choreography of business messages and signals. The ebXML Message Service Specification provides the infrastructure for message / signal identification, typing, and integrity; as well as placing any one message in sequence with respect to other messages in the choreography.

5.9 Key Concepts of the ebXML Business Process Specification Schema

The ebXML *Business Process Specification Schema* specifies the structure and semantics of machine processable business collaborations definitions. These semantics are aligned with the one of UMM and represent a subset of the UMM semantics.

At a high level a business collaboration consists of a set of roles collaborating through a set of choreographed transactions by exchanging business documents.

Two or more business partners participate in the business collaboration through roles. The roles always exchange messages in the context of Business Transactions. Each Business Transaction consists of one or two predefined Business document flows. One or more Business Signals may additionally be exchanged as part of a Business Transaction to ensure state alignment of both parties. The business transactions are performed relative to each other as part of a choreography.

These basic semantics of a business collaboration are illustrated in Figure 4.

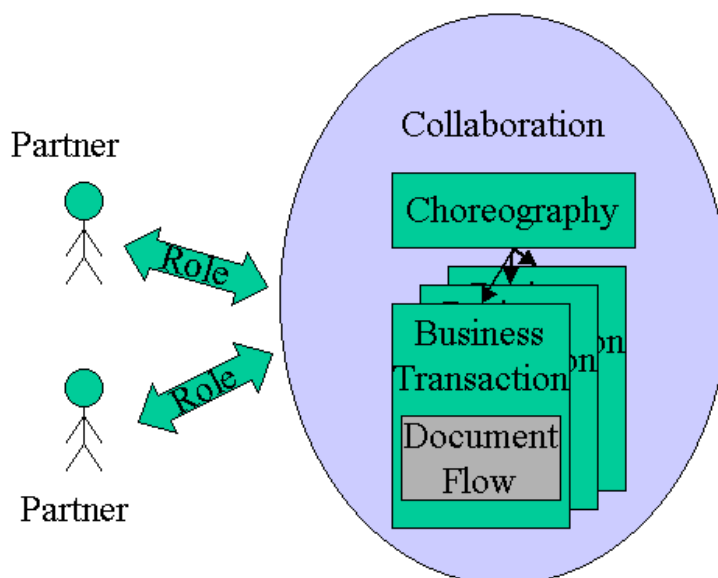


Figure 4. Illustration of the basic semantics of a business collaboration

The following section describes the concepts of a Business Collaboration, a Business Transaction, a Business document flow, and Choreography.

1. Business Collaborations

A business collaboration is a set of Business Transactions between business partners. Each partner plays one or more roles in the collaboration.

The ebXML *Business Process Specification Schema* supports two levels of business collaborations, Binary Collaborations and Multiparty Collaborations.

Binary Collaborations are between two roles only.

Multiparty Collaborations are between more than two roles, but such Multiparty Collaborations are always synthesized from two or more Binary Collaborations. For instance if Roles A, B, and C collaborate and all parties interact with each other, there will be a separate Binary Collaboration between A and B, one between B and C, and one between A and C. The Multiparty Collaboration will be the synthesis of these three Binary Collaborations. The concepts developed to specify multi-party collaboration are experimental and are being deprecated. It is recommended not to use this capability of the

467 specification as it might change substantially in future releases. The implementation of this
468 feature is therefore optional for any compliant ebXML Business Service Interface.

469 Binary Collaborations are expressed as a set of Business Activities between the two roles.
470 The Business Activity can be a Business Transaction Activity, i.e. the activity of conducting
471 a single Business Transaction, or a Collaboration Activity, i.e. the activity of conducting
472 another Binary Collaboration. An example of the former is the activity of “process purchase
473 order”. An example of the latter is the activity of “negotiating a contract”. In either case the
474 activities can be choreographed relative to other activities as per below.

475 The ability of a Binary Collaboration to have activities that in effect are executing other
476 Binary Collaborations is the key to recursive compositions of Binary Collaboration, and to
477 the re-use of Binary Collaborations. An activity, whether it is a Business Transaction
478 Activity or a Collaboration Activity represents the usage of a definition within a Binary
479 Collaboration Specification. For instance, a Business Transaction is defined once and for all,
480 but could appear several times – as a Business Transaction Activity -, sometimes even with
481 opposite roles, within the same binary collaboration definition.

482 In essence each Binary Collaboration is a re-useable protocol between two roles.

483 2. Business Transactions

484 A Business Transaction represents an atomic unit of work in a trading arrangement between
485 two business partners. The scope of the BPSS technical specification is not to cover how
486 BPSS Business Transactions are related to trading activities between business partners. This
487 is the role of the UMM. A Business Transaction is conducted between two parties playing
488 opposite roles in the transaction. The roles are always a requesting role and a responding
489 role. They are not specific roles like buyer or seller. These roles will be specified at the
490 Business Transaction Activity level, when the Business Transaction definition is used for a
491 specific purpose.

492 Like a Binary Collaboration, a Business Transaction is a re-useable protocol between two
493 roles. The way it is re-used is by referencing it from a Binary Collaboration through the use
494 of a Business Transaction Activity as per above. In a Business Transaction Activity the
495 roles of the Binary Collaboration are assigned to the execution of the Business Transaction.

496 Unlike a Binary Collaboration, however, the Business Transaction is atomic; it cannot be
497 decomposed into lower level Business Transactions that could be reused independently of
498 each other.

499 A Business Transaction is a very specialized and very constrained protocol, in order to
500 achieve very precise and enforceable transaction semantics. These semantics are expected to
501 be enforced by the software managing the transaction, i.e. an ebXML Business Service
502 Interface (BSI) software component.

A Business Transaction will always either succeed or fail both from a protocol and a business perspective. If it succeeds from both perspectives it may be designated as legally binding between the two partners, or otherwise govern their collaborative activity. If it fails it is null and void, and each partner must relinquish any mutual claim established by the transaction. In addition, if it fails from protocol perspective, each party must synchronize their state to the state prior the start of the transaction. For instance, a purchase order state should advance to “sent” when and only when a protocol success is reported by the BSI. In case of a business failure, the state has already been “synchronized” and it is the duty of each application to take the proper actions. A Business failure is any failure that is identified by an application during the processing of the business document(s) and based on information not available to the BPSS. For instance, a “reject purchase order” response document would be considered as a business failure. In this case, it is the role of the applications to mark the state of the purchase order appropriately.

There are six types of business transactions patterns as defined by UMM:

- Commercial Transaction
- Information Distribution
- Notification
- Query Response
- Request Confirm
- Request Response

3. Business Document flows

A business transaction is realized as Business Document flows between the requesting and responding roles. There is always a requesting Business Document, and optionally a responding Business Document, depending on the desired transaction configuration: e.g. one-way notification vs. two-way conversation.

Actual document definition is achieved using the UN/CEFACT Business Collaboration Models, or by some methodology external to ebXML but resulting in Schema definition (XSD or DTD) that an ebXML Business Process Specification can point to.

4. Choreography

The Business Collaboration Choreography describes the ordering and transitions between business transactions or sub collaborations within a binary collaboration. For example, in a UML tool this could be represented with a UML activity diagram. Actually, the choreography is specified in the ebXML Business Process Specification Schema using activity diagram concepts such as: start state, completion state, activities, forks, joins, decisions, transitions between activities, and guards on the transitions. However, it is beyond the scope of this document to specify a notation of a business collaboration.

5. Patterns

540 The ebXML Business Process Specification Schema provides a set of unambiguous
541 semantics, as a subset of UMM semantics, which enable us to specify transactions and
542 collaborations. Within these semantics the user community has flexibility to specify an
543 infinite number of specific transactions and collaborations. The use of predefined patterns
544 combines this flexibility with a consistency that facilitates faster design, faster
545 implementation, and enables generic processing.

546 A set of predefined transaction interaction patterns, defining common combinations of
547 transaction interaction parameter settings can be found in the UMM.

548 While the UMM transaction interaction patterns themselves are not part of the ebXML
549 BPSS technical specification, all the security and timing parameters required to express the
550 pattern properties are provided as attributes of elements in the Business Process
551 Specification Schema.

552 It is also anticipated that patterns for collaboration choreographies will emerge. An example
553 of such a pattern is in the ebXML E-Commerce Patterns.

554 Re-use, recursion, and patterns are among the key concepts of the ebXML Business Process
555 Specification Schema. The following section will illustrate these key concepts.

5.10 How to use the ebXML Business Process Specification Schema

The ebXML *Business Process Specification Schema* should be used wherever ebXML compliant software is being specified to execute Business Collaborations.

The ebXML *Business Process Specification Schema* is used to specify the business process related configuration parameters for configuring a BSI to execute these collaborations.

This section discusses

- How the ebXML *Business Process Specification Schema* fits in with other ebXML specifications.
- How to use the ebXML *Business Process Specification Schema* at design time, either for specifying brand new collaborations and transactions, or for re-using existing ones.
- How to specify core transaction semantics and parameters needed for a Collaboration-Protocol Profile and Agreement (CPP/CPA).
- Run-time transaction and collaboration semantics that the ebXML *Business Process Specification Schema* specifies and the Business Service Interface (BSI) is expected to manage.

5.11 How ebXML Business Process Specification Schema is used with other ebXML specifications

The ebXML *Business Process Specification Schema* provides the structure and semantics, as a subset of UMM semantics of Business Collaboration definitions.

A collaboration consists of a set of roles collaborating through a set of choreographed transactions by exchanging Business Documents.

As shown in Figure 5, a BPSS instance will reference, but not define, a set of required Business Documents. Within a BPSS instance, Business Documents are either defined by some external document specification, or assembled directly or indirectly from lower level information structures called core components. The assembly is based on a set of contexts, many of which are provided by the business processes, i.e. collaborations that use the documents in their document flows.

The combination of the business process specification and the document specification become the basis against which partners can make agreements on conducting electronic business with each other.

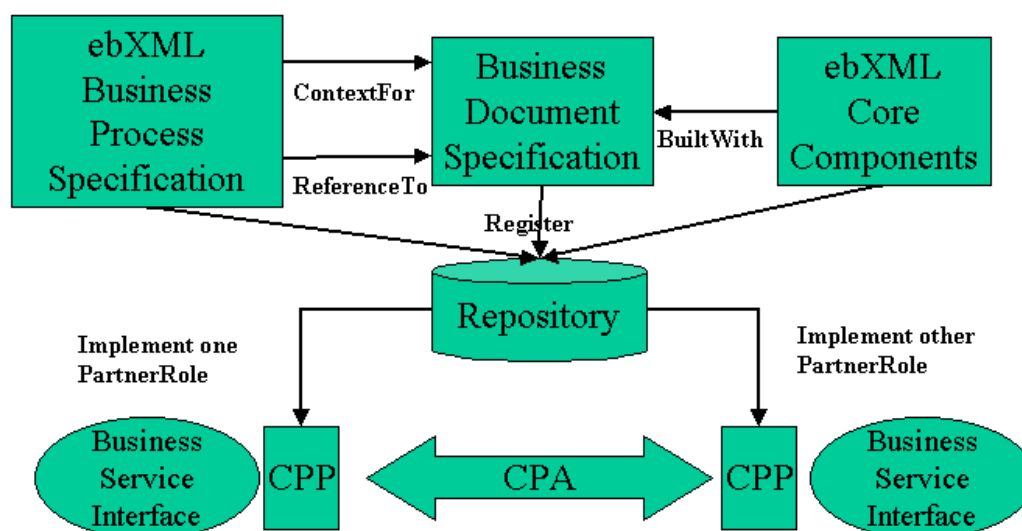


Figure 5. ebXML Business Process Specification Schema and other ebXML Specifications

The user will extract and transform the necessary information from an existing Business Process and Information Model. Associated production rules could aid in creating an XML representation of a *BPSS instance*.

Alternatively a user would use an XML based tool to produce the XML representation directly. Production rules could then aid in converting into XMI, so that it could be loaded into a UML tool, if required.

In either case, the XML representation of the *BPSS instance* gets stored in the ebXML repository and registered in the ebXML registry for future retrieval. The *BPSS instance* would be registered using classifiers derived during its design.

When implementers want to establish trading partner Collaboration Protocol Profile and Agreement the *BPSS instance* document, or the relevant parts of it, are simply referenced by the CPP and CPA XML documents. ebXML CPP and CPA XML documents can reference business process specifications in XML such as an ebXML BPSS instance .

Guided by the CPP and CPA specifications the resulting XML document then becomes the configuration file for one or more Business Service Interfaces (BSI), i.e. the software that will actually manage either partner's participation in the collaboration.

5.12 How to design collaborations and transactions, re-using at design time

This section describes the ebXML *Business Process Specification Schema* by building a complete Multiparty Collaboration BPSS instance from the bottom up, as follows:

1. Specify a Business Transaction
2. Specify the Business Document flow for a Business Transaction
3. Specify a Binary Collaboration re-using the Business Transaction
4. Specify a Choreography for the Binary Collaboration
5. Specify a higher level Binary Collaboration re-using the lower level Binary Collaboration
6. Specify a Multiparty Collaboration re-using Binary Collaborations

Although this section, for purposes of introduction, discusses the specification of collaboration from the bottom up, the ebXML *Business Process Specification Schema* is intended for specifying collaborations from the top down, re-using existing lower level content as much as possible.

The constructs listed above support the specification of fairly complex multi party collaborations. However, a BPSS instance may be as simple as a single Binary Collaboration referencing a single Business Transaction. This involves only numbers 1 through 3 above. Note the ebXML BPSS technical specification does not specify any BusinessProcess modeling methodology nor does it require the use of such methodology. Should a modeling methodology be needed, it is recommended to use the one of the UMM specification.

We have chosen a “drop ship” example which involves a buyer, a retailer, a vendor, and a credit organization. The order is placed by the buyer and fulfilled by the vendor. The credit authority makes sure that payments are made to appropriate creditors. We are using UML activity diagrams and use case diagrams to give a picturesc representation of this multi-party collaboration. This notation is non-normative and only here to help understand the example. It is used in a way that do not adhere to the UML semantics.

Figure 6 represents the overall multi-party collaboration. The conventation that we are using is such that an “activity” represents a binary collaboration between two roles. Since we have four roles represented here, we have adopted the following convention: the activity is placed in the swimlane of the role that starts the binary collaboration. The responding role is the one directly facing the activity. This is why the swimlane have different sizes.

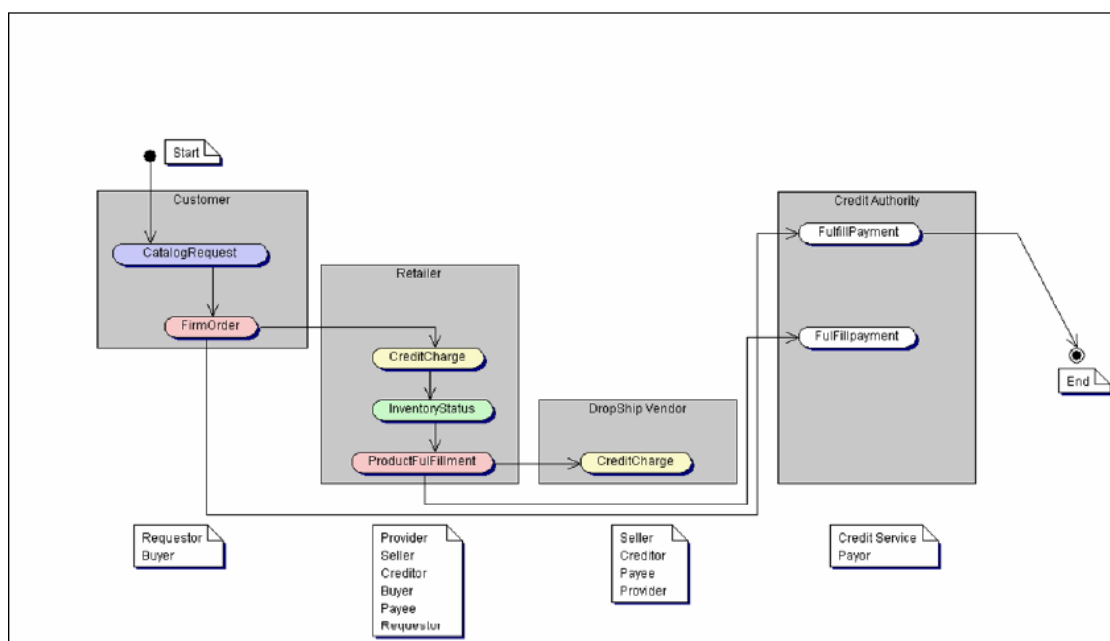


Figure 6. Representation of the “DropShip” multi-party collaboration with a UML activity diagram.

All binary collaboration in the example feature only one business transaction activity except two of them: Credit Charge and Product Fulfillment. These binary collaborations are represented on figure 7. with the same convention.

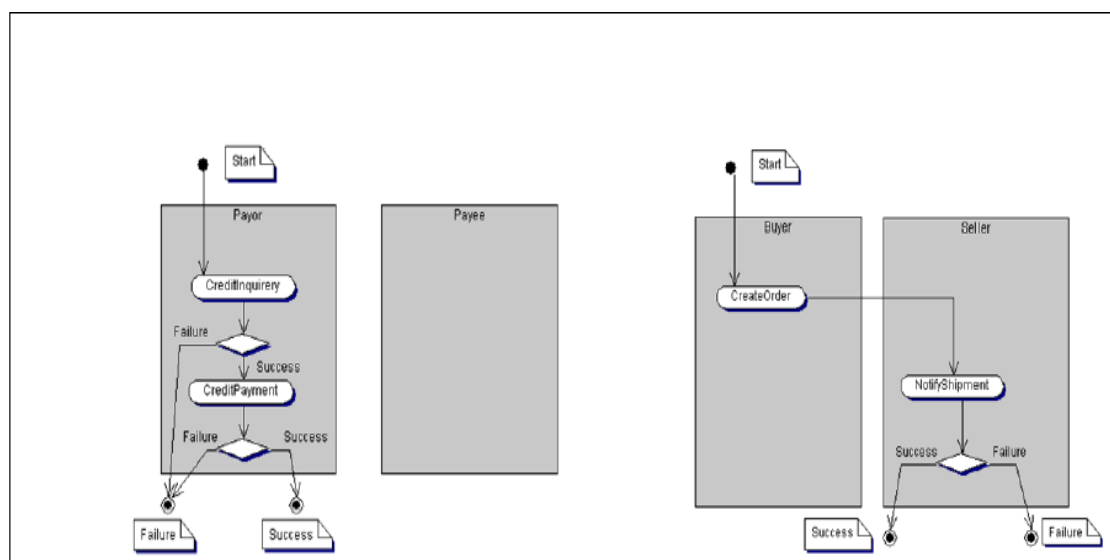


Figure 7. Representation of the “CreditCharge” and “ProductFulfillment” binary collaborations

Figure 8. features all the binary collaboration definitions of the example (between abstract roles and business partner roles).

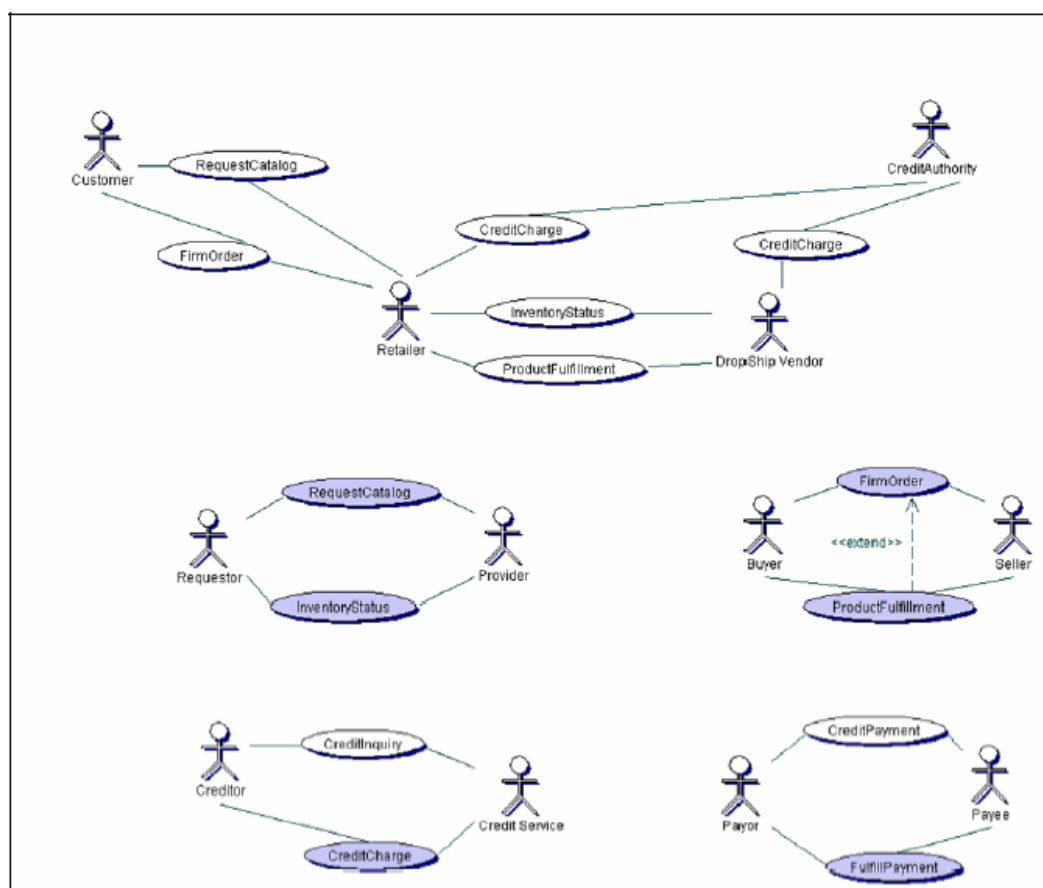


Figure 8. Multi-party and binary collaboration definitions of the example.

The complete XML is provided in Appendix A.

5.12.1 Packages and Includes

All elements of this specification are defined within the context of a package. Packages may contain other package, therefore defining a hierarchy of packages. The identification scheme for BPSS elements or documents should be specified and considered at design time by the user while producing the BPS.

A package defines the qualified name of the elements inside it. You cannot have two model elements with the same name within the same package. Model element names can be

658 qualified with the package using the Java notation:
659 `org.ebxml.transaction.order.ProcessPurchaseOrder`

660 Which means that the *ProcessPurchaseOrder* business transaction is defined within the
661 package *order*, which is itself, defined within the *transaction* package.

662 If a model element in package Order Entry needs to name something in a package called
663 Billing, it must include this package to make its elements visible to its own model elements.
664 Unlike an import, include requires that all model elements from the Billing package be fully
665 qualified. So if we want to designate the Invoice business document within the Order
666 Entry.Process Purchase Order transaction we need to refer to the Billing.Invoice document,
667 assuming it is defined in the Business Transaction.Billing package.

668

669 **5.12.2 Substitution Sets**

670 There is a requirement for Business specifications that are less coupled to technology and
671 business details, such as specific document formats and structures and timing parameters.
672 Substitution sets support the capability to take a generic business process and specialize it
673 for a specific use. For example, an ordering process may be very generic but a specific use
674 of that process may require specific document capabilities that go beyond the generic.

675 A substitution set is placed in the more specific process specification and replaces or makes
676 more explicit document definition references and attribute values. A Substitution Set is a
677 container for one or more AttributeSubstitution and/or DocumentSubstitution elements. The
678 entire SubstitutionSet specifies document or attribute values that should be used in place of
679 some documents and attribute values in an existing process specification.

5.12.3 Specify a Business Transaction and its Business Document Flow

Figure 9 shows a part of the BPSS metamodel that defines the concept of Business Transaction.

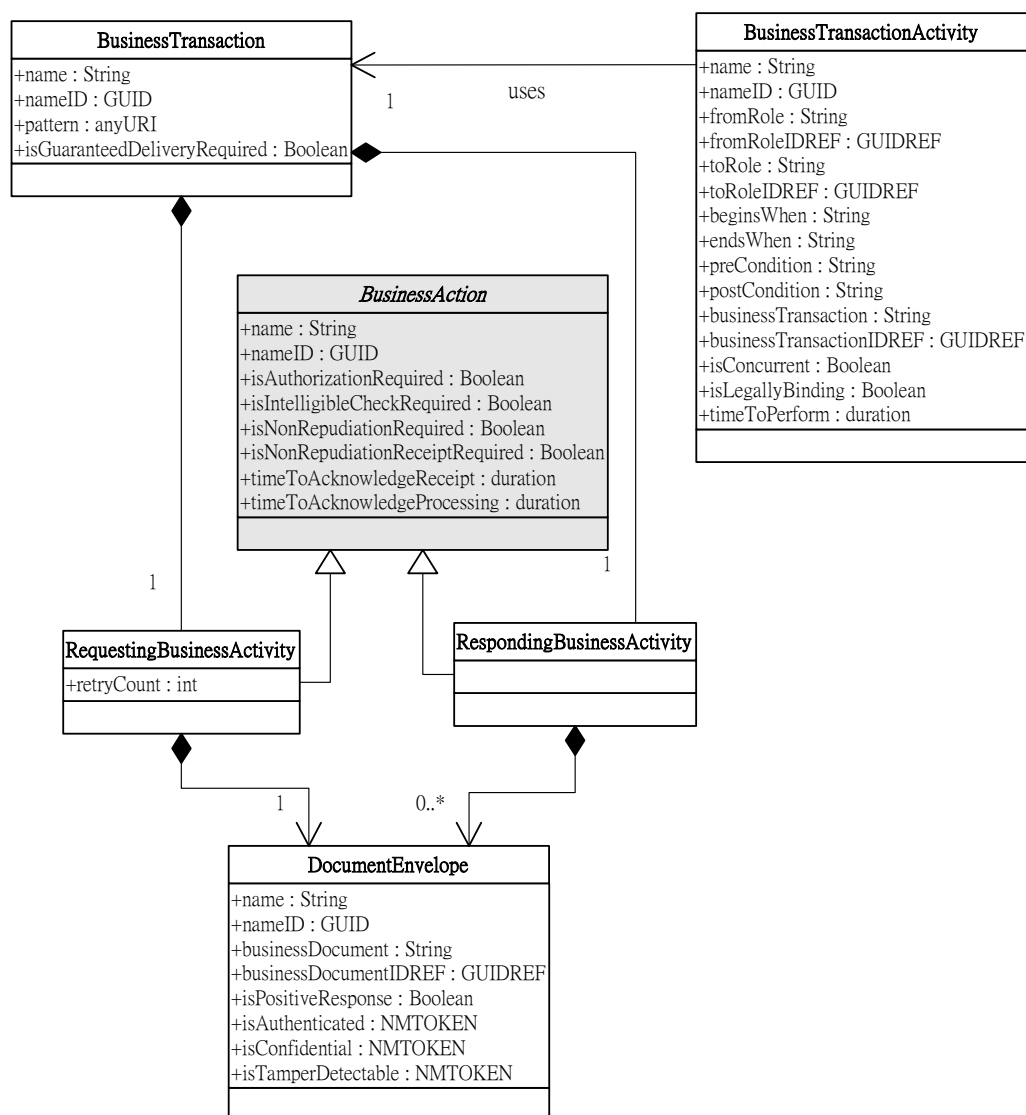


Figure 9. UML Diagram of a Business Transaction

5.12.3.1 Key Semantics of a Business Transaction

A Business Transaction is an atomic unit of work in a trading arrangement between two business partners.

689 A *Business Transaction* consists of a *Requesting Business Activity*, a *Responding Business*
 690 *Activity*, and one or two document flows between them. A *Business Transaction* may
 691 support one or more Business Signals that govern the use and meaning of
 692 acknowledgements.

693 Implicitly there is a requesting role performing the *Requesting Business Activity* and a
 694 responding role performing the *Responding Business Activity*. These roles become explicit
 695 when the transaction is used within a *Business Transaction Activity* within a *Binary*
 696 *Collaboration*. There is no need to make these roles more explicit such as buyer or seller. In
 697 particular some business transactions, for example “Cancel Purchase Order” may be used
 698 either way within the same binary collaboration definition as two different *Business*
 699 *Transaction Activities*.

700 There is always a Request document flow.

701 A Business Transaction definition specifies whether a response document is required. This
 702 type of business transactions is typically associated with the formation of contracts or
 703 agreements. A Business Transaction with a request only is typically used for notifications.

704 An abstract superclass, *Business Action*, is the holder of attributes that are common to both
 705 Requesting Business Activity and Responding Business Activity. This element is abstract,
 706 so it does not appear in ebXML BPSS instances.

707 Business signals acknowledging the document flow may be associated with each document
 708 flow. These acknowledgment signals are not specified explicitly however, two Business
 709 Transaction parameters specify whether the signals are required or not.

710 Figure 10 presents the possible Document Flows and business signals within a Business
 711 Transaction.

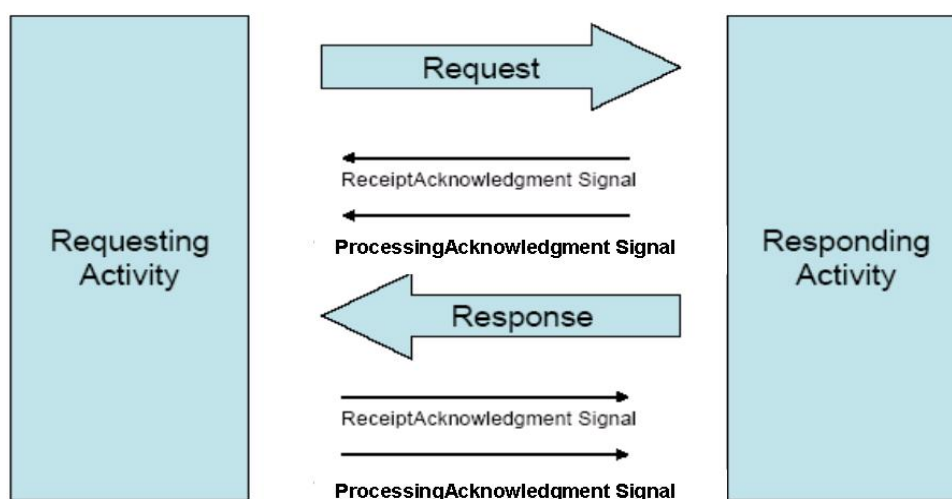


Figure 10. Possible document flows and signals and their sequence

714 These acknowledgment signals (a.k.a. Business Signals) are application level documents
715 that 'signal' the current state of the business transaction.

716 A Business Transaction element could be designated as one business transaction pattern.
717 There are six types of Business Transactions specified which are related to the Business
718 Transaction patterns defined by UMM:

- 719 ● Commercial Transaction : Known colloquially as the *Business Transaction pattern* .
720 Typically Commercial Transaction is a formal obligation between parties.
- 721 ● Query / Response : Used by a requester for an information query that responding
722 partner already has.
- 723 ● Request / Confirm : Used where an initiating partner requests confirmation about their
724 status with respect to previous obligations or a responder's business rules.
- 725 ● Request / Response : Used when an initiating partner requests information that a
726 responding partner already has and when the request for business information requires
727 a complex interdependent set of results.
- 728 ● Information Distribution : Represents an informal information exchange between
729 parties.
- 730 ● Notification : Used for notification of failure in line with a Commercial Transaction.
731 Represents a formal exchange between parties. Typically used to render a business
732 transaction as null and void.

733

| Business Transaction | Response | Request Receipt Ack | Request Processing Ack | Response Receipt Ack | Response Processing Ack |
|--------------------------|----------|---------------------|------------------------|----------------------|-------------------------|
| Commercial Transaction | Yes | Yes | Yes | Yes | Yes |
| Query / Response | Yes | Optional | No | Optional | No |
| Request / Confirm | Yes | Yes | No | Yes | No |
| Request / Response | Yes | Optional | No | Optional | No |
| Information Distribution | No | Yes | No | N/A | N/A |
| Notification | No | Yes | Optional | N/A | N/A |

734

735 5.12.3.2 Sample syntax

736 Here is a simple business transaction definition with just a requesting and responding
737 document flow:

```

738 <BusinessTransaction name="Catalog Request">
739   <RequestingBusinessActivity name="requestCatalog"
740     <DocumentEnvelope
741       businessDocument="Catalog Request"/>
742   </RequestingBusinessActivity>
743   <RespondingBusinessActivity name="sendCatalog">
744     <DocumentEnvelope
745       isPositiveResponse="true"
746       businessDocument="Catalog" />
747   </RespondingBusinessActivity>
748 </BusinessTransaction>

```

749 The pattern of a *Business Transaction* may be used to specify whether a Receipt
 750 Acknowledgement and/or an Processing Acknowledgement signal are required. If the
 751 *pattern* is not used, a non null value in the *timeToAcknowledgeReceipt* and
 752 *timeToAcknowledgeProcessing* will mean that these signals must be issued by the
 753 corresponding party. Business transaction protocol signals are independent from lower
 754 protocol and transport signals such as reliable messaging.

755 The Receipt Acknowledgement business signal, if used, signals that a message (Request or
 756 Response) has been properly received by the ebXML Business Service Interface software
 757 component. The property *isIntelligibleCheckRequired* allows partners to agree that a
 758 message should be confirmed by a Receipt acknowledgement only if it is also legible.
 759 Legible means that it has passed structure/ schema validity check. The content of the receipt
 760 and the legibility of a message (if required) are reviewed *prior* to the processing of the
 761 Business Document or the evaluation of condition expressions in the message's business
 762 documents or document envelope.

763 The Processing Acknowledgement business signal, if used, signals that the message
 764 received (Request or Response) has been accepted for business processing by the receiving
 765 application, or a receiving business application proxy. This is the case if the contents of the
 766 message's business documents and document envelope have passed a business rule validity
 767 check. These business rules are not necessarily specified as part of the collaboration. The
 768 state of each party is considered to be aligned when the receiving application (in general
 769 unknown to the other party) has signaled, *via* the BSI and an Processing Acknowledgement,
 770 that the business document has been successfully processed. Note that this
 771 acknowledgement is non-substantive, and simply indicate that the receiving party has
 772 reached a satisfactory state. If for any reason, the application could not process the business
 773 document, the sending party should be notified via a negative Processing Acknowledgement
 774 signal such that it can transition to a meaningful "internal" business state. For instance, a
 775 Purchase Order could not be considered in the "sent" state, unless the other party had sent
 776 the corresponding Processing Acknowledgment. The substantive response would come after
 777 the signal indicating whether the order had been Accepted or Rejected.

778 Failure to send either signal, when *required* (by specifying a timeout value in
 779 *timeToAcknowledgeReceipt* or *timeToAcknowledgeProcessing*), will result in the
 780 transaction being null and void, and therefore will prevent to reach any "success" end state
 781 (protocol or business) that would have depended on receipt of a business document

782 satisfying the associated *timeToPerform*. In order for a business transaction activity instance
 783 to reach a “success” state at run-time, the following things would need to happen:

- 784 ● no timeout would have occurred (signals or response)
- 785 ● no signal can have a negative content
- 786 ● the response document sent to the requestor must be marked as *isPositiveResponse* =
 787 ‘true’ in the ebXML BPSS instance that specifies the business collaboration

788 Conversely, if all signals are positive and sent and received on time, the transaction will be
 789 successful from a protocol perspective.

790 The *isPositiveResponse* attribute of a *DocumentEnvelope* is not part of the business
 791 transaction protocol and therefore does not impact the protocol success or failure of a
 792 collaboration. If the *DocumentEnvelope* received as a response is specified with the
 793 *isPositiveResponse*=false (at design time) the business transaction will end in a business
 794 failure state. The choreography of the binary collaboration may use this information to
 795 execute corresponding transitions or stop the collaboration altogether. Note that this
 796 attribute is optional and some document envelope may neither be positive or negative
 797 (consider for instance the case of a partial acceptance on a purchase order, where only a few
 798 line items are refused, or a back order response). In this case, the business transaction
 799 activity is considered successful, again after it has reached a protocol success state.

800 The *isGuaranteedMessageDeliveryRequired* refers to the underlying messaging service
 801 used to implement the business transaction protocol. The business transaction protocol is
 802 designed to achieve state alignment between both parties involved in the transaction and
 803 signals to the sending party the successful processing of the business documents, request or
 804 response, by the receiving application, whatever it might be. However, to achieve this result,
 805 the business transaction protocol shall be implemented on top of a reliable messaging
 806 service that provides guaranteed message delivery at the transport level. If the sending party
 807 was not guaranteed that its message or in particular signal reached the intended recipient, it
 808 could never be sure that the other party state is aligned with its own state. Since a signal
 809 structure is fixed there is no ambiguity about the BSI processing it and understanding its
 810 meaning provided you know that it reached its destination, unlike a request or response
 811 which could have an invalid structure or content. In the case where the business transaction
 812 does not need to guarantee processing by the receiving application this condition can be
 813 relaxed and regular messaging services may be used.

814 Note that we can only guarantee the successful synchronization of state between two parties
 815 if reliable messaging is used and if the business transaction is defined to use the request and
 816 response Processing acknowledgement signals, which guarantee that the corresponding
 817 business documents were processed by the respective applications.

818 5.12.3.3 Sample syntax

819 Here is a slightly more complex transaction with two document flows and three business
820 signals.

821 The request requires both receipt and Processing acknowledgement, the response requires
822 only receipt acknowledgement. “P2D” is a W3C Schema syntax adopted from the ISO 8601
823 standard and means Period=2 Days. P3D means Period=3 Days, P5D means Period=5 Days.
824 These periods are all measured from original sending of request.

```

825 <BusinessTransaction
826     name="CreateOrder"
827     nameID="122A3DD33"
828     isGuaranteedDeliveryRequired="true">
829     <RequestingBusinessActivity
830         name="sendOrder"
831         nameID="122A3E833"
832         isNonRepudiationReceiptRequired="false"
833         isNonRepudiationRequired="false"
834         timeToAcknowledgeProcessing="P1H"
835         timeToAcknowledgeReceipt="P1H">
836         <DocumentEnvelope
837             businessDocument="Purchase Order"
838             businessDocumentIDREF="122A3F613"/>
839         </RequestingBusinessActivity>
840         <RespondingBusinessActivity
841             name="sendPOAcknowledgement"
842             nameID="122A3E863"
843             isNonRepudiationReceiptRequired="false"
844             isNonRepudiationRequired="false"
845             timeToAcknowledgeReceipt="P1D">
846             <DocumentEnvelope
847                 isPositiveResponse="false"
848                 businessDocument="Reject Order"
849                 businessDocumentIDREF="122A3F8E3"/>
850             <DocumentEnvelope
851                 isPositiveResponse="true"
852                 businessDocument="Accept Order"
853                 businessDocumentIDREF="122A3F6C3"/>
854             </RespondingBusinessActivity>
855     </BusinessTransaction>

```

856 Note that duration are expressed using the standard duration type from the W3C's XML
857 Schema specification. For instance “P1D” means that we are specifying a “period” of 1 day.

858 5.12.3.4 Specifying Business Document flows

859 Request document flows and response document flows contain Business Documents that
860 pertain to the *Business Transaction* request and response. The model for this is shown in
861 Figure 11. Business Documents have varying structures. Business signals, however always
862 have the same structure, defined once and for all as part of the ebXML *Business Process*
863 *Specification Schema* technical specification.

864 A document flow is not modeled directly. Rather it is modeled indirectly as a *Document*
 865 *Envelope* sent by one role and received by the other. The *Document Envelope* is always
 866 associated with one *Requesting Business Activity* or one *Responding Business Activity* to
 867 specify the flow.

868 Document Envelopes are named. There is always only one named Document Envelope for a
 869 Requesting Activity. There may be zero, one, or many mutually exclusive, named
 870 Document Envelopes for a Responding Activity. For example, the Response Document
 871 Envelopes for a purchase order transaction might be named PurchaseOrderAcceptance,
 872 PurchaseOrderDenial, and PartialPurchaseOrderAcceptance. In the actual execution of the
 873 purchase order transaction, however, only one of the defined possible responses will be sent.

874 Each Document Envelope carries exactly one primary Business Document.

875 A Document Envelope can optionally have one or more attachments, all related to the
 876 primary Business Document. The document and its attachments in essence form one
 877 transaction in the payload in the ebXML Message Service message structure.

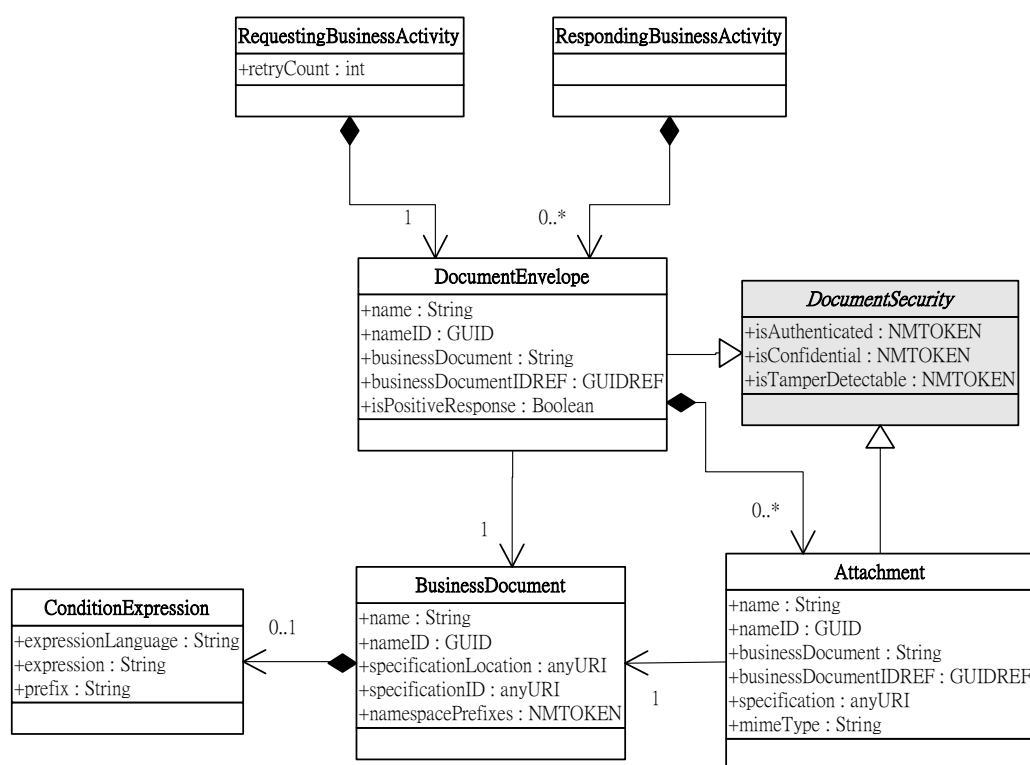


Figure 11. UML Diagram of document flow

880 5.12.3.5 Sample syntax

881 This example shows a business transaction with one request and two possible responses, a
 882 success and a failure. The response has an attachment. All the Business Documents are fully
 883 qualified with the schema name.

```

884 <BusinessDocument
885     name="Credit Request"
886     nameID="122A3F613C "
887     specificationLocation="http://.../creditRequest.xsd"
888     specificationID="http://... /creditRequest.xsd"
889     namespacePrefixes="fix">
890 </BusinessDocument>
891
892 <!-- The following two documents refer to the same physical document, however, by their content as evaluated at
893 run-time, they are logically different -->
894 <BusinessDocument
895     name="Credit Denied"
896     nameID="122A3F8E3"
897     specificationLocation="http://.../creditResponse.xsd"
898     specificationID="http://.../creditResponse.xsd"
899     namespacePrefixes="fix">
900     <ConditionExpression
901         expressionLanguage="XPATH 1.0"
902         expression="//@CreditResponse='denied'"
903     prefix="fix"/>
904 </BusinessDocument>
905
906 <BusinessDocument
907     name="Credit Approved"
908     nameID="122A3F6C3"
909     specificationLocation="http://.../creditResponse.xsd"
910     specificationID="http://.../creditResponse.xsd"
911     namespacePrefixes="fix">
912     <ConditionExpression
913         expressionLanguage="XPATH 1.0"
914         expression="//@CreditResponse='approved'"
915     prefix="fix"/>
916 </BusinessDocument>
917
918 <BusinessDocument
919     name="Credit Rating"
920     nameID="122A3F8E4"
921     specificationID="http://.../creditRating.id">
922 </BusinessDocument>
923
924 <BusinessTransaction
925     name="Check Credit"
926     nameID="122A3DD33"
927     isGuaranteedDeliveryRequired="true">
928     <RequestingBusinessActivity
929         name="checkCredit"
930         nameID="122A3E833"
931         isAuthorizationRequired="true"
932         isIntelligibleCheckRequired="true"
933         isNonRepudiationReceiptRequired="true"
934         isNonRepudiationRequired="true"
  
```

```

935         timeToAcknowledgeProcessing=" PT30S"
936         timeToAcknowledgeReceipt=" PT10S">
937         <DocumentEnvelope
938             isAuthenticated="persistent"
939             isConfidential="persistent"
940             isTamperDetectable="persistent"
941             businessDocument=" Credit Request"
942             businessDocumentIDREF="122A3F613C"/>
943     </RequestingBusinessActivity>
944
945     <RespondingBusinessActivity
946         name="confirmCredit"
947         nameID="122A3E863"
948         isAuthorizationRequired="true"
949         isIntelligibleCheckRequired="true"
950         isNonRepudiationReceiptRequired="true"
951         isNonRepudiationRequired="true"
952         timeToAcknowledgeReceipt="PT10S">
953         <DocumentEnvelope
954             isPositiveResponse="false"
955             isAuthenticated="persistent"
956             isConfidential="persistent"
957             isTamperDetectable="persistent"
958             businessDocument="Credit Denied"
959             businessDocumentIDREF="122A3F8E3"/>
960         <DocumentEnvelope
961             isPositiveResponse="true"
962             isAuthenticated="persistent"
963             isConfidential="persistent"
964             isTamperDetectable="persistent"
965             businessDocument="Credit Approved"
966             businessDocumentIDREF="122A3F6C3">
967             <Attachment
968                 name="Credit Report"
969                 mimeType="XML"
970                 businessDocument="Credit Rating"
971                 businessDocumentIDREF="122A3F8E4"
972                 isConfidential="none"
973                 isTamperDetectable="none"
974                 isAuthenticated="none">
975             </Attachment>
976         </DocumentEnvelope>
977     </RespondingBusinessActivity>
978 </BusinessTransaction>

```

See section 5.13.5 for a discussion on document security parameters.

5.12.4 Specify a Binary Collaboration

Figure 12 shows part of the metamodel of a binary collaboration.

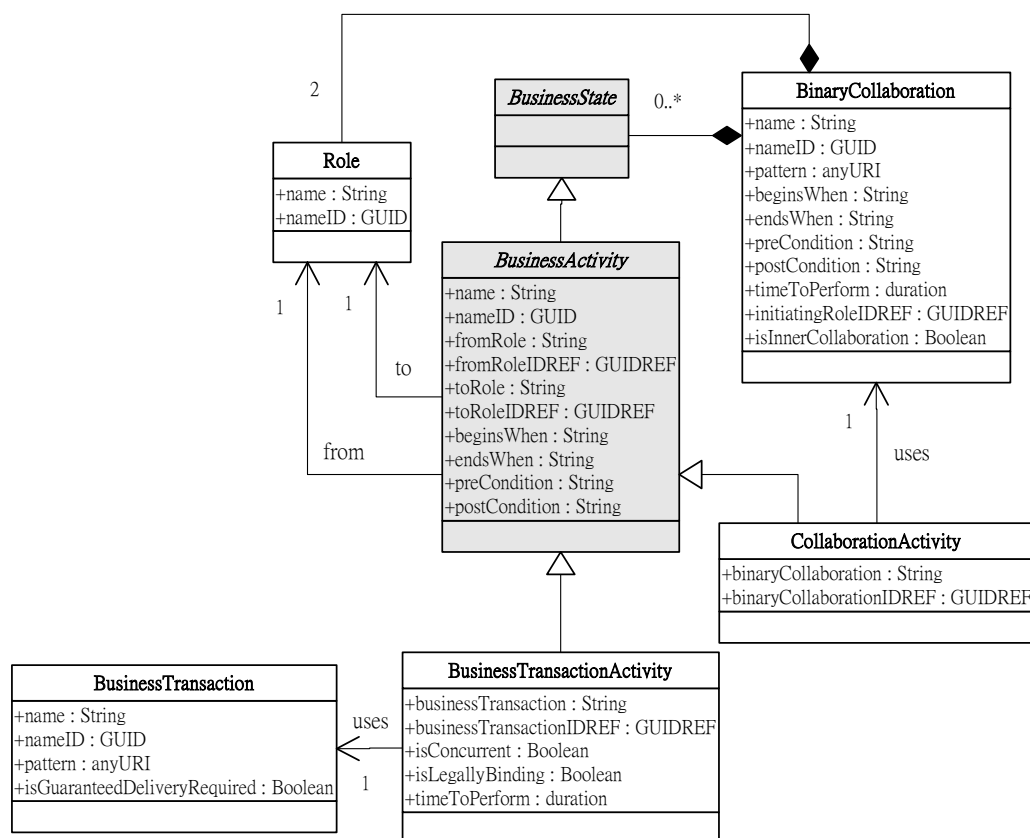


Figure 12. UML Diagram of a Binary Collaboration

5.12.4.1 Key Semantics of a Binary Collaboration

A *Binary Collaboration* is always defined between two roles. One of the roles is initiating the collaboration. This is the role, which sends the first message (i.e. Request) of the first *Business Transaction Activity*. This attribute is used to “bind” the roles of an inner *Collaboration Activity* to the parent *Binary Collaboration* roles. Even if this role is not known until run-time, we can still specify a “logical” initiating role. In that case, the initiating role of the parent binary collaboration definition will be bound to the initiating role of the inner binary collaboration definition.

It is critical that the *Role nameID* be unique with respect to a *Binary Collaboration* definition even if role names are identical for two *Binary Collaboration* definitions. This means that two binary collaboration may never have the same physical role, but share a “logical” role.

997 A *Binary Collaboration* consists of one or more Business Activities. These Business
 998 Activities are always conducted **between** the two Roles of the Binary Collaboration. For
 999 each activity one of two roles is assigned to be the *initiatingRole* (from) and the other to be
 1000 the *respondingRole* (to). This is irrespective who initiated the binary collaboration.

1001 A *Business Activity* can be either a *Business Transaction Activity* or a *Collaboration Activity*.

1002 A *Business Transaction Activity* is the performance of a *Business Transaction*. Business
 1003 Transaction definitions can be associated to any number of Business Transaction Activity
 1004 elements. This means that the same *Business Transaction* can be performed by multiple that
 1005 the same *Business Transaction* can be performed by multiple *Business Transaction*
 1006 *Activities* in different *Binary Collaborations*, or by multiple *Business Transaction Activities*
 1007 in the same *Binary Collaboration*, sometimes with opposite roles. For instance a “Cancel
 1008 Purchase Order” *Business Transaction* could be used by two Business Transaction
 1009 Activities, which can be performed by either party, meaning that after a purchase order has
 1010 been accepted, either party could cancel it (for a certain period of time) using the same
 1011 business document interchange.

1012 A *Collaboration Activity* is the performance of a *Binary Collaboration*, within another
 1013 *Binary Collaboration*. *Binary Collaboration definitions* are re-useable relative to
 1014 Collaboration Activity. The same *Binary Collaboration* can be performed by multiple
 1015 *Collaboration Activities* in different *Binary Collaborations*, or by multiple *Collaboration*
 1016 *Activities* in the same *Binary Collaboration*. A binary collaboration definition may be
 1017 restricted to be an “inner collaboration” only via the the Boolean attribute
 1018 *isInnerCollaboration*. In this case, the binary collaboration definition can only be initiated
 1019 as part of a *Collaboration Activity* and cannot be initiated by itself.

1020 *Business Transaction Activity* and *Collaboration Activity* may define business rules with the
 1021 *beginsWhen*, *endsWhen*, *preCondition* and *postCondition* attributes. These attributes do not
 1022 have a specific syntax as part of this specification, so the current type is string. Because
 1023 these expressions cannot be generally executed by an ebXML infrastructure, in the current
 1024 release of the ebXML BPSS technical specification, they are considered to have a
 1025 “documentation” purpose. In particular they cannot be used to specify any part of the
 1026 choreography of the collaboration. In future releases they will play a role along with
 1027 transitions and pseudo-states. The semantics of *beginsWhen* and *endsWhen* indicate that the
 1028 corresponding business activity needs to be started or ended as soon as the expression in the
 1029 attribute value is true. *PreConditions* and *postConditions* indicate that the corresponding
 1030 business activity may start only if the corresponding expressions are true.

1031 When performing a *Binary Collaboration* within a *Binary Collaboration* there is an implicit
 1032 relationship between the roles at the two levels. Assume that *Binary Collaboration X* is
 1033 performing *Binary Collaboration Y* through Collaboration Activity Q. Binary Collaboration
 1034 X has the following roles: Customer and Vendor. In Collaboration Activity Q we assign
 1035 Customer to be the initiator, and Vendor to be the responder. Binary Collaboration Y has
 1036 the following roles: Buyer and Seller and a Business Transaction Activity where Buyer is
 1037 the initiator and Seller the responder. We have now established a role relationship between

1038 the roles Customer and Buyer because they are both initiators in activities in the related
1039 performing and performed Binary Collaborations.

1040 Since a *Business Transaction* is atomic in nature, the performing of a single *Business*
1041 *Transaction* through a *Business Transaction Activity* is also atomic in nature. If the desired
1042 semantic is not atomic, and then the task should be split over multiple transactions. For
1043 instance if it is desired to specify several partial acceptances of a request, then the request
1044 should be specified as one transaction within a binary collaboration and the partial
1045 acceptance(s) as separate transactions.

1046 The CPA/CPP Specification allows that parties agree upon a Collaboration Protocol
1047 Agreement (CPA) in order to transact business. A CPA may associate itself with a specific
1048 *Binary Collaboration*. Thus, all *Business Transactions* performed between two parties
1049 should be referenced through *Business Transaction Activities* contained within a *Binary*
1050 *Collaboration*.

1051 **5.12.4.2 Sample syntax**

1052 Here is a simple Binary Collaboration using one of the Business Transactions defined above:

```

1053 <BinaryCollaboration
1054     name="Firm Order"
1055     nameID="122A38D93"
1056     initiatingRoleIDREF="122A38DA3"
1057     timeToPerform="P1D">
1058     <Role
1059         name="buyer"
1060         nameID="122A38DA3"/>
1061     <Role
1062         name="seller"
1063         nameID="122A38DA5"/>
1064     <Start
1065         toBusinessState="Place Order"
1066         toBusinessStateIDREF="122A39C23" />
1067     <BusinessTransactionActivity
1068         name="Place Order"
1069         nameID="122A39C23"
1070         businessTransaction="Create Order"
1071         businessTransactionIDREF="122A3DD33"
1072         fromRole="buyer"
1073         fromRoleIDREF="122A38DA3"
1074         toRole="seller"
1075         toRoleIDREF="122A38DA5"
1076         isConcurrent="true"
1077         isLegallyBinding="false"
1078         timeToPerform="P2H"/>
1079     <Failure
1080         fromBusinessState="Place Order"
1081         fromBusinessStateIDREF="122A39C23"
1082         conditionGuard="AnyProtocolFailure"/>
1083     <Success
1084         fromBusinessState="Place Order"
1085         fromBusinessStateIDREF="122A39C23"

```



```
1086         conditionGuard="BusinessSuccess |  
1087         BusinessFailure"/>  
1088     </BinaryCollaboration>  
1089  
1090
```

5.12.5 Specify a MultiParty Collaboration

Figure 13 shows the metamodel a multiparty collaboration

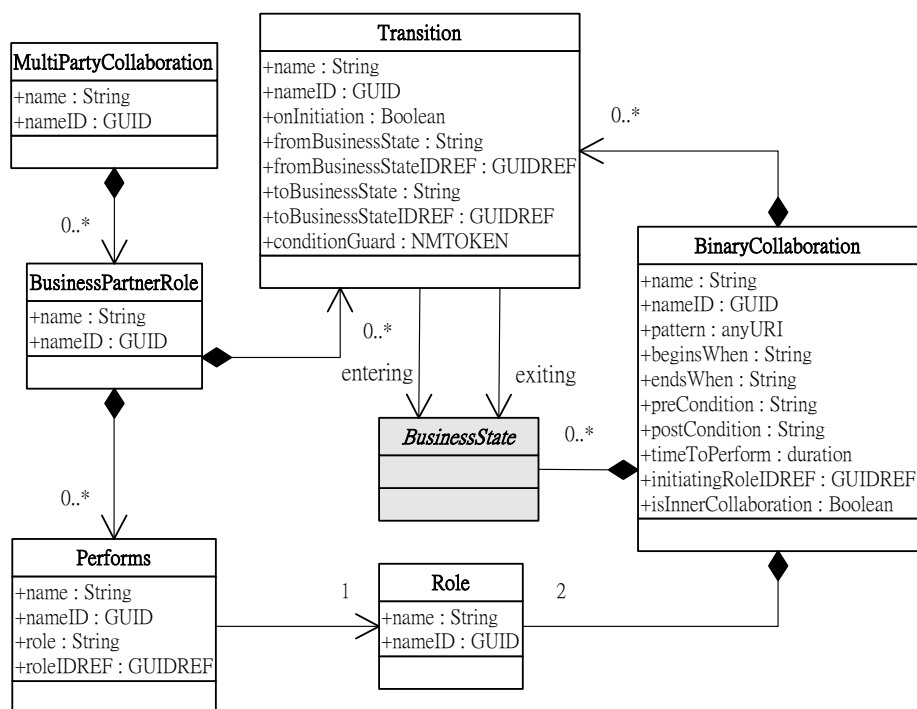


Figure 13. UML Diagram of a MultiParty Collaboration

5.12.5.1 Key Semantics of a Multiparty Collaboration

A Multiparty Collaboration is specified as a synthesis of Binary Collaborations definitions.

A Multiparty Collaboration is defined as a list of “*Business Partner Roles*”.

Each *Business Partner Role* *Performs* one or more *Role* several binary collaboration definitions. Note that the Binary Collaboration to Role relationship is navigable, which means that a *Performs* element uniquely identify a Binary Collaboration and a Role within this Binary Collaboration definition. It is often the case that a business partner (role) plays several “roles” in a multi-party collaboration. For instance, a distributor role, would play both the roles of buyer and seller in a purchasing collaboration involving a customer (buyer), distributor (seller, buyer) and a manufacturer (seller).

1106 This association between a *Business Partner Role* and a specific *Role* in a specific *Binary*
 1107 *Collaboration* is specified by the *Performs* element and is what constitute the synthesis of
 1108 *Binary Collaborations* into *Multiparty Collaborations*.

1109 Each binary pair of trading partners may be subject to one or more distinct CPAs.

1110 Within a Multiparty Collaboration, you may choreograph transitions between Business
 1111 Transaction Activities in different Binary Collaborations, as described below.

1112 5.12.5.2 Sample syntax

1113 Here is a simple Multiparty Collaboration which involves 3 parties (Requester, Intermediary
 1114 and Provider) performing the simple roles of “sender” and “receiver”. B is considered an
 1115 intermediary. The same binary collaborations is executed amongst the parties: one between
 1116 the Requester and the Intermediary and the other between the intermediary and the Provider.
 1117 In this case, the Intermediary plays both roles of the Binary Collaboration.

```

1118 <MultiPartyCollaboration name="DropShip">
1119   <BusinessPartnerRole name="Customer">
1120     <Performs role="requestor" roleIDREF="1122B1"/>
1121     <Performs role="buyer" roleIDREF="1122B2"/>
1122     <Transition
1123       fromBusinessState="Catalog Request"
1124       toBusinessState="Create Order"/>
1125   </BusinessPartnerRole>
1126   <BusinessPartnerRole name="Retailer">
1127     <Performs role="provider" roleIDREF="2211A1"/>
1128     <Performs role="seller" roleIDREF="1122B3"/>
1129     <Performs role="creditor" roleIDREF="9122B1"/>
1130     <Performs role="buyer" roleIDREF="1122B2"/>
1131     <Performs role="payee" roleIDREF="6122B1"/>
1132     <Performs role="requestor" roleIDREF="1122B1"/>
1133     <Transition
1134       fromBusinessState="Create Order"
1135       toBusinessState="Check Credit"/>
1136     <Transition
1137       fromBusinessState="Check Credit"
1138       toBusinessState="Credit Payment"/>
1139   </BusinessPartnerRole>
1140   <BusinessPartnerRole name="DropShip Vendor">
1141     <Performs role="seller" roleIDREF="1122B3"/>
1142     <Performs role="payee" roleIDREF="6122B1"/>
1143     <Performs role="creditor" roleIDREF="9122B1"/>
1144     <Performs role="provider" roleIDREF="2211A1"/>
1145   </BusinessPartnerRole>
1146   <BusinessPartnerRole name="Credit Authority">
1147     <Performs role="credit service" roleIDRef="8122B1"/>
1148     <Performs role="payor" roleIDREF="7122B1"/>
1149   </BusinessPartnerRole>
1150 </MultiPartyCollaboration>

```

1151 Note that the role value links the corresponding Binary Collaboration definition to this
 1152 Multiparty Collaboration definition.

5.12.6 Specify a Choreography

Figure 14 shows the metamodel of a choreography.

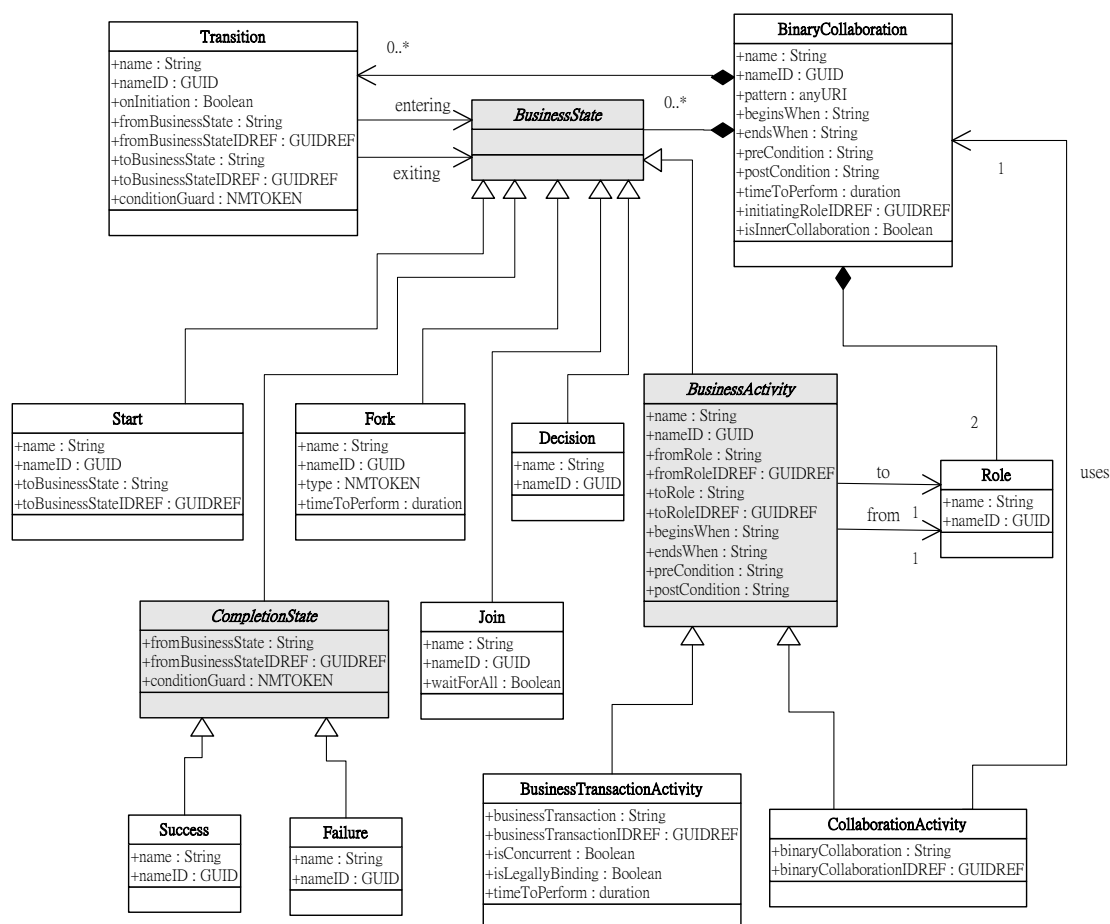


Figure 14. UML Diagram of a Choreography

5.12.6.1 Key Semantics of a Choreography

A Choreography is an ordering of Business Activities within a Binary Collaboration. The purpose of a Choreography is to specify which Business Transaction Activity and/or Collaboration Activity should happen at any point in time. As a result, the specification of choreography definition and the business transaction protocol defines unambiguously which message (DocumentEnvelope or Signal) is expected by any of the parties at any point in time.

The choreography is specified in terms of Business States, and transitions between those Business States. When such a Business State is a Business Activity, a transition happens between the Business Activity completion state and the start state of the following Business Activity. When a transition is validated, it does not mean that the target Business Activity

1169 would start immediately. Instead, it means that the Business Activity is “enabled” and the
 1170 initiating party may now send the request whenever appropriate, provided that it remains
 1171 within the *timeToPerform* of the binary collaboration.

1172 A Business Activity is an abstract kind of Business State. Its two subtypes Business
 1173 Transaction Activity and Collaboration Activity are concrete Business States. The business
 1174 collaboration can be said to be in the state of performing a given business activity. Once a
 1175 business activity complete a transition from this business activity is navigated to another
 1176 business activity or pseudo-state. A message shall either initiate a collaboration or advance
 1177 its state.

1178 There are a number of auxiliary kinds of Business States that facilitate the choreographing
 1179 of Business Activities. These include a Start state, a Completion state (which comes in a
 1180 Success and Failure flavor), a Fork state, a Join state and a Decision state. These are all
 1181 equivalent to diagramming artifacts on a UML activity diagram, however, the semantics are
 1182 not exactly the same. An XOR value in the type attribute of a fork means that only one
 1183 Business State of the fork will be allowed to be reached. All the other will become invalid
 1184 as soon as one of the business state is reached (e.g. a Business Transaction Activity starts).
 1185 An OR value will mean that any business activity pointed to by a transition coming from the
 1186 fork might be initiated. These business activities may occur in parallel. Note that it is not
 1187 important to specify the order in which condition expression on a transition coming from a
 1188 fork will be evaluated. It is merely the order in which the request of the business transaction
 1189 activities will arrive that will determine the order in which the condition expression need to
 1190 be evaluated. A fork has a *timeToPerform* attribute. At the end of this time interval, the
 1191 state of the Binary Collaboration will automatically be moved to its corresponding join.
 1192 This feature is useful in cases where the business activities are optional. For instance a
 1193 Cancel Purchase Order and Change Purchase Order business transaction activity could be
 1194 defined as part of a Fork/Join control block. However, most often none of these activity
 1195 would happen. If any given Business Transaction Activity within the Fork/Join pair is has
 1196 not reached its completion state, the BSI will generate a corresponding timeout exception.
 1197 As a well formed rule, the *timeToPerform* of a fork can not be less that any *timeToPerform*
 1198 of its business activities. The *waitForAll* attribute of the join will indicate that all transitions
 1199 coming into the join shall be executed in order for the collaboration to reach the join
 1200 pseudo-state (AND-join), by default, the join is an AND-join. When this parameter is set to
 1201 false, it is an OR-join. The BSI will generate a timeout exception if an OR-join is reached
 1202 while a Business Activity has not reached its completion state. The semantics of fork and
 1203 join are such that for instance a fork may be defined without a corresponding join. In this
 1204 case, the *timeToPerform* attribute shall not be used. It must only be used in the case where
 1205 all outgoing transitions from the fork have incoming transitions to the join.

| Fork | Join | Comments |
|------|----------------------|--|
| OR | WaitforAll (true) | This models the behavior of an AND-fork and AND-Join |
| OR | WaitforAll | If timeout is null, should rather use XOR as the join will happen on the first transition reaching |

| | | |
|------------------|-----------------------|---|
| | (false) | the join state |
| XOR | WaitforAll (true) | This combination is forbidden (would lead to a dead lock) |
| XOR | WaitforAll (false) | Only one path between the fork and join will be allowed to happen |
| timeToPerform >0 | Any value | The join happens when timeToPerform is reached. |

Transitions can originate from Business Transaction Activities or Collaboration Activities within a Binary Collaboration, or from Binary Collaborations within a Multiparty Collaboration. Guards can gate transitions. Guards refer to the status of the Business Transaction Activity from which the transition originates. The guard values include: ProtocolSuccess, AnyProtocolFailure, RequestReceiptFailure, RequestProcessingFailure, ResponseReceiptFailure, ResponseProcessingFailure, SignalTimeOut, ResponseTimeOut, Failure, BusinessSuccess, BusinessFailure and Success. Transitions may also have a condition expression element. A ConditionExpression element has a language attribute, which specifies in which language the predicate is written. We do not limit the type and number of languages a BSI may support. However, for compliance, a BSI is required to support at least the XPath language, as well as the DocumentEnvelopeNotation. An XPath expression may involve the content of any DocumentEnvelope received prior to the transition within the scope of the current binary collaboration instance. The DocumentEnvelopeNotation is simply defined as the name or ID of a document envelope.

The Success and Failure elements represent an aggregation of a state and a transition to this particular state. This transition like regular transitions can be guarded by a conditionGuard. The conditionGuard can be used to indicate that a binary collaboration ends in success or failure based on the fact that the last business transaction activity response is a business document of a particular type, or based on the content of the response. It is important to note that the success or failure of the collaboration does not affect the success or failure of the individual business transaction activities, which compose the binary collaboration. In particular, the nature of the commitments is not changed when the collaboration ends in a specific state. The success or failure of a collaboration is rather an indication, which can be reported on, or acted upon to initiate other collaborations. If several completion states are specified within a collaboration definition, the business collaboration run-time instance state is “complete” as soon as one of the completion state is reached. It is the responsibility of the designer to ensure that all completion states are mutually exclusive and that once one of them is reached there are no further Business Activity open. A timeout exception will be generated by the BSI in such a case.

A Transition can also be used to create nested BusinessTransaction- Activities. A nested BusinessTransactionActivity is enabled when a transition flows from a parent BTA to the nested BTA and this transition is marked onInitiation = ‘true’. In this case, the transition is enabled after the receipt of the request in the parent transaction but after the request has been acknowledged appropriately if applicable. At this point, the second activity is

performed before returning sending the response to the original requestor. No “return” transition is specified. If more than one transaction ought to be executed they must be specified within a collaboration activity. There are no possible outgoing transitions from a nested activity unless it is associated to an exception. If the activity terminates normally, the thread of control is handed back to the parent activity. The flag ‘onInitiation’ in Transition is used for this purpose. Nested *Business Transaction Activity* are often found within a multiparty collaboration. In essence a Role in one Binary Collaboration receives a request, then turns around and becomes the requestor in other Binary Collaboration before coming back and sending the response in the first Binary Collaboration.

isConcurrent is a parameter that governs the flow of transactions. Unlike the security and timing parameters it does not govern the internal flow of a transaction, rather it determines whether at run-time multiple instances of that business transaction activity can be ‘open’ at the same time within any collaboration instance performed between any two partners. As a result, when *isConcurrent* is set to false, the BSIs of each party are responsible for serializing these business transaction activities.

5.12.6.2 Sample syntax

Here is the same Binary Collaboration as used before, with choreography added at the end. There is a transition between the two, a start and two possible outcomes of this collaboration, success and failure:

```
<BinaryCollaboration
  name="Product Fullfillment"
  nameID="122A38D93"
  Role="233A38DA3"
  timeToPerform="P3D">
  <Role
    name="buyer"
    nameID="122A38DA3"/>
  <Role
    name="seller"
    nameID="122A38DA5"/>
  <Start
    toBusinessState="Create Order"
    toBusinessStateIDREF="122A39C23"/>
  <BusinessTransactionActivity
    name="Create Order"
    nameID="122A39C23"
    businessTransaction="Create Order"
    businessTransactionIDREF="122A39C24"
    fromRole="buyer"
    fromRoleIDREF="122A38DA3"
    toRole="dealer"
    toRoleIDREF="122A38DA5"
    isConcurrent="true" isLegallyBinding="false"
    timeToPerform="P1H"/>
  <BusinessTransactionActivity
    name="Notify Shipment"
    nameID="122A39CA3"
    businessTransaction="Notify Shipment"
    businessTransactionIDREF="122A39CA4"
```

```

1289         fromRole="seller"
1290         fromRoleIDREF="122A38DA3"
1291         toRole="buyer"
1292         toRoleIDREF="122A38DA3"
1293         isConcurrent="true" isLegallyBinding="false"
1294         timeToPerform="P2D"/>
1295     <Success
1296         fromBusinessState="Test Success"
1297         fromBusinessStateIDREF="54654B789"
1298         conditionGuard="Success"/>
1299     <Failure
1300         fromBusinessState="Test Success"
1301         fromBusinessStateIDREF="54654B789"
1302         conditionGuard="AnyProtocolFailure |
1303         BusinessFailure"/>
1304     <Failure
1305         fromBusinessState="Test Order Accepted"
1306         fromBusinessStateIDREF="54654B567"
1307         conditionGuard="AnyProtocolFailure |
1308         BusinessFailure">
1309         <ConditionExpression
1310             expressionLanguage=
1311             "DocumentEnvelopeNotation"
1312             conditionExpression=
1313             "Reject Order"/>
1314     </Failure>
1315     <Transition
1316         fromBusinessState="Test Order Accepted"
1317         fromBusinessStateIDREF="54654B567"
1318         toBusinessState="Notify Shipment"
1319         toBusinessStateIDREF="122A39CA3"
1320         conditionGuard="Success">
1321         <ConditionExpression
1322             expressionLanguage=
1323             "DocumentEnvelopeNotation"
1324             conditionExpression=
1325             "Accept Order"/>
1326     </Transition>
1327     <Transition
1328         fromBusinessState="Create Order"
1329         fromBusinessStateIDREF="122A39CA3"
1330         toBusinessState="Test Order Accepted"
1331         toBusinessStateIDREF="54654B789"/>
1332     <Transition
1333         fromBusinessState="Notify Shipment"
1334         fromBusinessStateIDREF="122A39CA3"
1335         toBusinessState="Test Success"
1336         toBusinessStateIDREF="54654B789"/>
1337     <Decision
1338         name="Test Success"
1339         nameID="54654B789"/>
1340     <Decision
1341         name="Test Order Accepted"
1342         nameID="54654B567"/>
1343 </BinaryCollaboration>

```


1344 Note that all the completion states of this binary collaboration definition are mutually
 1345 exclusives. Optionally the transition with the condition expression could be expressed with
 1346 an XPath predicate:

```

1347 <Transition
1348     onInitiation="false"
1349     fromBusinessState="Update Repair Order"
1350     fromBusinessStateIDREF="122A39CA3"
1351     toBusinessState="Process Repair Order"
1352     toBusinessStateIDREF="122A39C23"
1353     conditionGuard="Success">
1354     <ConditionExpression
1355         expressionLanguage="XPath 1.0"
1356         conditionExpression=
1357         "//POAck[@status='Reject']"/>
1358 </Transition>

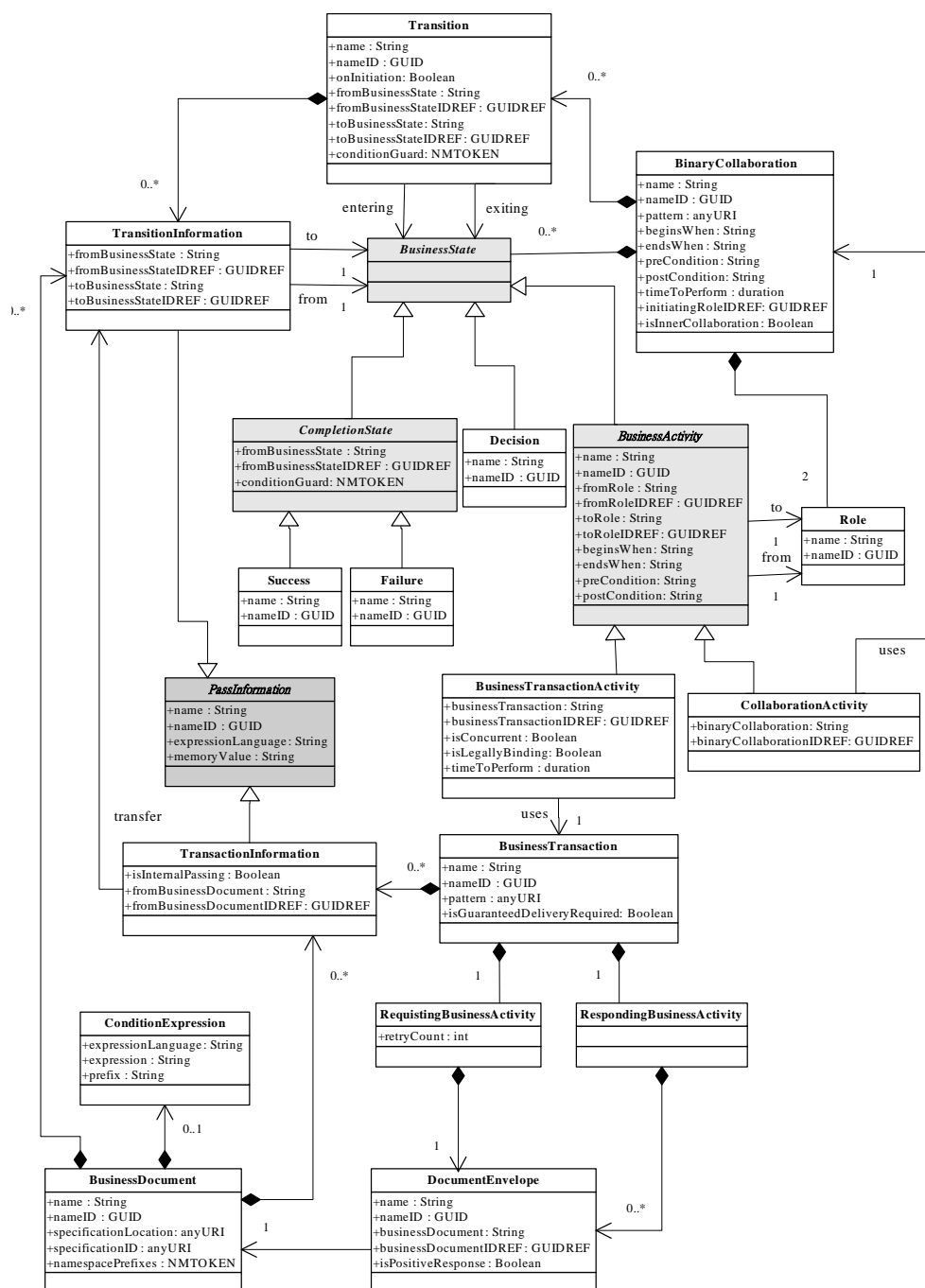
```

1359 Similarly, transitions can be defined between Business Activities of a multi-party
 1360 collaboration.

```

1361 <MultiPartyCollaboration name="DropShip">
1362     <BusinessPartnerRole name="Customer">
1363         <Performs role="requestor" roleIDREF="1122B1"/>
1364         <Performs role="buyer" roleIDREF="1122B2"/>
1365         <Transition
1366             fromBusinessState="Catalog Request"
1367             toBusinessState="Create Order"/>
1368     </BusinessPartnerRole>
1369     <BusinessPartnerRole name="Retailer">
1370         <Performs role="provider" roleIDREF="2211A1"/>
1371         <Performs role="seller" roleIDREF="1122B3"/>
1372         <Performs role="creditor" roleIDREF="9122B1"/>
1373         <Performs role="buyer" roleIDREF="1122B2"/>
1374         <Performs role="payee" roleIDREF="6122B1"/>
1375         <Performs role="requestor" roleIDREF="1122B1"/>
1376         <Transition
1377             fromBusinessState="Create Order"
1378             toBusinessState="Check Credit"/>
1379         <Transition
1380             fromBusinessState="Check Credit"
1381             toBusinessState="Credit Payment"/>
1382     </BusinessPartnerRole>
1383     <BusinessPartnerRole name="DropShip Vendor">
1384         <Performs role="seller" roleIDREF="1122B3"/>
1385         <Performs role="payee" roleIDREF="6122B1"/>
1386         <Performs role="creditor" roleIDREF="9122B1"/>
1387         <Performs role="provider" roleIDREF="2211A1"/>
1388     </BusinessPartnerRole>
1389     <BusinessPartnerRole name="Credit Authority">
1390         <Performs role="credit service" roleIDREF="8122B1"/>
1391         <Performs role="payor" roleIDREF="7122B1"/>
1392     </BusinessPartnerRole>
1393 </MultiPartyCollaboration>

```

1394 **5.12.7 Pass Information within/between Business Transactions**

1395

1396

Figure 15. UML Diagram of Business Transaction with Pass Information

1397 The mechanisms to remember/keep information of previous events and exchanged data
 1398 messages is an abstract class named "Pass Information" in Figure 15. And there are two
 1399 concrete classes inherited from Pass Information – Transaction Information and Transition
 1400 Information. The "Pass Information" class and its two subclasses in the UML diagram is
 1401 explained as the following.

1402 Transaction Information would keep important information about the value of any attributes
 1403 in the business document from the requesting role within the business transaction and add
 1404 the information in the responding business documents at run time if necessary. The optional
 1405 attribute "expressionLanguage" can be used to specify the method to retrieve the value from
 1406 the element of the document with "fromBusinessDocumentIDREF". If it is not used, the
 1407 "memoryValue" would be a fixed value specified by design time. By this way, the
 1408 important information within the business transaction would be stored in the Transaction
 1409 Information and thereby would not be lost. For example, if there are two important
 1410 information in the business transaction "Purchase Order", the value of "PO number"
 1411 attribute and the product list. We can identify two Transaction Informations in the BPS to
 1412 keep the values of the related attributes in the requesting business document and echo them
 1413 in the responding business documents. After the requesting party receive the responding
 1414 business documents, it can check if the important information (the PO number and the
 1415 product list) was missing or mismatched during the message exchange.

1416 Here is a business transaction – "CreateOrder " with two document flows, three business
 1417 documents and one Transaction Information.

```

1418 <BusinessDocument
1419     name="Purchase Order"
1420     nameID="122A3F613"
1421     specificationLocation="http://.../purchaseOrder.xsd"
1422     specificationID="http://... / purchaseOrder.xsd"
1423     namespacePrefixes="fix">
1424 </BusinessDocument>
1425
1426 <BusinessDocument
1427     name="Reject Order"
1428     nameID="122A3F8E3"
1429     specificationLocation="http://.../rejectOrder.xsd"
1430     specificationID="http://... /rejectOrder.xsd"
1431     namespacePrefixes="fix">
1432 </BusinessDocument>
1433
1434 <BusinessDocument
1435     name="Accept Order"
1436     nameID="122A3F6C3"
1437     specificationLocation="http://.../ acceptOrder.xsd"
1438     specificationID="http://... /acceptOrder.xsd"
1439     namespacePrefixes="fix">
1440 </BusinessDocument>
1441
1442 <BusinessTransaction
1443     name="CreateOrder"
1444     nameID="122A3DD33"
1445     isGuaranteedDeliveryRequired="true">
  
```

```

1446     <RequestingBusinessActivity
1447         name="sendOrder"
1448         nameID="122A3E833"
1449         isNonRepudiationReceiptRequired="false"
1450         isNonRepudiationRequired="false"
1451         timeToAcknowledgeProcessing="P1H"
1452         timeToAcknowledgeReceipt="P1H">
1453         <DocumentEnvelope
1454             businessDocument="Purchase Order"
1455             businessDocumentIDREF="122A3F613"/>
1456     </RequestingBusinessActivity>
1457
1458     <Transaction Information
1459         name="purchaseOrderNumber"
1460         nameID="122A3F7Z3"
1461         expressionLanguage="XPath1.0"
1462         memoryValue="//purchaseOrderNumber"
1463         isInternalPassing="true"
1464         fromBusinessDocument="Purchase Order"
1465         fromBusinessDocumentIDREF="122A3F613">
1466     </Transaction Information>
1467
1468     <RespondingBusinessActivity
1469         name="sendPOAcknowledgement"
1470         nameID="122A3E863"
1471         isNonRepudiationReceiptRequired="false"
1472         isNonRepudiationRequired="false"
1473         timeToAcknowledgeReceipt="P1D">
1474         <DocumentEnvelope
1475             isPositiveResponse="false"
1476             businessDocument="Reject Order"
1477             businessDocumentIDREF="122A3F8E3"/>
1478         <DocumentEnvelope
1479             isPositiveResponse="true"
1480             businessDocument="Accept Order"
1481             businessDocumentIDREF="122A3F6C3"/>
1482     </RespondingBusinessActivity>
1483 </BusinessTransaction>

```

In the business transaction above, the purchaseOrderNumber of the business document "Purchase Order" with nameID "122A3F613" in the RequestingBusinessActivity would be stored in the Transaction Information with name "purchaseOrderNumber". This value could be echoed into the business document "Accept Order" and "Reject Order" by the application software of responding role at run time. After the requesting role receive the responding business document "Accept Order" or "Reject Order", the application software of requesting role could check whether the responding business document matches the original requesting business document.

If any important information has to be passed from one business transaction to another, the other concrete class inherited from Pass Information— Transition Information must be used to store this information. Transition Information would keep the value of any attribute in any business document in the business transaction from one business state to another business state. And the information could be added into the business documents in the business transactions of the new business state by the application software at run time. By

1498 this way, the important information could be passed through business states between
 1499 different business transactions or different business collaborations and thereby would not be
 1500 lost.

1501 Each Transition Information refers to both the business state where it came from and the
 1502 business state where it transit to. So it can pass the information from one state to another,
 1503 from one business transaction to another, and from one business collaboration to another.

1504 Unlike the value of memoryValue attribute in Transaction Information, the value of
 1505 memoryValue attribute in Transition Information could be added in both the requesting and
 1506 the responding business documents at run time. A Transition Information can transfer to a
 1507 Transition Information if the information stored in memoryValue attribute has to be passed
 1508 outside the local business transaction from one business transition to another business
 1509 transition and the scope of the Pass Information has should be extended from Transaction
 1510 Information to Transition Information.

1511 For example, suppose that there are two business transactions—"Purchase Order (PO)" and
 1512 "Request for Payment (RP)" in a business collaboration. If two partners start two bsiness
 1513 collaborations to order different products in parallel, there shall be two "Purchase Order"
 1514 and two "Request for Payment": PO1—RP1 and PO2—RP2 in different business
 1515 collaborations. In order to make sure that the business document "Invoice" sent in of RP1 of
 1516 the first business collaboration is in any way related to PO1, some information of PO1 must
 1517 be remembered and then used to check if the invoice received in RP1 corresponds to PO1,
 1518 otherwise it is possible to send any invoice related to PO2 in the first business collaboration
 1519 and thus complete the first business collaboration leaving the second behind.

1520 If we keep the ID information of the requesting business document in PO1, pass the ID
 1521 information from "Purchase Order" business transaction to another business transaction,
 1522 "Request for Payment", and add this ID information in the requesting business document,
 1523 "Invoice", of Request for Payment, the connection of PO1—Invoice1—RP1 and PO2—
 1524 Invoice1—RP2 is established and thus resolve the problem of mismatched business
 1525 documents.

1527 Figure 16 represents the complete metamodel of ebXML BPSS as a UML class diagram.



1529 Figure 16. Overall ebXML Business Process Specification Schema as UML class diagram

5.13 Core Business Transaction Semantics

The ebXML concept of a business transaction and the semantics behind it are central to predictable, enforceable commerce. It is expected that any Business Service Interface (BSI) will be capable of managing a transaction according to these semantics.

The ebXML Business Transaction semantics, i.e. the rules and configuration parameters required for Business Service Interface software to predictably and deterministically execute ebXML Business Transactions, allows you to specify electronic commerce transactions that provide

- Interaction Predictability, i.e. have clear roles, clear transaction scope, clear time bounds, clear business information semantics, clear determination of success or failure. Each party can compute without ambiguity and the status of a transaction independently.
- Ability to create Legally Binding Contracts, i.e. the ability to specify that Business Transactions may be agreed to bind the parties. This concept is being deprecated as of this version.
- Nonrepudiation, i.e. may specify the keeping of artifacts to aid in legal enforceability.
- Authorization Security, i.e. may be specified to require authorization of parties performing roles.
- Document Security, i.e. may be specified to be authorized, authenticated, confidential, tamper detectable.
- Reliability, i.e. the ability to specify reliable delivery of Business Documents and signals.

Each of the above characteristics of the concept that we call an ebXML Business Transaction semantics is discussed in detail below.

These desirable characteristics are only applicable to ebXML *Business Transactions*, where an ebXML *Business Transaction* is a single request or single request / response pair only. A future version of this specification may extend the applicability of these characteristics to other types of electronic commerce transactions. In particular, we do not claim that the ebXML *Business Transaction* concept covers all possible electronic commerce transactions. For instance, a use case could involve an electronic commerce transaction that exchanges a request and two responses as a unit of work. If we would want to have similar properties, this kind of use cases would not be directly covered by this specification. The only way to handle such a use case would be to specify the electronic commerce transaction as a binary collaboration involving as many ebXML Business Transaction as necessary. The binary collaboration definition would then be specified in such a way to handle the individual ebXML Business Transaction exceptions and aggregate them into the electronic commerce transaction.

5.13.1 Interaction Predictability

All Business Transactions follow a very precisely prescribed flow, or a precisely defined subset there-of. The following is an overall illustration of this flow. It can be thought of as the state machine across the two business partners.

The goal of the Business Transaction Protocol is to synchronize the business state between two parties. As few resources can be shared between company boundaries, we must use such protocol to achieve the business state synchronization as recorded by each party enterprise systems.

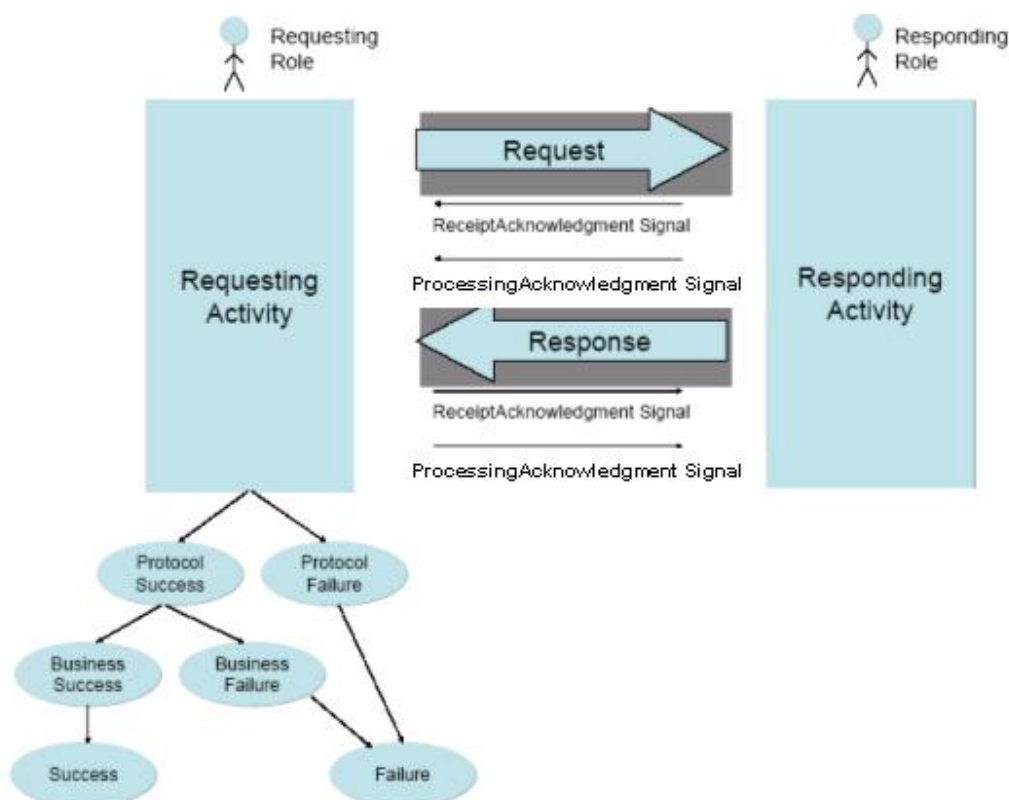


Figure 17. Schematic of core Business Transaction semantics.

Figure 16 does not assume any hierarchy in the way exceptions are generated or evaluated. It simply state that in oder to achieve a success state a business transaction activity complete with both a procotol and a business success. These exception are constantly evaluated by the BSI, and thrown as soon as detected.

If either a protocol or business failure occurs, the business transaction activity will be put into a failure state.

In the ebXML model the business transaction always has the following semantics.

- 1584 1. The Business Transaction is an atomic unit of work. All of the interactions in a business
1585 transaction must succeed or each party must not change their state.
- 1586 2. A Business Transaction is conducted between two business partners playing opposite
1587 roles in the transaction. These roles are always the Requesting Role and the Responding
1588 Role.
- 1589 3. A Business Transaction definition specifies exactly when the Requesting Activity is in
1590 control, when the Responding Activity is in control, and when control transitions from
1591 one to the other. In all Business Transactions control starts at the Requesting Activity,
1592 then transitions to the Responding Activity, and then returns to the Requesting Activity.
- 1593 4. A Business Transaction always starts with a request sent out by the requesting activity.
- 1594 5. The request serves to transition control to the responding role.
- 1595 6. After the receipt of the Request document flow, the responding activity may send a
1596 receiptAcknowledgement signal and/or an processingAcknowledgement signal to the
1597 requesting role.
- 1598 7. The responding role then enters a responding activity. During or upon completion of the
1599 responding activity zero or one response is sent.
- 1600 8. Control will be returned back to the requesting activity if either a
1601 receiptAcknowledgement and/or processingAcknowledgement and/or a response is
1602 specified as required. A receiptAcknowledgement (if required) must always occur
1603 before an processingAcknowledgement (if required), and an
1604 processingAcknowledgement must always occur before a response (if required). Control
1605 is returned to the requesting activity based on the last required of these three (if any). If
1606 none required, control stays with the responding activity.
- 1607 9. All business transactions succeed or fail. Success or failure depends on:
 - 1608 a. The successful transmission of the request, the response and/or receipt and
1609 processing signals
 - 1610 b. The occurrence of time-outs
 - 1611 c. The occurrence of exceptions, as indicated by a negative receipt or processing
1612 signals
 - 1613 d. The computation of business failure or success by detecting if the response
1614 document was specified – at design time – with isPositiveResponse=false.
- 1615 10. Both parties can compute the success or failure of the transaction if reliable messaging,
1616 as well as request and response processing acknowledgement signals, are used. Once
1617 success or failure is thus established, the Business Transaction is considered closed with
1618 respect to both parties. If reliable messaging is not used, we cannot guarantee state
1619 alignment and therefore it could happen that one party believe the transaction has been
1620 successful, while the other believes it ended in failure.
- 1621 11. Upon receipt of a response the requesting activity may send a receiptAcknowledgement
1622 and/or processingAcknowledgement signal back to the responding role. This operation

does not pass control back to the responding activity. If the requesting party send the signals after the timeout has occurred, the transaction is considered null and void.

12. Upon identifying a time-out or exception in the processing of a Business Transaction each party will close the transaction and end in a protocol failure state.

5.13.1.1 Transaction Interaction Patterns

The business transaction specification will specify whether a requesting document requires a responding substantive document in order to achieve a "success" end state. In addition, the transaction may specify a proper nonzero time duration for timeToPerform, imposing a deadline for the substantive response.

Furthermore, the specification of a business transaction may indicate, for the request whether receiptAcknowledgement and/or processingAcknowledgement are required, and for the response whether receiptAcknowledgement and/or processingAcknowledgement are required.

The way to specify that a receiptAcknowledgement is required is to set the parameter timeToAcknowledgeReceipt to any proper time duration other than zero. If this parameter has been set to a proper nonzero time duration, optionally either or both of the isIntelligibleCheckRequired and isNonrepudiationOfReceiptRequired parameters may also be set to 'Yes'.

The way to specify that a processingAcknowledgement is required is to set the parameter timeToAcknowledgeProcessing to any proper time duration other than zero.

So these two acknowledgement related parameters double as Boolean flags for whether the signal is required as part of the transaction, and as values for time-out of the transaction if the signal is not received.

The specification of a business transaction may require each one of these signals independently of whether the other is required. Therefore there is a finite set of combinations. The UMM supplies the currently defined set of transaction patterns.

The specification of a business transaction MAY require each one of these signals independently of whether the other is required. Therefore there is a finite set of combinations. The UN/CEFACT BPSS specification supports a subset of all possible combinations:

- Commercial Transaction
- Query / Response
- Request / Confirm
- Request / Response

- 1658 ● Information Distribution
- 1659 ● Notification

1660

1661 **5.13.2 Creating legally binding contracts**

1662 Trading partners may wish to indicate that a Business Transaction performed as part of an
 1663 ebXML arrangement is, or is not, intended to be binding. A declaration of intent to be
 1664 bound is a key element in establishing the legal equivalence of an electronic message to an
 1665 enforceable signed physical writing. Parties may create explicit evidence of that intent by (1)
 1666 adopting the ebXML Business Process Specification Schema standard and (2) manipulating
 1667 the parameter ("isLegallyBinding") designated by the standard to indicate that intent.

1668 In some early electronic applications, trading partners have simply used the presence, or
 1669 absence, of an electronic signature (such as under the XML- DSIG standard) to indicate that
 1670 intent. However, documents which rely solely on the presence of a signature may or may
 1671 not be correctly interpreted, if there is semantic content indicating that a so-called contract
 1672 is a draft, or nonbinding, or the like.

1673 In ebXML, the presence or absence of an electronic signature cannot indicate by itself
 1674 legally binding assent, because XML-DSIG signatures are reserved for other uses as an
 1675 assurance of sender identity and message integrity. isLegallyBinding is a parameter at the
 1676 BusinessTransactionActivity level, which means that the performing of a
 1677 BusinessTransaction within a Binary Collaboration is either specified as legally binding or
 1678 not.

1679 When operating under this standard, parties form binding agreements by exchanging
 1680 binding messages that agree to terms (e.g., offer and acceptance). The "isLegallyBinding"
 1681 parameter is Boolean, and its default value is "true." Under this standard, the exclusive
 1682 manner for indicating that a Business Activity is not intended to be binding is to include a
 1683 "false" value for the "isLegallyBinding" parameter for the transaction activity. As in EDI,
 1684 the ebXML standard assumes that Business Transactions are intended by the trading parties
 1685 to be binding unless otherwise indicated.

1686 As a non-normative matter, parties may wish to conduct nonbinding transactions for a
 1687 variety of reasons, including testing, and the exchange of proposed offers and counteroffers
 1688 on a non-committal basis so as to discover a possible agreed set of terms. When using
 1689 tangible signed documents, parties often do so by withholding a manual signature, or using
 1690 a "DRAFT" stamp. In ebXML, trading partners may indicate that result by use of the
 1691 "isLegallyBinding" parameter. See the illustrative Simple Negotiation Pattern set forth in
 1692 the ebXML E-Commerce Patterns.

1693 **5.13.3 Non-Repudiation**

1694 Trading partners may wish to conduct legally enforceable business transactions over
 1695 ebXML. A party may elect to use non-repudiation protocols in order to generate
 1696 documentation that would assist in the enforcement of the contractual obligation in court, in
 1697 the case that the counterparty later attempts to repudiate its ebXML Business Documents
 1698 and messages.

1699 Repudiation generally refers to the ability of a trading partner to argue at a later time, based
 1700 on the persistent artifacts of a transaction, that it did not agree to the transaction. That
 1701 argument might be based on assertions that a replying document was not sent, or was not
 1702 sent by the proper party, or was incorrectly interpreted (under the applicable standard or the
 1703 trading partners' business rules) as forming agreement.

1704 There are two kinds of non-repudiation protocol available under this document. Each
 1705 protocol provides the user with some degree of additional evidentiary assurance by creating
 1706 or requesting additional artifacts that would assist in a later dispute over repudiation issues.
 1707 Neither is a dispositive absolute assurance. As in the paper world, trading partners are
 1708 always free to invent colorful new arguments that an apparently-enforceable statement
 1709 should be ignored. These parameters simply offer some opportunities to make that more
 1710 difficult.

1711 One imposes a duty on each party to save copies of all Business Documents and Document
 1712 Envelopes comprising the transaction in the form they where received(e.g. save in
 1713 encrypted form if they where received in encrypted form) , each on their own side, i.e.,
 1714 requestor saves his request, responder saves his response. This is the
 1715 isNonRepudiationRequired parameter in the requesting or responding activity. It is logically
 1716 equivalent to a request that the other trading partner maintain an audit trail. However,
 1717 failure to comply with that request is not necessarily computationally detectable at run time,
 1718 nor would it override the determination of a "success" or "failure" end state. This relates to
 1719 the business action concept in the UMM.

1720 The other requires the receiver of a business document to send a signed receipt, which the
 1721 original sender saves. This is the isNonRepudiationOfReceiptRequired parameter in the
 1722 requesting and responding business activity.

1723 NonRepudiationOfReceipt is tied to the ReceiptAcknowledgement, in that it requires the
 1724 latter to be digitally signed. So NonRepudiationOfReceipt is meaningless if
 1725 ReceiptAcknowledgement is not required. Failure to comply with NonRepudiation of
 1726 Receipt would be computationally detectable at run time, and would override the
 1727 determination of a "failure" end state. If a timeToAcknowledgeReceipt is imposed on a
 1728 requesting message, and NonRepudiationOfReceipt is true, only a digitally signed receipt
 1729 will satisfy the imposed timeout deadline. Thus, a failure to send a *signed* receipt within
 1730 timeToAcknowledgeReceipt, would make the transaction null and void.

1731 **5.13.4 Authorization security**

1732 Each request or response may be sent by a variety of individuals, representatives or
 1733 automated systems associated with a business partner. There may be cases where trading
 1734 partners have more than one ebXML-capable business service interface, representing
 1735 different levels of authority. In such a case, the parties may establish rules regarding which
 1736 interfaces or authors may be confidently relied upon as speaking for the enterprise.

1737 In order to invoke those rules, a party may specify *isAuthorizationRequired* on a requesting
 1738 and/or a responding activity accordingly, with the result that [the activity] will only be
 1739 processed as valid if the party interpreting it successfully matches the stated identity of the
 1740 activity's [Role] to a list of allowed values previously supplied by that party.

1741 *isAuthorizationRequired* is specified on the requesting and responding activity accordingly.

1742 This concept is deprecated as of this version. Its specification might change in a future
 1743 release and is not required for an ebXML BPSS 1.1 compliant BSI infrastructure. In this
 1744 version, a BSI would have no way to specify that an attempt has been made by an
 1745 application or system to initiate a Business Transaction (therefore sending a request) and
 1746 this application or system was not authorized to do so.

1747 **5.13.5 Document security**

1748 The value of *isConfidential*, *isTamperDetectable*, *isAuthenticated* at the Document
 1749 Envelope always applies to the primary Business Document. It also applies to each of the
 1750 attachments unless specifically overridden at the Attachment level. These parameters can
 1751 have four possible values: none, transient, persistent, transient-and-persistent.

1752 Transient authentication is provided by the communications channel used to transport the
 1753 *Message*. The specific method will be determined by the communications protocol used.

1754 Persistent authentication means the Business Document signer's identity shall be verified at
 1755 the receiving application level.

1756 Transient confidentiality is provided by a secure network protocol, such as SSL as the
 1757 document is transferred between two adjacent MSH nodes.

1758 Persistent confidentiality is intended to preserve the confidentiality of the message such that
 1759 only the intended party (application) can see it. The message shall remain in encrypted form
 1760 after it is delivered to the MSH node and will be decrypted only by the authorized
 1761 application. S/MIME can be used to provide that functionality, independent of the transient
 1762 confidentiality.

1763 Transient *isTamperDetectable* is the ability to detect if the information has been tampered
 1764 with during transfer between two adjacent MSH nodes.

1765 Persistent *isTamperDetectable* is the ability to detect if the information has been tampered
 1766 with after it has been received by MSH, between the MSH and the application.

1767 **5.13.6 Reliability**

1768 This parameter *isGuaranteedDeliveryRequired* at the Business Transaction level states
 1769 whether guaranteed delivery of the transaction's Business Documents is required.

1770 This is a declaration that trading partners must employ only a delivery channel that provides
 1771 a delivery guarantee, to send Business Documents in the relevant transaction.

1772 **5.13.7 Parameters required for CPP/CPA**

1773 The ebXML *Business Process Specification Schema* provides parameters that can be used to
 1774 specify certain levels of security and reliability. The ebXML *Business Process Specification*
 1775 *Schema* provides these parameters in general business terms.

1776 These parameters are generic requirements for the business process, but for ebXML
 1777 implementations, these parameters are specifically used to instruct the CPP and CPA to
 1778 require BSI and/or delivery channel capabilities to achieve the specified service levels.

1779 The CPP and CPA translate these into parameters of two kinds.

1780 One kind of parameters determines the selection of certain security and reliability
 1781 parameters applicable to the transport method and techniques used by the delivery channel.
 1782 Document security, and Reliability above, are determinators of delivery channel selection.

1783 The other kind of parameters determines the selection of certain service levels or
 1784 capabilities of the BSI itself, in order for it to support the run time Business Transaction
 1785 semantics as listed below.

1786 **5.14 Run time Business Transaction semantics**

1787 The ebXML concept of a business transaction and the semantics behind it are central to
 1788 predictable, enforceable commerce. It is expected that any Business Service Interface (BSI)
 1789 will be capable of managing a transaction according to these semantics.

1790 Therefore, the Business Service Interface (BSI), or any software that implements one role in
 1791 an ebXML collaboration needs at minimum to be able to support the following transaction
 1792 semantics:

- 1793 1. Detection of the opening of a transaction
- 1794 2. Detection of transfer of control
- 1795 3. Detection of successful completion of a transaction
 - 1796 a. Application of business rules expressed as schema definitions and
 - 1797 isPositiveResponse for determination of success
- 1798 4. Detection of failed completion of a transaction
 - 1799 a. Detection of time-outs
 - 1800 b. Detection of protocol exceptions
 - 1801 c. Validation of the received response and identify if it was specified with
 - 1802 isPositiveResponse = false

1803 ebXML does not specify how these transaction semantics are implemented but it is assumed
 1804 that any Business Service Interface (BSI) will be able to support these basic transaction
 1805 semantics at runtime. If either party cannot provide full support, then the requirements may
 1806 be relaxed as overrides in the CPP/CPA.

1807 The following sections discuss the two causes of failure: timeouts and exception. When
 1808 either one happens, it is the responsibility of the two roles to exit the transaction. It is also
 1809 expected that the corresponding collaboration will be designed (and choreographed) to
 1810 execute the appropriate compensating transactions if needed and may reach a completion
 1811 state after that. The responsibilities of the two roles differ slightly and are described in each
 1812 of the sections below. Generally, if a failure other than a timeout happens at either the
 1813 responding or requesting role, they will send an exception signal to the other role, and both
 1814 parties will exit the current transaction.

1815 **5.14.1 Timeouts**

1816 Since all business transactions must have a distinct time boundary, there are timeout
 1817 parameters associated with the response and each of the acknowledgement signals. If the
 1818 timeout occurs before the corresponding response or signal arrives, the transaction is null
 1819 and void.

1820 Here are the timeout parameters relative to the three response types:

| Response required | Parameter Name and meaning of the timeout |
|---|--|
| Receipt Acknowledgement | <i>timeToAcknowledgeReceipt</i> |
| | The time a responding or requesting role has to acknowledge receipt of a business document. |
| Processing Acknowledgement (Non-substantive) | <i>timeToAcknowledgeProcessing</i> |
| | The time a responding or requesting role has to non-substantively acknowledge business processing acceptance of a business document. |
| Substantive Response | <i>timeToPerform</i> |
| | The maximum amount of time between the time at which the request is sent and the substantive response is sent. |

1821 Note that the Processing Acknowledgement signal is often called the “non-substantive”
1822 response to the request.

1823 A timeout parameter must be specified whenever a requesting or responding partner expects
1824 signals in return to the business document request or response. A requesting partner must
1825 not remain in an infinite wait state.

1826 The timeout value for each of the timeout parameters is absolute i.e. not relative to each
1827 other. All timers start when the initial requesting business document is sent. The timer
1828 values must comply with the well-formedness rules for timer values.

1829 A BSI needs to comply with the above parameters to detect the appropriate timeouts. To
1830 preserve the atomic semantics of the Business Transaction, the requesting and responding
1831 roles take different action based on timeouts. A responding partner simply terminates if a
1832 timeout is thrown. This prevents responding business transactions from hanging indefinitely.

1833 The total time allowed for a business transaction activity to complete is therefore,
1834 timeToPerform plus the timeToAcknowledgeReceipt on the response, and the
1835 timeToAcknowledgeProcessing on the response. Additionally, timeToPerform must be
1836 greater than the sum of timeAcknowledgeReceipt and timeToAcknowledgeProcessing and
1837 the request.

1838 **5.14.2 Protocol Exceptions**

1839 In addition to timeouts, the Business Transaction protocol provides a series of protocol
1840 exception which indicate whether the business processing of the transaction went wrong at
1841 either the responding or the requesting role.

5.14.2.1 Receipt Acknowledgement Exception

A *Receipt Exception* signals an error condition in the management of a business transaction. This business signal is returned to the initiating activity that originated the request. This exception must terminate the business transaction. These errors deal with the mechanisms of message exchange such as verification, validation, authentication and authorization and will occur up to message acceptance. Typically the rules and constraints applied to the message will have only dealt with the well-formedness of the message.

A receipt exception terminates the business transaction. The following are receipt exceptions:

- Syntax exceptions. There is invalid punctuation, vocabulary or grammar in the business document or business signal.
- Authorization exceptions. Roles are not authorized to participate in the business transaction activity. Note that this exception can only be identified by the receiving BSI.
- Signature exceptions. Business documents are not signed for non-repudiation when required.
- Sequence exceptions. The order or type of a business document or business signal is incorrect.

A receipt exception typically means that the current message could not be handed to an application for processing.

5.14.2.2 Processing Acknowledgement Exceptions

An Processing Exception signals an error condition in a business activity. This business signal is returned to the initiating role that originated the request. This exception must terminate the *business transaction*. These errors deal with the mechanisms that process the *business transaction* and will occur after message verification. Typically the rules and constraints applied to the message will deal with the semantics of message elements and the validity of the request itself. The content is not valid with respect to a responding role's business rules.

An Processing Exception terminates the business transaction. The following are business protocol exceptions:

- Business exception. The business rules of the responding activity are violated. The application refused to process the incoming business document. Most often because it violated some pre-processing business rules.
- Performance exceptions. The requested business action cannot be performed. The application may not be available.

1877 Typically, an Processing Exception means that the processing application (usually unknown
 1878 to the other party) received the corresponding businessdocument but was enable to process
 1879 them.

1880 A Business Transaction is defined in very atomic and deterministic terms. It always is
 1881 initiated by the requesting role, and will always conclude at the requesting role. Upon
 1882 receipt of the required response and/or signals, or time-out of same, the requesting role can
 1883 unambiguously determine the success or failure of the Business Transaction. A responding
 1884 role that encounters an Processing Exception signals the exception back to the requesting
 1885 role and then terminates the business transaction.

1886 Conversely, a requesting role that encounters an Processing Exception signals the exception
 1887 back to the responding role and terminates the transaction

1888 **5.14.2.3 BSI compliance**

1889 A BSI needs to comply specifically with the following parameters to produce the associated
 1890 special exceptions. The requesting and responding roles take different action as per below.

1891 ***IsAuthorizationRequired***

1892 If a partner role needs authorization to request a business action or to respond to a business
 1893 action then the sending partner role must sign the business document exchanged and the
 1894 receiving partner role must validate this business control and approve the authorizer. A
 1895 responding partner must signal an authorization exception (receipt exception) if the
 1896 requesting partner role is not authorized to perform the business activity. A sending partner
 1897 must send notification of failed authorization if a requesting partner is not authorized to
 1898 perform the responding business activity.

1899 ***IsNonRepudiationRequired***

1900 If non-repudiation of origin and content is required then the business activity must store the
 1901 business document in its original form for the duration mutually agreed to in a trading
 1902 partner agreement. A responding partner must signal a receipt exception if the sending
 1903 partner role has not properly delivered their business document. Similarly, a requesting
 1904 partner must send receipt exception if a responding partner has not properly delivered their
 1905 business document.

1906 ***isNonRepudiationOfReceiptRequired.***

1907 Both partners agree to mutually verify receipt of a requesting business document and that
 1908 the receipt must be non-repudiable. A requesting partner must initiate a notification of
 1909 failure business transaction business (possibly revoking a contractual offer) if a responding
 1910 partner has not properly delivered signed their receipt. For a further discussion of
 1911 nonrepudiation of receipt, see also the ebXML E-Commerce and Simple Negotiation
 1912 Patterns.

- 1913 Non-repudiation of receipt provides the data for the following audit controls.
- 1914 **Verify responding role identity** (authenticate) – Verify the identity of the responding role
1915 (individual or organization) that received the requesting business document.
- 1916 **Verify content integrity** – Verify the integrity of the original content of the business
1917 document request.
- 1918 *isPositiveResponse*
- 1919 An expression whose evaluation results in TRUE or FALSE. If TRUE this
1920 DocumentEnvelope is intended as a positive response to the request. If isPositiveResponse
1921 = FALSE, the business transaction activity ends in business failure mode. The value for this
1922 parameter supplied for a DocumentEnvelope is an assertion by the sender of the
1923 DocumentEnvelope regarding its intent for the transaction to which it relates, but does not
1924 bind the recipient, or override the computation of transactional success or failure.

1925 5.14.3 Computation of the status of a Business Transaction Activity

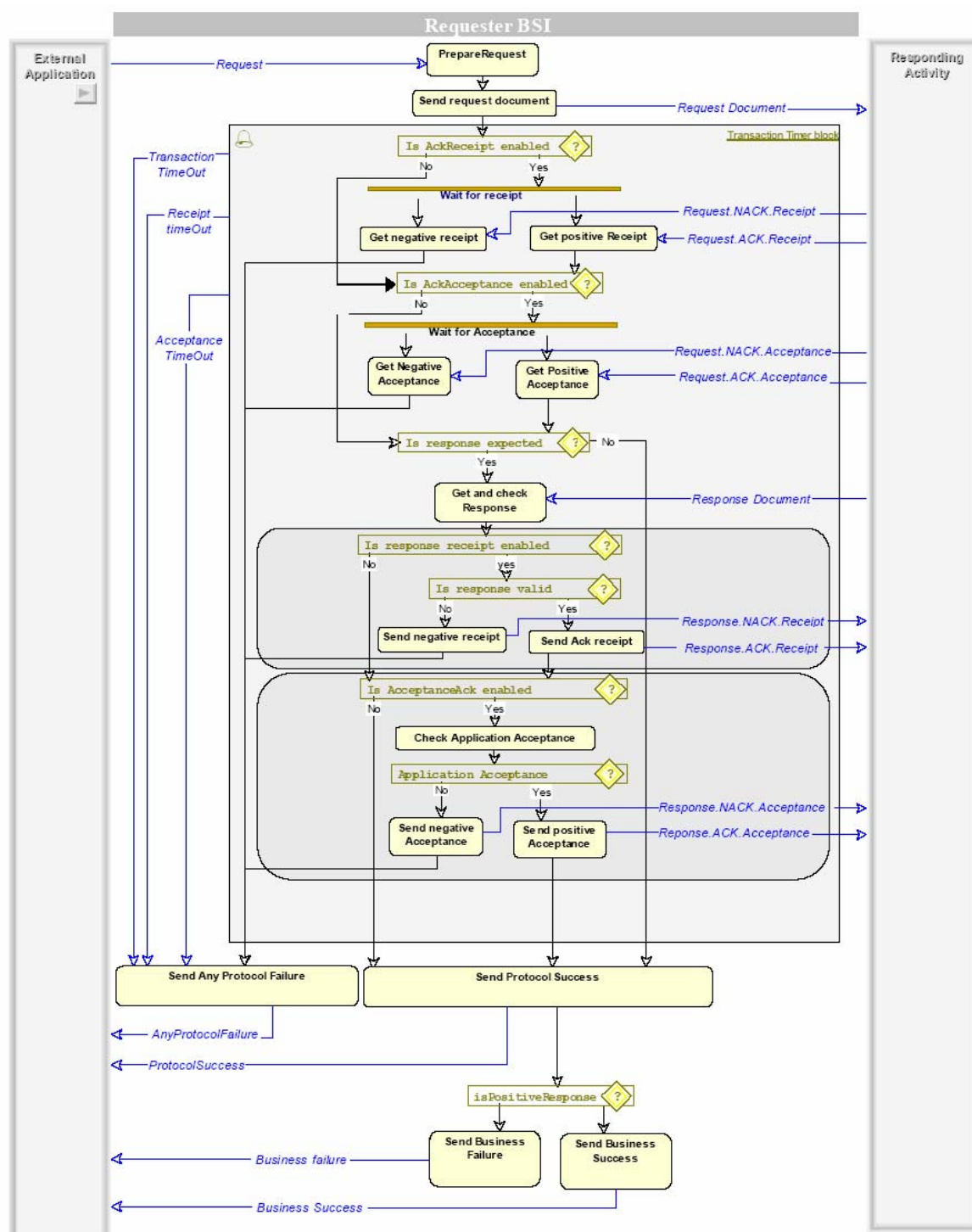


Figure 18. Computation of the Status of a Business Transaction Activity

Figure 17 represent the computation of the success or failure of a business transaction activity based on the different possible scenarios.

1929 The values of the enumeration of the state of a business transaction of the conditionGuard
1930 on a transition are:

- 1931 ● ProtocolSuccess
- 1932 ● AnyProtocolFailure
 - 1933 □ RequestReceiptFailure
 - 1934 □ RequestProcessingFailure
 - 1935 □ ResponseReceiptFailure
 - 1936 □ ResponseProcessingFailure
 - 1937 □ SignalTimeout
 - 1938 □ ResponseTimeout
- 1939 ● BusinessSuccess (isPositiveResponse=true or no isPositiveResponse attribute)
- 1940 ● BusinessFailure(isPositiveResponse=false)
- 1941 ● Success (both protocol and business success)
- 1942 ● Failure (AnyProtocolFailure or BusinessFailure).

1943 This figure does not represents the retryCount semantics.

1944 BusinessFailure assumes that the transaction was successful from a “protocol” perspective,
1945 meaning that the state between the two parties could be effectively synchronized. However,
1946 the intent of the response was negative with respect to the request. As we mentioned earlier,
1947 this is an optional qualification of the response, agreed upon at design time, and some
1948 messages may not be qualifiable, i.e. they are neither positive or negative. The way business
1949 document specifications are designed allows to define two “logical” documents from the
1950 same physical document and a condition expression evaluated at run-time by the BSI. If the
1951 condition is true and isPositiveResponse = false, then the transaction ends in business
1952 failure based on the business document content. Of course entire documents can be directly
1953 associated with isPositiveResponse=false, not just when they contain a particular field value.

1954 It is required that each business transaction activity be designed such that there is at a
1955 minimum two transitions from the business transaction activity, one with a conditionGuard
1956 with a Success value, the other one with a Failure value, even if in case of failure the
1957 transitions goes to the failure state of the collaboration.

1958 **5.15 Runtime Collaboration Semantics**

1959 The ebXML collaboration semantics contain a number of relationships between multiparty
1960 collaborations and binary collaborations, between recursive layers of binary collaborations,
1961 and choreographies among transactions in binary collaborations. It is anticipated that over
1962 time BSI software will evolve to the point of monitoring and managing the state of a

1963 collaboration, similar to the way a BSI today is expected to manage the state of a
1964 transaction. For the immediate future, such capabilities are not expected and not required.

1965 **5.16 Where the ebXML Business Process Specification Schema May Be** 1966 **Implemented**

1967 The ebXML *Business Process Specification Schema* should be used wherever software is
1968 being specified to perform a role in an ebXML business collaboration. Specifically, the
1969 ebXML *Business Process Specification Schema* is intended to provide the business process
1970 and document specification for the formation of ebXML trading partner Collaboration
1971 Protocol Profiles and Agreements.

1972 However, the ebXML *Business Process Specification Schema* may be used to specify any
1973 electronic commerce collaboration. It may also be used for non-commerce collaborations,
1974 for instance in defining transactional collaborations among non-profit organizations or
1975 between applications, within the enterprise.

1976 Every BSI which is in the position of sending a signal or a document envelop shall verify if
1977 sending this message will violate the business transaction definitions and shall not send it if
1978 such a condition is detected. For instance sending a signal or a response after a timeout has
1979 occurred is prohibited. Similarly, sending a receipt on a document envelop which do not
1980 have the same digest as the original document envelop is prohibited. Rather, the BSI should
1981 send an exception back to the BSI that initiated the particular message.

1982 As of the current version, an ebXML compliant BSI is not requested that BSI be able to
1983 support multi-party collaboration. The current specification does not support the notions of
1984 context and correlation.

1985 **5.17 Guidelines for Business Service Interface Interoperability**

1986 We have taken great care in this new version of the specification to distinguish what is
1987 executable and computable versus general expressions written in text and associated with
1988 model elements. In particular, we exclude, beginsWhen, endsWhen, preCondition and
1989 postCondition from the responsibility of a BSI.

1990 Another important point for interoperability is that the context of a binary collaboration is
1991 limited to the document flows that are received or sent by the BSI. The BSI do not need to
1992 query information in other systems, internal or external to calculate the result of condition
1993 expressions.

1994 A BSI is required to support two forms of the ConditionExpression element: the XPath
1995 language, as well as the "DocumentEnvelopeNotation". An Xpath expression may involve
1996 the content of any DocumentEnvelope received prior to the transition within the scope of
1997 the current binary collaboration instance. The "DocumentEnvelopeNotation" is simply
1998 defined as the name or ID of a document envelope.

1999 **5.18 Collaboration and transaction well-formedness rules**

2000 The following rules should be used in addition to standard parsing to properly constrain the
2001 values of the attributes of the elements in an ebXML Business Process Specification.

2002 *Business Transaction*

2003 [0] If non-repudiation is required then the input or returned business document must be a
2004 tamper-detectable entity.

2005 [1] If authorization is required then the input business document and business signal must
2006 be an authenticated or a tamper detectable secure entity.

2007 [2] The time to acknowledge receipt must be less than the time to acknowledge processing
2008 if both properties have values.

2009 [3] If the time to acknowledge processing is null then the time to perform an activity must
2010 be greater than the time to acknowledge receipt.

2011 [4] The time to perform a transaction cannot be null unless it is specified to be request
2012 without a response.

2013 [5] If non-repudiation of receipt is required then the time to acknowledge receipt cannot be
2014 null.

2015 [6] The time to acknowledge receipt, time to acknowledge processing and time to perform
2016 cannot all be zero.

2017 *BusinessActivity*

2018 [7] Completion states must be defined on mutually exclusive paths guarantying that only
2019 one of the completion state will be reached.

2020 [8] A BusinessActivity may have any number of incoming transition but only one output
2021 transition. Either a Fork or Decision business states must be used to logically specify more
2022 than one outgoing transition.

2023 *Business Collaboration*

2024 [9] There must be at most one Start business state in a binary collaboration defintion.

2025 [10] There must be at least one Completion state in a binary collaboration definition

2026 [11] A Role cannot perform both roles of the same business transaction activity.

2027 [12] The two roles associated with a business collaboration must be different.

2028 **6.0 ebXML Business Process Specification Schema —**

2029 In this section we describe the XML Schema version of the Specification Schema.

- 2030 ● An example XML Business Process Specification listed in Appendix A
- 2031 ● A table listing all the elements with definitions and parent/child relationships
- 2032 ● A table listing all the elements, each with a cross reference to the corresponding class
- 2033 in the UML version of the specification schema
- 2034 ● Rules about namespaces and element references

2035 **6.1 Introduction**

2036 This section will document the Schema. The Schema has been derived from the UML
 2037 model. The correlation between the UML classes and Schema elements will be shown
 2038 separately later in this document.

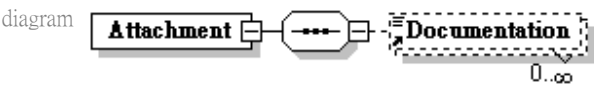
targetNamespace: <http://www.untmg.org/downloads/General/approved/BPSS-v1pt11.xsd>

2039

| Elements | Complex types | Simple types | Attr. groups |
|---|---|-------------------------|----------------------------------|
| Attachment | BusinessAction | GUID | documentSecurity |
| AttributeSubstitution | BusinessActivity | GUIDREF | name |
| BinaryCollaboration | BusinessTransactionType | | |
| BusinessDocument | PassInformation | | |
| BusinessPartnerRole | RoleType | | |
| BusinessTransaction | | | |
| BusinessTransactionActivity | | | |
| CollaborationActivity | | | |
| CommercialTransaction | | | |
| ConditionExpression | | | |
| Decision | | | |
| Documentation | | | |
| DocumentEnvelope | | | |
| DocumentSubstitution | | | |
| Failure | | | |
| Fork | | | |
| Include | | | |
| InformationDistribution | | | |
| Join | | | |
| MultiPartyCollaboration | | | |
| Namespace | | | |
| Namespaces | | | |
| Notification | | | |
| Package | | | |
| Performs | | | |
| ProcessSpecification | | | |
| QueryResponse | | | |
| RequestConfirm | | | |
| RequestingBusinessActivity | | | |

[RequestResponse](#)
[RespondingBusinessActivity](#)
[Start](#)
[SubstitutionSet](#)
[Success](#)
[TransactionInformation](#)
[Transition](#)
[TransitionInformation](#)

6.1.1 element Attachment



namespace <http://www.untmg.org/downloads/General/approved/BPSS-v1pt11.xsd>

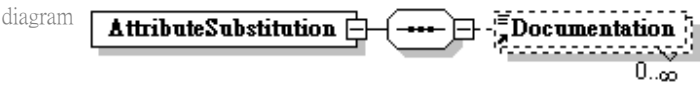
children [Documentation](#)

used by element [DocumentEnvelope](#)

| | | | | | | |
|----------------------|-----------------------|---------------|----------|----------|-------|------------|
| attributes | Name | Type | Use | Default | Fixed | Annotation |
| | name | xsd:string | optional | | | |
| | nameID | GUID | required | | | |
| | businessDocument | xsd:string | required | | | |
| | businessDocumentIDREF | GUIDREF | | | | |
| | specification | xsd:anyURI | | | | |
| | mimeType | xsd:string | optional | | | |
| | isAuthenticated | xsd:NMTOKEN | | | | |
| | isConfidential | xsd:NMTOKEN | | | | |
| | isTamperDetectable | xsd:NMTOKEN | | | | |
| identity constraints | Name | Refer | Selector | Field(s) | | |
| | unique | Attachment-ID | . | nameID | | |

```
source <xsd:element name="Attachment">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="Documentation" minOccurs="0" maxOccurs="unbounded"/>
    </xsd:sequence>
    <xsd:attributeGroup ref="name"/>
    <xsd:attribute name="businessDocument" type="xsd:string" use="required"/>
    <xsd:attribute name="businessDocumentIDREF" type="GUIDREF"/>
    <xsd:attribute name="specification" type="xsd:anyURI"/>
    <xsd:attribute name="mimeType" type="xsd:string" use="optional"/>
    <xsd:attributeGroup ref="documentSecurity"/>
  </xsd:complexType>
  <xsd:unique name="Attachment-ID">
    <xsd:selector xpath="."/>
    <xsd:field xpath="nameID"/>
  </xsd:unique>
</xsd:element>
```

2044 **6.1.2 element AttributeSubstitution**
2045



namespace <http://www.untnrg.org/downloads/General/approved/BPSS-v1pt11.xsd>

children [Documentation](#)

used by element [SubstitutionSet](#)

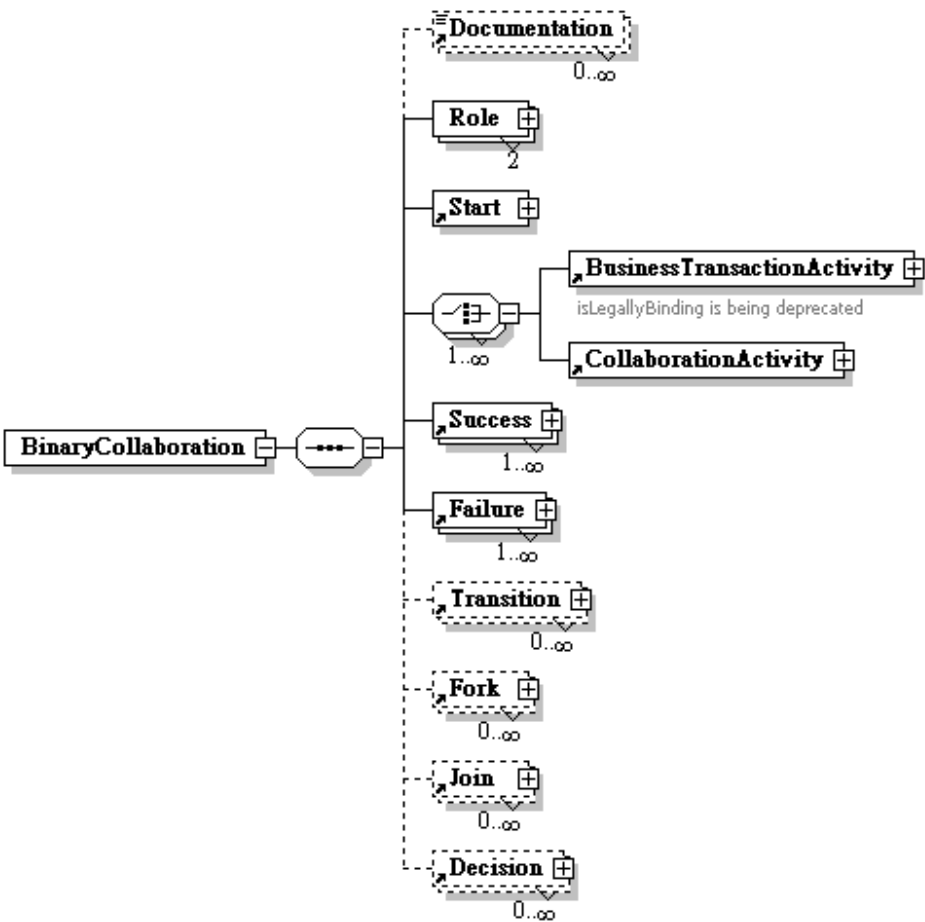
| attributes | Name | Type | Use | Default | Fixed | Annotation |
|------------|---------------|------------|----------|---------|-------|------------|
| | attributeName | xsd:string | required | | | |
| | value | xsd:string | required | | | |

source

```
<xsd:element name="AttributeSubstitution">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="Documentation" minOccurs="0" maxOccurs="unbounded"/>
    </xsd:sequence>
    <xsd:attribute name="attributeName" type="xsd:string" use="required"/>
    <xsd:attribute name="value" type="xsd:string" use="required"/>
  </xsd:complexType>
</xsd:element>
```

2046 6.1.3 element BinaryCollaboration
2047

diagram



namespace <http://www.untmg.org/downloads/General/approved/BPSS-v1pt11.xsd>

children [Documentation](#) [Role](#) [Start](#) [BusinessTransactionActivity](#) [CollaborationActivity](#) [Success](#) [Failure](#) [Transition](#) [Fork](#) [Join](#) [Decision](#)

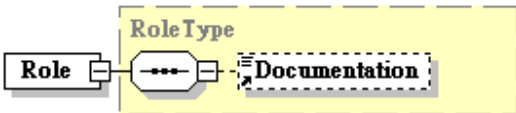
used by elements [Package](#) [ProcessSpecification](#)

| attributes | Name | Type | Use | Default | Fixed | Annotation |
|-------------|----------------------|----------------------------|----------|---------|----------|------------|
| | name | xsd:string | optional | | | |
| | nameID | GUID | required | | | |
| | pattern | xsd:anyURI | | | | |
| | beginsWhen | xsd:string | | | | |
| | endsWhen | xsd:string | | | | |
| | preCondition | xsd:string | | | | |
| | postCondition | xsd:string | | | | |
| | timeToPerform | xsd:duration | | | | |
| | initiatingRoleIDREF | GUIDREF | optional | | | |
| | isInnerCollaboration | xsd:boolean | | false | | |
| identity | Name | | Refer | | Selector | Field(s) |
| constraints | unique | BinaryCollaboration-ID | | | . | nameID |
| | unique | BinaryCollaborationRole-ID | | | ./Role | nameID |

source `<xsd:element name="BinaryCollaboration">`

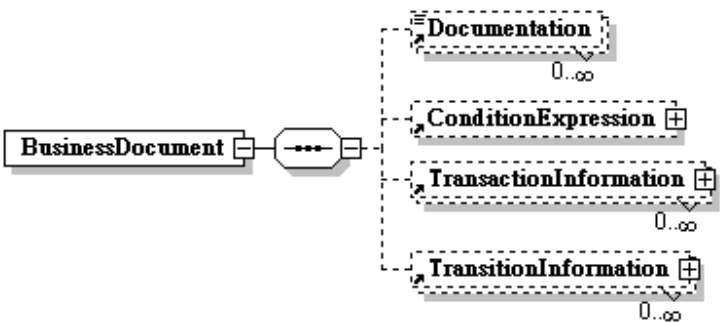
```
<xsd:complexType>
  <xsd:sequence>
    <xsd:element ref="Documentation" minOccurs="0" maxOccurs="unbounded"/>
    <xsd:element name="Role" type="RoleType" minOccurs="2" maxOccurs="2"/>
    <xsd:element ref="Start"/>
    <xsd:choice maxOccurs="unbounded">
      <xsd:element ref="BusinessTransactionActivity"/>
      <xsd:element ref="CollaborationActivity"/>
    </xsd:choice>
    <xsd:element ref="Success" maxOccurs="unbounded"/>
    <xsd:element ref="Failure" maxOccurs="unbounded"/>
    <xsd:element ref="Transition" minOccurs="0" maxOccurs="unbounded"/>
    <xsd:element ref="Fork" minOccurs="0" maxOccurs="unbounded"/>
    <xsd:element ref="Join" minOccurs="0" maxOccurs="unbounded"/>
    <xsd:element ref="Decision" minOccurs="0" maxOccurs="unbounded"/>
  </xsd:sequence>
  <xsd:attributeGroup ref="name"/>
  <xsd:attribute name="pattern" type="xsd:anyURI"/>
  <xsd:attribute name="beginsWhen" type="xsd:string"/>
  <xsd:attribute name="endsWhen" type="xsd:string"/>
  <xsd:attribute name="preCondition" type="xsd:string"/>
  <xsd:attribute name="postCondition" type="xsd:string"/>
  <xsd:attribute name="timeToPerform" type="xsd:duration"/>
  <xsd:attribute name="initiatingRoleIDREF" type="GUIDREF" use="optional"/>
  <xsd:attribute name="isInnerCollaboration" type="xsd:boolean" default="false"/>
</xsd:complexType>
<xsd:unique name="BinaryCollaboration-ID">
  <xsd:selector xpath="."/>
  <xsd:field xpath="nameID"/>
</xsd:unique>
<xsd:unique name="BinaryCollaborationRole-ID">
  <xsd:selector xpath="//Role"/>
  <xsd:field xpath="nameID"/>
</xsd:unique>
</xsd:element>
```

6.1.3.1 element BinaryCollaboration/Role

| | | | | | |
|------------|---|------------|----------|---------|------------|
| diagram |  | | | | |
| namespace | http://www.untmg.org/downloads/General/approved/BPSS-v1pt11.xsd | | | | |
| type | RoleType | | | | |
| children | Documentation | | | | |
| attributes | Name | Type | Use | Default | Annotation |
| | name | xsd:string | optional | | |
| | nameID | GUID | required | | |
| source | <xsd:element name="Role" type="RoleType" minOccurs="2" maxOccurs="2"/> | | | | |

2053 6.1.4 element BusinessDocument
2054

diagram



namespace <http://www.untmg.org/downloads/General/approved/BPSS-v1pt11.xsd>

children [Documentation](#) [ConditionExpression](#) [TransactionInformation](#) [TransitionInformation](#)

used by elements [Package](#) [ProcessSpecification](#)

| attributes | Name | Type | Use | Default | Fixed | Annotation |
|------------|-----------------------|--------------|----------|---------|-------|------------|
| | name | xsd:string | optional | | | |
| | nameID | GUID | required | | | |
| | specificationLocation | xsd:anyURI | | | | |
| | specificationID | xsd:anyURI | | | | |
| | namespacePrefixes | xsd:NMTOKENS | | | | |

source `<xsd:element name="BusinessDocument">
<xsd:complexType>
<xsd:sequence>
<xsd:element ref="Documentation" minOccurs="0" maxOccurs="unbounded"/>
<xsd:element ref="ConditionExpression" minOccurs="0"/>
<xsd:element ref="TransactionInformation" minOccurs="0" maxOccurs="unbounded"/>
<xsd:element ref="TransitionInformation" minOccurs="0" maxOccurs="unbounded"/>
</xsd:sequence>
<xsd:attributeGroup ref="name"/>
<xsd:attribute name="specificationLocation" type="xsd:anyURI"/>
<xsd:attribute name="specificationID" type="xsd:anyURI"/>
<xsd:attribute name="namespacePrefixes" type="xsd:NMTOKENS"/>
</xsd:complexType>
</xsd:element>`

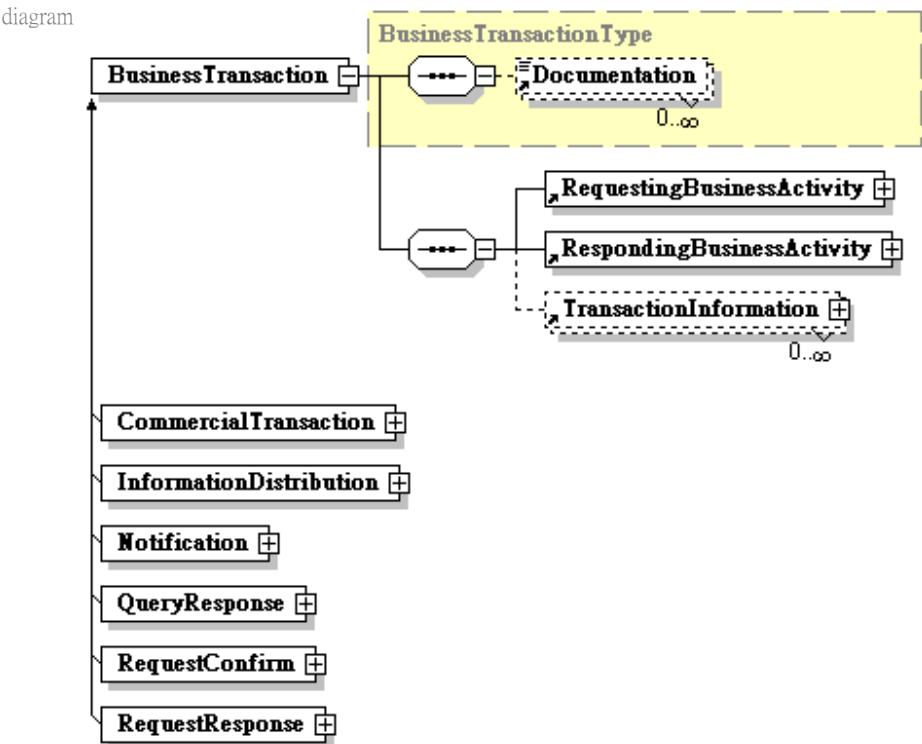
2055

2056 6.1.5 element BusinessPartnerRole
2057

| | | | | | | |
|----------------------|---|-----------------------------|----------|---------|------------|-----------------|
| diagram | | | | | | |
| namespace | http://www.untmg.org/downloads/General/approved/BPSS-v1pt11.xsd | | | | | |
| children | Documentation Performs Transition | | | | | |
| used by | element MultiPartyCollaboration | | | | | |
| attributes | Name | Type | Use | Default | Annotation | |
| | name | xsd:string | optional | | | |
| identity constraints | nameID | GUID | required | | | |
| | unique | Name BusinessPartnerRole-ID | | Refer | Selector . | Field(s) nameID |
| annotation | documentation Deprecated for v1.1 | | | | | |
| source | <pre><xsd:element name="BusinessPartnerRole"> <xsd:annotation> <xsd:documentation source="BPSS 1.1">Deprecated for v1.1</xsd:documentation> </xsd:annotation> <xsd:complexType> <xsd:sequence> <xsd:element ref="Documentation" minOccurs="0" maxOccurs="unbounded"/> <xsd:element ref="Performs" minOccurs="0" maxOccurs="unbounded"/> <xsd:element ref="Transition" minOccurs="0" maxOccurs="unbounded"/> </xsd:sequence> <xsd:attributeGroup ref="name"/> </xsd:complexType> <xsd:unique name="BusinessPartnerRole-ID"> <xsd:selector xpath="."/> <xsd:field xpath="nameID"/> </xsd:unique> </xsd:element></pre> | | | | | |

2058

2059 6.1.6 element BusinessTransaction
2060



namespace <http://www.untmg.org/downloads/General/approved/BPSS-v1pt10.xsd>

type extension of [BusinessTransactionType](#)

children [Documentation](#) [RequestingBusinessActivity](#) [RespondingBusinessActivity](#) [TransactionInformation](#)

used by elements [Package](#) [ProcessSpecification](#)

| attributes | Name | Type | Use | Default | Fixed | Annotation |
|------------|------------------------------|-------------|----------|---------|-------|------------|
| | name | xsd:string | optional | | | |
| | nameID | GUID | required | | | |
| | pattern | xsd:anyURI | | | | |
| | isGuaranteedDeliveryRequired | xsd:boolean | | false | | |

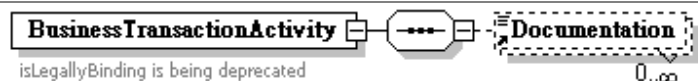
| identity constraints | Name | Refer | Selector | Field(s) |
|----------------------|------------------------|-------|----------|----------|
| unique | BusinessTransaction-ID | | . | nameID |

source

```
<xsd:element name="BusinessTransaction">
  <xsd:complexType>
    <xsd:complexContent>
      <xsd:extension base="BusinessTransactionType">
        <xsd:sequence>
          <xsd:element ref="RequestingBusinessActivity"/>
          <xsd:element ref="RespondingBusinessActivity"/>
          <xsd:element ref="TransactionInformation" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
      </xsd:extension>
    </xsd:complexContent>
  </xsd:complexType>
```

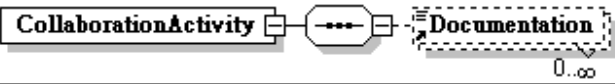
```
<xsd:unique name="BusinessTransaction-ID">
  <xsd:selector xpath="."/>
  <xsd:field xpath="nameID"/>
</xsd:unique>
</xsd:element>
```

6.1.7 element BusinessTransactionActivity

| | | | | | |
|------------|--|--------------|----------|---------|-----------------------------------|
| diagram |  | | | | |
| namespace | http://www.untmg.org/downloads/General/approved/BPSS-v1pt11.xsd | | | | |
| type | extension of BusinessActivity | | | | |
| children | Documentation | | | | |
| used by | element BinaryCollaboration | | | | |
| attributes | Name | Type | Use | Default | Annotation |
| | name | xsd:string | optional | | |
| | nameID | GUID | required | | |
| | fromRole | xsd:string | required | | |
| | fromRoleIDREF | GUIDREF | | | |
| | toRole | xsd:string | required | | |
| | toRoleIDREF | GUIDREF | | | |
| | beginsWhen | xsd:string | | | |
| | endsWhen | xsd:string | | | |
| | preCondition | xsd:string | | | |
| | postCondition | xsd:string | | | |
| | businessTransaction | xsd:string | required | | |
| | businessTransactionIDREF | GUIDREF | | | |
| | isConcurrent | xsd:boolean | | true | |
| | isLegallyBinding | xsd:boolean | | true | documentation Deprecated for v1.1 |
| | timeToPerform | xsd:duration | | | |
| annotation | documentation isLegallyBinding is being deprecated | | | | |
| source | <xsd:element name="BusinessTransactionActivity"> <xsd:annotation> <xsd:documentation>isLegallyBinding is being deprecated</xsd:documentation> </xsd:annotation> <xsd:complexType> <xsd:complexContent> <xsd:extension base="BusinessActivity"> <xsd:sequence> <xsd:element ref="Documentation" minOccurs="0" maxOccurs="unbounded"/> </xsd:sequence> <xsd:attribute name="businessTransaction" type="xsd:string" use="required"/> <xsd:attribute name="businessTransactionIDREF" type="GUIDREF"/> <xsd:attribute name="isConcurrent" type="xsd:boolean" default="true"/> <xsd:attribute name="isLegallyBinding" type="xsd:boolean" default="true"> <xsd:annotation> <xsd:documentation source="BPSS 1.1">Deprecated for v1.1</xsd:documentation> </xsd:annotation> </xsd:attribute> <xsd:attribute name="timeToPerform" type="xsd:duration"/> </xsd:extension> </xsd:complexContent> </xsd:complexType> </xsd:element> | | | | |

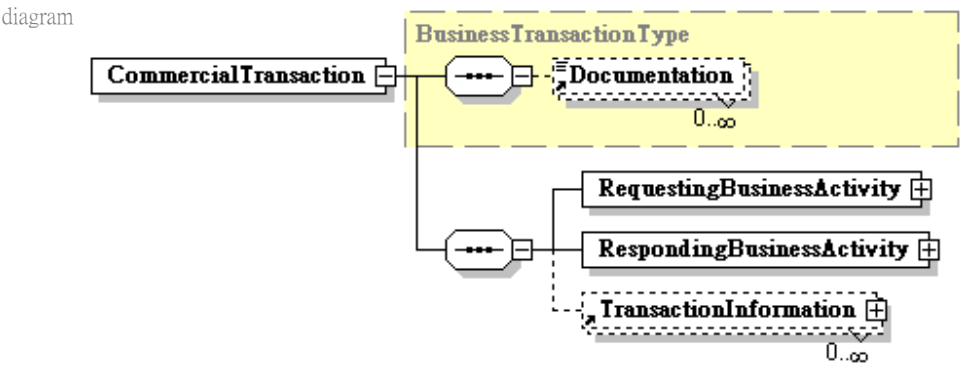
2066 **6.1.8 element CollaborationActivity**

2067

| | | | | | |
|------------|--|------------|----------|---------|------------|
| diagram |  | | | | |
| namespace | http://www.untmg.org/downloads/General/approved/BPSS-v1pt11.xsd | | | | |
| type | extension of BusinessActivity | | | | |
| children | Documentation | | | | |
| used by | element BinaryCollaboration | | | | |
| attributes | Name | Type | Use | Default | Annotation |
| | name | xsd:string | optional | | |
| | nameID | GUID | required | | |
| | fromRole | xsd:string | required | | |
| | fromRoleIDREF | GUIDREF | | | |
| | toRole | xsd:string | required | | |
| | toRoleIDREF | GUIDREF | | | |
| | beginsWhen | xsd:string | | | |
| | endsWhen | xsd:string | | | |
| | preCondition | xsd:string | | | |
| | postCondition | xsd:string | | | |
| | binaryCollaboration | xsd:string | required | | |
| | binaryCollaborationIDREF | GUIDREF | | | |
| source | <pre><xsd:element name="CollaborationActivity"> <xsd:complexType> <xsd:complexContent> <xsd:extension base="BusinessActivity"> <xsd:sequence> <xsd:element ref="Documentation" minOccurs="0" maxOccurs="unbounded"/> </xsd:sequence> <xsd:attribute name="binaryCollaboration" type="xsd:string" use="required"/> <xsd:attribute name="binaryCollaborationIDREF" type="GUIDREF"/> </xsd:extension> </xsd:complexContent> </xsd:complexType> </xsd:element></pre> | | | | |

2068

2069 **6.1.9 element CommercialTransaction**
2070



namespace <http://www.untmg.org/downloads/General/approved/BPSS-v1pt10.xsd>

type extension of [BusinessTransactionType](#)

children [Documentation](#) [RequestingBusinessActivity](#) [RespondingBusinessActivity](#) [TransactionInformation](#)

| attributes | Name | Type | Use | Default | Fixed | Annotation |
|------------|------------------------------|-------------|----------|---------|-------|------------|
| | name | xsd:string | optional | | | |
| | nameID | GUID | required | | | |
| | pattern | xsd:anyURI | | | | |
| | isGuaranteedDeliveryRequired | xsd:boolean | | false | | |

source

```
<xsd:element name="CommercialTransaction" substitutionGroup="BusinessTransaction">
  <xsd:complexType>
    <xsd:complexContent>
      <xsd:extension base="BusinessTransactionType">
        <xsd:sequence>
          <xsd:element name="RequestingBusinessActivity">
            <xsd:complexType>
              <xsd:complexContent>
                <xsd:restriction base="BusinessAction">
                  <xsd:sequence>
                    <xsd:element ref="Documentation" minOccurs="0" maxOccurs="unbounded"/>
                  </xsd:sequence>
                  <xsd:attribute name="isAuthorizationRequired" default="true"/>
                  <xsd:attribute name="isIntelligibleCheckRequired" default="true"/>
                  <xsd:attribute name="isNonRepudiationRequired" default="true"/>
                  <xsd:attribute name="isNonRepudiationReceiptRequired" default="true"/>
                  <xsd:attribute name="timeToAcknowledgeReceipt" default="P2H"/>
                  <xsd:attribute name="timeToAcknowledgeProcessing" default="P6H"/>
                </xsd:restriction>
              </xsd:complexContent>
            </xsd:complexType>
          </xsd:element>
          <xsd:element name="RespondingBusinessActivity">
            <xsd:complexType>
              <xsd:complexContent>
                <xsd:restriction base="BusinessAction">
                  <xsd:sequence>
                    <xsd:element ref="Documentation" minOccurs="0" maxOccurs="unbounded"/>
                  </xsd:sequence>
                </xsd:restriction>
              </xsd:complexContent>
            </xsd:complexType>
          </xsd:element>
          <xsd:element name="TransactionInformation">
            <xsd:complexType>
              <xsd:complexContent>
                <xsd:restriction base="BusinessAction">
                  <xsd:sequence>
                    <xsd:element ref="Documentation" minOccurs="0" maxOccurs="unbounded"/>
                  </xsd:sequence>
                </xsd:restriction>
              </xsd:complexContent>
            </xsd:complexType>
          </xsd:element>
        </xsd:sequence>
      </xsd:extension>
    </xsd:complexContent>
  </xsd:complexType>
</xsd:element>
```

```

    </xsd:sequence>
    <xsd:attribute name="isAuthorizationRequired" default="true"/>
    <xsd:attribute name="isIntelligibleCheckRequired" default="true"/>
    <xsd:attribute name="isNonRepudiationRequired" default="true"/>
    <xsd:attribute name="isNonRepudiationReceiptRequired" default="true"/>
    <xsd:attribute name="timeToAcknowledgeReceipt" default="P2H"/>
    <xsd:attribute name="timeToAcknowledgeProcessing" default="P6H"/>
  </xsd:restriction>
</xsd:complexContent>
</xsd:complexType>
</xsd:element>
<xsd:element ref="TransactionInformation" minOccurs="0" maxOccurs="unbounded"/>
</xsd:sequence>
</xsd:extension>
</xsd:complexContent>
</xsd:complexType>
</xsd:element>

```

6.1.9.1 element CommercialTransaction/RequestingBusinessActivity



namespace <http://www.untmg.org/downloads/General/approved/BPSS-v1pt10.xsd>

type restriction of [BusinessAction](#)

children [Documentation](#)

| | | | | | | |
|------------|---------------------------------|--------------|----------|---------|-------|------------|
| attributes | Name | Type | Use | Default | Fixed | Annotation |
| | name | xsd:string | optional | | | |
| | nameID | GUID | required | | | |
| | isAuthorizationRequired | xsd:boolean | | true | | |
| | isIntelligibleCheckRequired | xsd:boolean | | true | | |
| | isNonRepudiationRequired | xsd:boolean | | true | | |
| | isNonRepudiationReceiptRequired | xsd:boolean | | true | | |
| | timeToAcknowledgeReceipt | xsd:duration | | P2H | | |
| | timeToAcknowledgeProcessing | xsd:duration | | P6H | | |

```

source <xsd:element name="RequestingBusinessActivity">
  <xsd:complexType>
    <xsd:complexContent>
      <xsd:restriction base="BusinessAction">
        <xsd:sequence>
          <xsd:element ref="Documentation" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:attribute name="isAuthorizationRequired" default="true"/>
        <xsd:attribute name="isIntelligibleCheckRequired" default="true"/>
      </xsd:restriction>
    </xsd:complexContent>
  </xsd:complexType>
</xsd:element>

```

```
<xsd:attribute name="isNonRepudiationRequired" default="true"/>
<xsd:attribute name="isNonRepudiationReceiptRequired" default="true"/>
<xsd:attribute name="timeToAcknowledgeReceipt" default="P2H"/>
<xsd:attribute name="timeToAcknowledgeProcessing" default="P6H"/>
</xsd:restriction>
</xsd:complexContent>
</xsd:complexType>
</xsd:element>
```

2075
2076
2077
2078

6.1.9.2 element CommercialTransaction/RespondingBusinessActivity



namespace <http://www.untmg.org/downloads/General/approved/BPSS-v1pt10.xsd>

type restriction of [BusinessAction](#)

children [Documentation](#)

| attributes | Name | Type | Use | Default | Fixed | Annotation |
|------------|---------------------------------|--------------|----------|---------|-------|------------|
| | name | xsd:string | optional | | | |
| | nameID | GUID | required | | | |
| | isAuthorizationRequired | xsd:boolean | | true | | |
| | isIntelligibleCheckRequired | xsd:boolean | | true | | |
| | isNonRepudiationRequired | xsd:boolean | | true | | |
| | isNonRepudiationReceiptRequired | xsd:boolean | | true | | |
| | timeToAcknowledgeReceipt | xsd:duration | | P2H | | |
| | timeToAcknowledgeProcessing | xsd:duration | | P6H | | |

```
source <xsd:element name="RespondingBusinessActivity">
  <xsd:complexType>
    <xsd:complexContent>
      <xsd:restriction base="BusinessAction">
        <xsd:sequence>
          <xsd:element ref="Documentation" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:attribute name="isAuthorizationRequired" default="true"/>
        <xsd:attribute name="isIntelligibleCheckRequired" default="true"/>
        <xsd:attribute name="isNonRepudiationRequired" default="true"/>
        <xsd:attribute name="isNonRepudiationReceiptRequired" default="true"/>
        <xsd:attribute name="timeToAcknowledgeReceipt" default="P2H"/>
        <xsd:attribute name="timeToAcknowledgeProcessing" default="P6H"/>
      </xsd:restriction>
    </xsd:complexContent>
  </xsd:complexType>
</xsd:element>
```

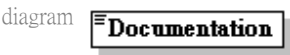
6.1.10 element ConditionExpression

| | | | | | |
|------------|---|------------|----------|---------|------------|
| diagram | | | | | |
| namespace | http://www.untmg.org/downloads/General/approved/BPSS-v1pt11.xsd | | | | |
| children | Documentation | | | | |
| used by | elements BusinessDocument Decision Failure Success Transition | | | | |
| attributes | Name | Type | Use | Default | Annotation |
| | expressionLanguage | xsd:string | required | | |
| | expression | xsd:string | required | | |
| | prefix | xsd:string | optional | | |
| source | <pre><xsd:element name="ConditionExpression"> <xsd:complexType> <xsd:sequence> <xsd:element ref="Documentation" minOccurs="0" maxOccurs="unbounded"/> </xsd:sequence> <xsd:attribute name="expressionLanguage" type="xsd:string" use="required"/> <xsd:attribute name="expression" type="xsd:string" use="required"/> <xsd:attribute name="prefix" type="xsd:string" use="optional"/> </xsd:complexType> </xsd:element></pre> | | | | |

6.1.11 element Decision

| | | | | | |
|----------------------|---|------------------|----------|----------|-----------------|
| diagram | | | | | |
| namespace | http://www.untmg.org/downloads/General/approved/BPSS-v1pt11.xsd | | | | |
| children | ConditionExpression | | | | |
| used by | element BinaryCollaboration | | | | |
| attributes | Name | Type | Use | Default | Annotation |
| | name | xsd:string | optional | | |
| | nameID | GUID | required | | |
| identity constraints | unique | Name Decision-ID | Refer | Selector | Field(s) nameID |
| source | <pre><xsd:element name="Decision"> <xsd:complexType> <xsd:sequence> <xsd:element ref="ConditionExpression"/> </xsd:sequence> <xsd:attributeGroup ref="name"/> </xsd:complexType> <xsd:unique name="Decision-ID"> <xsd:selector xpath="."/> <xsd:field xpath="nameID"/> </xsd:unique> </xsd:element></pre> | | | | |

2086 **6.1.12 element Documentation**
2087



namespace `http://www.untmg.org/downloads/General/approved/BPSS-v1pt11.xsd`

type extension of `xsd:string`

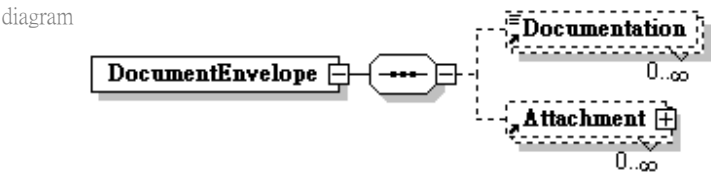
used by elements [Attachment](#) [AttributeSubstitution](#) [BinaryCollaboration](#) [BusinessDocument](#) [BusinessPartnerRole](#) [BusinessRules](#) [BusinessTransactionActivity](#) [CollaborationActivity](#) [ConditionExpression](#) [DocumentEnvelope](#) [DocumentSubstitution](#) [Failure](#) [Fork](#) [Include](#) [Join](#) [MultiPartyCollaboration](#) [Package](#) [Performs](#) [ProcessSpecification](#) [RequestResponse/RequestingBusinessActivity](#) [RequestConfirm/RequestingBusinessActivity](#) [QueryResponse/RequestingBusinessActivity](#) [CommercialTransaction/RequestingBusinessActivity](#) [Notification/RequestingBusinessActivity](#) [InformationDistribution/RequestingBusinessActivity](#) [RequestResponse/RespondingBusinessActivity](#) [RequestConfirm/RespondingBusinessActivity](#) [QueryResponse/RespondingBusinessActivity](#) [CommercialTransaction/RespondingBusinessActivity](#) [Start](#) [SubstitutionSet](#) [Success](#) [TransactionInformation](#) [Transition](#) [TransitionInformation](#) [BusinessAction](#) [BusinessTransactionType](#) [RoleType](#)

| attributes | complexTypes | | | | | |
|------------|--------------|------------|-----|---------|-------|------------|
| | Name | Type | Use | Default | Fixed | Annotation |
| | uri | xsd:anyURI | | | | |

```
source <xsd:element name="Documentation">
  <xsd:complexType>
    <xsd:simpleContent>
      <xsd:extension base="xsd:string">
        <xsd:attribute name="uri" type="xsd:anyURI"/>
      </xsd:extension>
    </xsd:simpleContent>
  </xsd:complexType>
</xsd:element>
```

2088

2089 6.1.13 element DocumentEnvelope
2090



namespace <http://www.untmg.org/downloads/General/approved/BPSS-v1pt11.xsd>

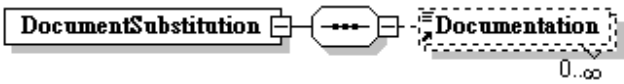
children [Documentation](#) [Attachment](#)

| | | | | | | |
|-------------|-----------------------|---------------------|----------|----------|-------|------------|
| attributes | Name | Type | Use | Default | Fixed | Annotation |
| | name | xsd:string | optional | | | |
| | nameID | GUID | required | | | |
| | businessDocument | xsd:string | required | | | |
| | businessDocumentIDREF | GUIDREF | | | | |
| | isPositiveResponse | xsd:boolean | | | | |
| | isAuthenticated | xsd:NMTOKEN | | | | |
| | isConfidential | xsd:NMTOKEN | | | | |
| identity | isTamperDetectable | xsd:NMTOKEN | | | | |
| | | | | | | |
| constraints | Name | Refer | Selector | Field(s) | | |
| | unique | DocumentEnvelope-ID | . | nameID | | |

source `<xsd:element name="DocumentEnvelope">`
 `<xsd:complexType>`
 `<xsd:sequence>`
 `<xsd:element ref="Documentation" minOccurs="0" maxOccurs="unbounded"/>`
 `<xsd:element ref="Attachment" minOccurs="0" maxOccurs="unbounded"/>`
 `</xsd:sequence>`
 `<xsd:attributeGroup ref="name"/>`
 `<xsd:attribute name="businessDocument" type="xsd:string" use="required"/>`
 `<xsd:attribute name="businessDocumentIDREF" type="GUIDREF"/>`
 `<xsd:attribute name="isPositiveResponse" type="xsd:boolean"/>`
 `<xsd:attributeGroup ref="documentSecurity"/>`
 `</xsd:complexType>`
 `<xsd:unique name="DocumentEnvelope-ID">`
 `<xsd:selector xpath="."/>`
 `<xsd:field xpath="nameID"/>`
 `</xsd:unique>`
`</xsd:element>`

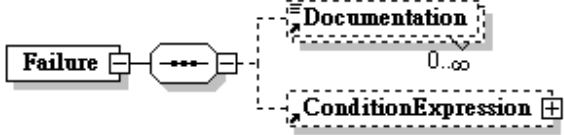
2091

2092 **6.1.14 element DocumentSubstitution**
2093

| | | | | | |
|------------|---|------------|----------|---------|------------|
| diagram |  | | | | |
| namespace | http://www.untmg.org/downloads/General/approved/BPSS-v1pt11.xsd | | | | |
| children | Documentation | | | | |
| used by | element SubstitutionSet | | | | |
| attributes | Name | Type | Use | Default | Annotation |
| | originalBusinessDocument | xsd:string | required | | |
| | originalBusinessDocumentID | GUIDREF | | | |
| | substituteBusinessDocumentLocation | xsd:anyURI | required | | |
| | substituteBusinessDocumentId | xsd:anyURI | | | |
| source | <pre><xsd:element name="DocumentSubstitution"> <xsd:complexType> <xsd:sequence> <xsd:element ref="Documentation" minOccurs="0" maxOccurs="unbounded"/> </xsd:sequence> <xsd:attribute name="originalBusinessDocument" type="xsd:string" use="required"/> <xsd:attribute name="originalBusinessDocumentID" type="GUIDREF"/> <xsd:attribute name="substituteBusinessDocumentLocation" type="xsd:anyURI" use="required"/> <xsd:attribute name="substituteBusinessDocumentId" type="xsd:anyURI"/> </xsd:complexType> </xsd:element></pre> | | | | |

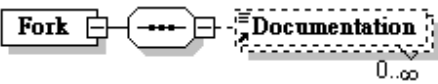
2094

2095 **6.1.15 element Failure**
2096

| | | | | | |
|-------------|--|-------------|----------|----------|------------|
| diagram |  | | | | |
| namespace | http://www.untmg.org/downloads/General/approved/BPSS-v1pt11.xsd | | | | |
| children | Documentation ConditionExpression | | | | |
| used by | element BinaryCollaboration | | | | |
| attributes | Name | Type | Use | Default | Annotation |
| | name | xsd:string | optional | | |
| | nameID | GUID | required | | |
| | fromBusinessState | xsd:string | required | | |
| | fromBusinessStateIDREF | GUIDREF | | | |
| | conditionGuard | xsd:NMTOKEN | | | |
| identity | Name | Refer | Selector | Field(s) | |
| constraints | unique | Failure-ID | . | nameID | |
| source | <pre><xsd:element name="Failure"> <xsd:complexType> <xsd:sequence> <xsd:element ref="Documentation" minOccurs="0" maxOccurs="unbounded"/> <xsd:element ref="ConditionExpression" minOccurs="0"/> </xsd:sequence> <xsd:attributeGroup ref="name"/> <xsd:attribute name="fromBusinessState" type="xsd:string" use="required"/> <xsd:attribute name="fromBusinessStateIDREF" type="GUIDREF"/> <xsd:attribute name="conditionGuard"/> <xsd:simpleType> <xsd:restriction base="xsd:NMTOKEN"> <xsd:enumeration value="ProtocolSuccess"/> <xsd:enumeration value="AnyProtocolFailure"/> <xsd:enumeration value="RequestReceiptFailure"/> <xsd:enumeration value="RequestProcessingFailure"/> <xsd:enumeration value="ResponseReceiptFailure"/> <xsd:enumeration value="ResponseProcessingFailure"/> <xsd:enumeration value="SignalTimeout"/> <xsd:enumeration value="ResponseTimeout"/> <xsd:enumeration value="BusinessSuccess"/> <xsd:enumeration value="BusinessFailure"/> <xsd:enumeration value="Success"/> <xsd:enumeration value="Failure"/> </xsd:restriction> </xsd:simpleType> </xsd:complexType> <xsd:unique name="Failure-ID"> <xsd:selector xpath="."/> <xsd:field xpath="nameID"/> </xsd:unique> </xsd:element></pre> | | | | |

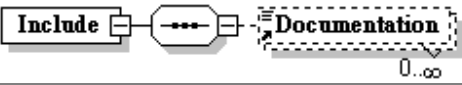
2097

2098 **6.1.16 element Fork**
2099

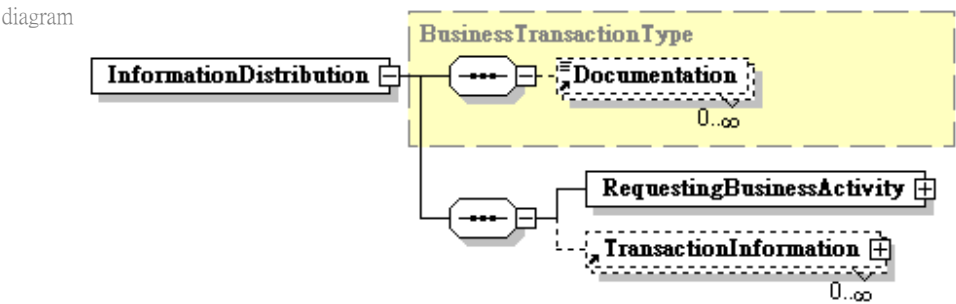
| | | | | | |
|-------------|---|--------------|----------|----------|------------|
| diagram |  | | | | |
| namespace | http://www.untmg.org/downloads/General/approved/BPSS-v1pt11.xsd | | | | |
| children | Documentation | | | | |
| used by | element BinaryCollaboration | | | | |
| attributes | Name | Type | Use | Default | Annotation |
| | name | xsd:string | optional | | |
| | nameID | GUID | required | | |
| | type | xsd:NMTOKEN | optional | OR | |
| | timeToPerform | xsd:duration | optional | | |
| identity | Name | Refer | Selector | Field(s) | |
| constraints | unique | Fork-ID | . | nameID | |
| source | <pre><xsd:element name="Fork"> <xsd:complexType> <xsd:sequence> <xsd:element ref="Documentation" minOccurs="0" maxOccurs="unbounded"/> </xsd:sequence> <xsd:attributeGroup ref="name"/> <xsd:attribute name="type" use="optional" default="OR"> <xsd:simpleType> <xsd:restriction base="xsd:NMTOKEN"> <xsd:enumeration value="OR"/> <xsd:enumeration value="XOR"/> </xsd:restriction> </xsd:simpleType> </xsd:attribute> <xsd:attribute name="timeToPerform" type="xsd:duration" use="optional"/> </xsd:complexType> <xsd:unique name="Fork-ID"> <xsd:selector xpath="."/> <xsd:field xpath="nameID"/> </xsd:unique> </xsd:element></pre> | | | | |

2100

6.1.17 element Include

| | | | | | |
|----------------------|---|------------|----------|----------|------------|
| diagram |  | | | | |
| namespace | http://www.untmg.org/downloads/General/approved/BPSS-v1pt11.xsd | | | | |
| children | Documentation | | | | |
| used by | elements Package ProcessSpecification | | | | |
| attributes | Name | Type | Use | Default | Annotation |
| | name | xsd:string | optional | | |
| | nameID | GUID | required | | |
| | uri | xsd:anyURI | required | | |
| | version | xsd:string | required | | |
| identity constraints | unique | Name | Refer | Selector | Field(s) |
| | | Include-ID | | . | nameID |
| source | <pre> <xsd:element name="Include"> <xsd:complexType> <xsd:sequence> <xsd:element ref="Documentation" minOccurs="0" maxOccurs="unbounded"/> </xsd:sequence> <xsd:attributeGroup ref="name"/> <xsd:attribute name="uri" type="xsd:anyURI" use="required"/> <xsd:attribute name="version" type="xsd:string" use="required"/> </xsd:complexType> <xsd:unique name="Include-ID"> <xsd:selector xpath="."/> <xsd:field xpath="nameID"/> </xsd:unique> </xsd:element> </pre> | | | | |

2104 6.1.18 element InformationDistribution
2105



namespace <http://www.untmg.org/downloads/General/approved/BPSS-v1pt10.xsd>

type extension of [BusinessTransactionType](#)

children [Documentation](#) [RequestingBusinessActivity](#) [TransactionInformation](#)

| attributes | Name | Type | Use | Default | Fixed | Annotation |
|------------|------------------------------|-------------|----------|---------|-------|------------|
| | name | xsd:string | optional | | | |
| | nameID | GUID | required | | | |
| | pattern | xsd:anyURI | | | | |
| | isGuaranteedDeliveryRequired | xsd:boolean | | false | | |

source

```
<xsd:element name="InformationDistribution" substitutionGroup="BusinessTransaction">
  <xsd:complexType>
    <xsd:complexContent>
      <xsd:extension base="BusinessTransactionType">
        <xsd:sequence>
          <xsd:element name="RequestingBusinessActivity">
            <xsd:complexType>
              <xsd:complexContent>
                <xsd:restriction base="BusinessAction">
                  <xsd:sequence>
                    <xsd:element ref="Documentation" minOccurs="0" maxOccurs="unbounded"/>
                  </xsd:sequence>
                  <xsd:attribute name="timeToAcknowledgeReceipt" default="P24H"/>
                </xsd:restriction>
              </xsd:complexContent>
            </xsd:complexType>
          </xsd:element>
          <xsd:element ref="TransactionInformation" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
      </xsd:extension>
    </xsd:complexContent>
  </xsd:complexType>
</xsd:element>
```

2106

2107

2108

6.1.18.1 element InformationDistribution/RequestingBusinessActivity



namespace <http://www.untmg.org/downloads/General/approved/BPSS-v1pt10.xsd>

type restriction of [BusinessAction](#)

children [Documentation](#)

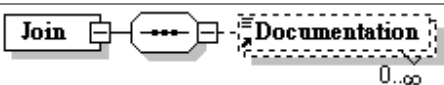
| attributes | Name | Type | Use | Default | Fixed | Annotation |
|------------|---------------------------------|--------------|----------|---------|-------|---------------------------|
| | name | xsd:string | optional | | | |
| | nameID | GUID | required | | | |
| | isAuthorizationRequired | xsd:boolean | | false | | docume deprecated ntation |
| | isIntelligibleCheckRequired | xsd:boolean | | false | | |
| | isNonRepudiationRequired | xsd:boolean | | false | | |
| | isNonRepudiationReceiptRequired | xsd:boolean | | false | | |
| | timeToAcknowledgeReceipt | xsd:duration | | P24H | | |
| | timeToAcknowledgeProcessing | xsd:duration | | | | |

```
source <xsd:element name="RequestingBusinessActivity">
  <xsd:complexType>
    <xsd:complexContent>
      <xsd:restriction base="BusinessAction">
        <xsd:sequence>
          <xsd:element ref="Documentation" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:attribute name="timeToAcknowledgeReceipt" default="P24H"/>
      </xsd:restriction>
    </xsd:complexContent>
  </xsd:complexType>
</xsd:element>
```

2109

2110 6.1.19 element Join

2111

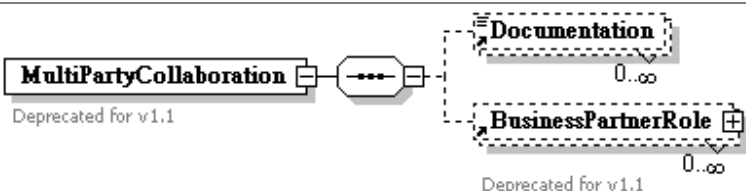
| | | | | | |
|----------------------|--|-------------|----------|----------|------------|
| diagram |  | | | | |
| namespace | http://www.untmg.org/downloads/General/approved/BPSS-v1pt11.xsd | | | | |
| children | Documentation | | | | |
| used by | element BinaryCollaboration | | | | |
| attributes | Name | Type | Use | Default | Annotation |
| | name | xsd:string | optional | | |
| | nameID | GUID | required | | |
| | waitForAll | xsd:boolean | | true | |
| identity constraints | unique | Name | Refer | Selector | Field(s) |
| | | Join-ID | | . | nameID |
| source | <pre> <xsd:element name="Join"> <xsd:complexType> <xsd:sequence> <xsd:element ref="Documentation" minOccurs="0" maxOccurs="unbounded"/> </xsd:sequence> <xsd:attributeGroup ref="name"/> <xsd:attribute name="waitForAll" type="xsd:boolean" default="true"/> </xsd:complexType> <xsd:unique name="Join-ID"> <xsd:selector xpath="."/> <xsd:field xpath="nameID"/> </xsd:unique> </xsd:element> </pre> | | | | |

2112

2113


2114 6.1.20 element MultiPartyCollaboration

2115

| | | | | | |
|----------------------|---|----------------------------|----------|----------|------------|
| diagram |  | | | | |
| namespace | http://www.untmg.org/downloads/General/approved/BPSS-v1pt11.xsd | | | | |
| children | Documentation BusinessPartnerRole | | | | |
| used by | elements Package ProcessSpecification | | | | |
| attributes | Name | Type | Use | Default | Annotation |
| | name | xsd:string | optional | | |
| | nameID | GUID | required | | |
| identity constraints | unique | Name | Refer | Selector | Field(s) |
| | | MultiPartyCollaboration-ID | | . | nameID |
| annotation | documentation Deprecated for v1.1 | | | | |
| source | <pre> <xsd:element name="MultiPartyCollaboration"> <xsd:annotation> <xsd:documentation source="BPSS 1.1">Deprecated for v1.1</xsd:documentation> </xsd:annotation> <xsd:complexType> <xsd:sequence> <xsd:element ref="Documentation" minOccurs="0" maxOccurs="unbounded"/> <xsd:element ref="BusinessPartnerRole" minOccurs="0" maxOccurs="unbounded"/> </xsd:sequence> <xsd:attributeGroup ref="name"/> </xsd:complexType> </xsd:element> </pre> | | | | |

```
</xsd:complexType>
<xsd:unique name="MultiPartyCollaboration-ID">
  <xsd:selector xpath="."/>
  <xsd:field xpath="nameID"/>
</xsd:unique>
</xsd:element>
```

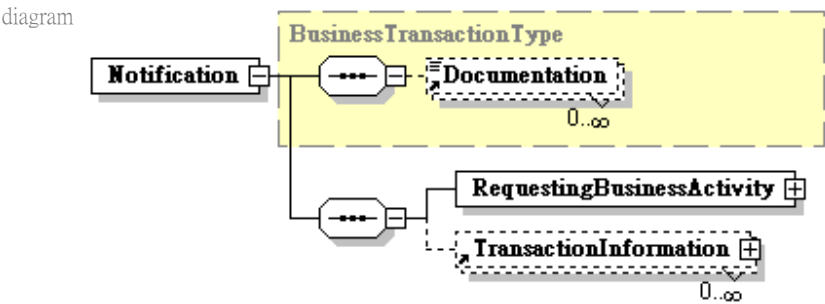
6.1.21 element Namespace

| | | | | | |
|------------|--|-------------|----------|---------|------------|
| diagram |  | | | | |
| namespace | http://www.untmg.org/downloads/General/approved/BPSS-v1pt11.xsd | | | | |
| children | Namespace | | | | |
| used by | elements Namespace Namespaces | | | | |
| attributes | Name | Type | Use | Default | Annotation |
| | URI | xsd:anyURI | required | | |
| | prefix | xsd:NMTOKEN | required | | |
| | nameID | GUID | required | | |
| source | <pre><xsd:element name="Namespace"> <xsd:complexType> <xsd:sequence> <xsd:annotation> <xsd:documentation>alternative namespaces</xsd:documentation> </xsd:annotation> <xsd:element ref="Namespace" minOccurs="0" maxOccurs="unbounded"/> </xsd:sequence> <xsd:attribute name="URI" type="xsd:anyURI" use="required"/> <xsd:attribute name="prefix" type="xsd:NMTOKEN" use="required"/> <xsd:attribute name="nameID" type="GUID" use="required"/> </xsd:complexType> </xsd:element></pre> | | | | |

6.1.22 element Namespaces

| | |
|-----------|--|
| diagram | |
| namespace | http://www.untmg.org/downloads/General/approved/BPSS-v1pt11.xsd |
| children | Namespace |
| used by | elements Package ProcessSpecification |
| source | <pre><xsd:element name="Namespaces"> <xsd:complexType> <xsd:sequence> <xsd:element ref="Namespace" maxOccurs="unbounded"/> </xsd:sequence> </xsd:complexType> </xsd:element></pre> |

2125 6.1.23 element Notification
2126



namespace <http://www.untmg.org/downloads/General/approved/BPSS-v1pt10.xsd>

type extension of [BusinessTransactionType](#)

children [Documentation](#) [RequestingBusinessActivity](#) [TransactionInformation](#)

| attributes | Name | Type | Use | Default | Fixed | Annotation |
|------------|------------------------------|-------------|----------|---------|-------|------------|
| | name | xsd:string | optional | | | |
| | nameID | GUID | required | | | |
| | pattern | xsd:anyURI | | | | |
| | isGuaranteedDeliveryRequired | xsd:boolean | | false | | |

```
source <xsd:element name="Notification" substitutionGroup="BusinessTransaction">
  <xsd:complexType>
    <xsd:complexContent>
      <xsd:extension base="BusinessTransactionType">
        <xsd:sequence>
          <xsd:element name="RequestingBusinessActivity">
            <xsd:complexType>
              <xsd:complexContent>
                <xsd:restriction base="BusinessAction">
                  <xsd:sequence>
                    <xsd:element ref="Documentation" minOccurs="0" maxOccurs="unbounded"/>
                  </xsd:sequence>
                  <xsd:attribute name="timeToAcknowledgeReceipt" default="P24H"/>
                  <xsd:attribute name="isNonRepudiationReceiptRequired" default="true"/>
                </xsd:restriction>
              </xsd:complexContent>
            </xsd:complexType>
          </xsd:element>
          <xsd:element ref="TransactionInformation" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
      </xsd:extension>
    </xsd:complexContent>
  </xsd:complexType>
</xsd:element>
```

2127

2128 **6.1.23.1 element Notification/RequestingBusinessActivity**
2129



namespace <http://www.untmg.org/downloads/General/approved/BPSS-v1pt10.xsd>

type restriction of [BusinessAction](#)

children [Documentation](#)

| attributes | Name | Type | Use | Default | Fixed | Annotation |
|------------|---------------------------------|--------------|----------|---------|-------|---------------------|
| | name | xsd:string | optional | | | |
| | nameID | GUID | required | | | |
| | isAuthorizationRequired | xsd:boolean | | false | | document deprecated |
| | isIntelligibleCheckRequired | xsd:boolean | | false | | |
| | isNonRepudiationReceiptRequired | xsd:boolean | | false | | |
| | isNonRepudiationReceiptRequired | xsd:boolean | | true | | |
| | timeToAcknowledgeReceipt | xsd:duration | | P24H | | |
| | timeToAcknowledgeProcessing | xsd:duration | | | | |

```
source <xsd:element name="RequestingBusinessActivity">
  <xsd:complexType>
    <xsd:complexContent>
      <xsd:restriction base="BusinessAction">
        <xsd:sequence>
          <xsd:element ref="Documentation" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:attribute name="timeToAcknowledgeReceipt" default="P24H"/>
        <xsd:attribute name="isNonRepudiationReceiptRequired" default="true"/>
      </xsd:restriction>
    </xsd:complexContent>
  </xsd:complexType>
</xsd:element>
```

2130

6.1.24 element Package

| diagram | <p>Deprecated for v1.1</p> | | | | | | | | | | | | | | | |
|----------------------|--|----------|----------|------------|----------|------------|------|------------|----------|---|--------|--------|------|----------|--|--|
| namespace | http://www.untmg.org/downloads/General/approved/BPSS-v1pt11.xsd | | | | | | | | | | | | | | | |
| children | Documentation Include SubstitutionSet Namespaces Package BusinessDocument BusinessTransaction BinaryCollaboration MultiPartyCollaboration | | | | | | | | | | | | | | | |
| used by | elements Package ProcessSpecification | | | | | | | | | | | | | | | |
| attributes | <table><tr><th>Name</th><th>Type</th><th>Use</th><th>Default</th><th>Annotation</th></tr><tr><td>name</td><td>xsd:string</td><td>optional</td><td></td><td></td></tr><tr><td>nameID</td><td>GUID</td><td>required</td><td></td><td></td></tr></table> | Name | Type | Use | Default | Annotation | name | xsd:string | optional | | | nameID | GUID | required | | |
| Name | Type | Use | Default | Annotation | | | | | | | | | | | | |
| name | xsd:string | optional | | | | | | | | | | | | | | |
| nameID | GUID | required | | | | | | | | | | | | | | |
| identity constraints | <table><tr><th>unique</th><th>Name</th><th>Refer</th><th>Selector</th><th>Field(s)</th></tr><tr><td></td><td>Package-ID</td><td></td><td>.</td><td>nameID</td></tr></table> | unique | Name | Refer | Selector | Field(s) | | Package-ID | | . | nameID | | | | | |
| unique | Name | Refer | Selector | Field(s) | | | | | | | | | | | | |
| | Package-ID | | . | nameID | | | | | | | | | | | | |
| source | <pre><xsd:element name="Package"> <xsd:complexType> <xsd:sequence> <xsd:element ref="Documentation" minOccurs="0" maxOccurs="unbounded"/> <xsd:element ref="Include" minOccurs="0" maxOccurs="unbounded"/> <xsd:element ref="SubstitutionSet" minOccurs="0"/> <xsd:element ref="Namespaces" minOccurs="0"/> <xsd:choice minOccurs="0" maxOccurs="unbounded"> <xsd:element ref="Package" minOccurs="0" maxOccurs="unbounded"/> <xsd:element ref="BusinessDocument" minOccurs="0" maxOccurs="unbounded"/> <xsd:element ref="BusinessTransaction" minOccurs="0" maxOccurs="unbounded"/> <xsd:element ref="BinaryCollaboration" minOccurs="0" maxOccurs="unbounded"/> <xsd:element ref="MultiPartyCollaboration" minOccurs="0" maxOccurs="unbounded"/> </xsd:choice> </xsd:sequence> <xsd:attributeGroup ref="name"/> </xsd:complexType> <xsd:unique name="Package-ID"> <xsd:selector xpath="."/> <xsd:field xpath="nameID"/> </xsd:unique> </element></pre> | | | | | | | | | | | | | | | |

2133
2134
2135
2136

| | |
|--|---|
| | <code></xsd:unique></code> <code></xsd:element></code> |
|--|---|

6.1.25 element Performs

| | | | | | |
|----------------------|---|-------------|----------|----------|------------|
| diagram | | | | | |
| namespace | http://www.untmg.org/downloads/General/approved/BPSS-v1pt11.xsd | | | | |
| children | Documentation | | | | |
| used by | element BusinessPartnerRole | | | | |
| attributes | Name | Type | Use | Default | Annotation |
| | name | xsd:string | optional | | |
| | nameID | GUID | required | | |
| | role | xsd:string | required | | |
| | roleIDREF | GUIDREF | required | | |
| identity constraints | unique | Name | Refer | Selector | Field(s) |
| | | Performs-ID | | . | nameID |
| annotation | documentation Deprecated for v1.1 | | | | |
| source | <pre><xsd:element name="Performs"> <xsd:annotation> <xsd:documentation source="BPSS 1.1">Deprecated for v1.1</xsd:documentation> </xsd:annotation> <xsd:complexType> <xsd:sequence> <xsd:element ref="Documentation" minOccurs="0" maxOccurs="unbounded"/> </xsd:sequence> <xsd:attributeGroup ref="name"/> <xsd:attribute name="role" type="xsd:string" use="required"/> <xsd:attribute name="roleIDREF" type="GUIDREF" use="required"/> </xsd:complexType> <xsd:unique name="Performs-ID"> <xsd:selector xpath="."/> <xsd:field xpath="nameID"/> </xsd:unique> </xsd:element></pre> | | | | |

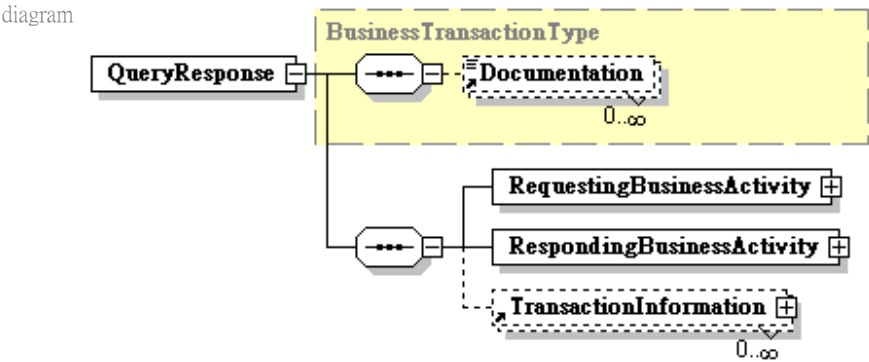
2137

2138 **6.1.26 element ProcessSpecification**
 2139

| | | | | | | |
|----------------------|---|------------|-------------------------|---------|------------|----------|
| diagram | <p>Diagram illustrating the structure of the ProcessSpecification element. It is a complex type containing a sequence of elements: Documentation, Include, SubstitutionSet, and Namespaces. The Namespaces element is followed by a choice of elements: Package, BusinessDocument, BusinessTransaction, BinaryCollaboration, and MultiPartyCollaboration. All elements have a cardinality of 0..∞. The MultiPartyCollaboration element is marked as deprecated for v1.1.</p> | | | | | |
| namespace | http://www.untmg.org/downloads/General/approved/BPSS-v1pt11.xsd | | | | | |
| children | Documentation Include SubstitutionSet Namespaces Package BusinessDocument BusinessTransaction BinaryCollaboration MultiPartyCollaboration | | | | | |
| attributes | Name | Type | Use | Default | Annotation | |
| | name | xsd:string | optional | | | |
| | nameID | GUID | required | | | |
| | version | xsd:string | required | | | |
| identity constraints | unique | Name | ProcessSpecification-ID | | Refer | Selector |
| source | <pre><xsd:element name="ProcessSpecification"> <xsd:complexType> <xsd:sequence> <xsd:element ref="Documentation" minOccurs="0" maxOccurs="unbounded"/> <xsd:element ref="Include" minOccurs="0" maxOccurs="unbounded"/> <xsd:element ref="SubstitutionSet" minOccurs="0"/> <xsd:element ref="Namespaces" minOccurs="0"/> <xsd:choice minOccurs="0" maxOccurs="unbounded"> <xsd:element ref="Package" minOccurs="0" maxOccurs="unbounded"/> <xsd:element ref="BusinessDocument" minOccurs="0" maxOccurs="unbounded"/> <xsd:element ref="BusinessTransaction" minOccurs="0" maxOccurs="unbounded"/> <xsd:element ref="BinaryCollaboration" minOccurs="0" maxOccurs="unbounded"/> <xsd:element ref="MultiPartyCollaboration" minOccurs="0" maxOccurs="unbounded"/> </xsd:choice> </xsd:sequence> <xsd:attributeGroup ref="name"/> <xsd:attribute name="version" type="xsd:string" use="required"/> </xsd:complexType> <xsd:unique name="ProcessSpecification-ID"> <xsd:selector xpath="."/> <xsd:field xpath="nameID"/> </xsd:unique> </element></pre> | | | | | |

</xsd:unique>
</xsd:element>

6.1.27 element QueryResponse



namespace <http://www.untmg.org/downloads/General/approved/BPSS-v1pt10.xsd>

type extension of [BusinessTransactionType](#)

children [Documentation](#) [RequestingBusinessActivity](#) [RespondingBusinessActivity](#) [TransactionInformation](#)

| | | | | | | |
|------------|------------------------------|-------------|----------|---------|-------|------------|
| attributes | Name | Type | Use | Default | Fixed | Annotation |
| | name | xsd:string | optional | | | |
| | nameID | GUID | required | | | |
| | pattern | xsd:anyURI | | | | |
| | isGuaranteedDeliveryRequired | xsd:boolean | | false | | |
| | | | | | | |

source

```
<xsd:element name="QueryResponse" substitutionGroup="BusinessTransaction">
  <xsd:complexType>
    <xsd:complexContent>
      <xsd:extension base="BusinessTransactionType">
        <xsd:sequence>
          <xsd:element name="RequestingBusinessActivity">
            <xsd:complexType>
              <xsd:complexContent>
                <xsd:restriction base="BusinessAction">
                  <xsd:sequence>
                    <xsd:element ref="Documentation" minOccurs="0" maxOccurs="unbounded"/>
                  </xsd:sequence>
                </xsd:restriction>
              </xsd:complexContent>
            </xsd:complexType>
          </xsd:element>
          <xsd:element name="RespondingBusinessActivity">
            <xsd:complexType>
              <xsd:complexContent>
                <xsd:restriction base="BusinessAction">
                  <xsd:sequence>
                    <xsd:element ref="Documentation" minOccurs="0" maxOccurs="unbounded"/>
                  </xsd:sequence>
                </xsd:restriction>
              </xsd:complexContent>
            </xsd:complexType>
          </xsd:element>
          <xsd:element name="TransactionInformation">
            <xsd:complexType>
              <xsd:complexContent>
                <xsd:restriction base="BusinessAction">
                  <xsd:sequence>
                    <xsd:element ref="Documentation" minOccurs="0" maxOccurs="unbounded"/>
                  </xsd:sequence>
                </xsd:restriction>
              </xsd:complexContent>
            </xsd:complexType>
          </xsd:element>
        </xsd:sequence>
      </xsd:extension>
    </xsd:complexContent>
  </xsd:complexType>
</xsd:element>
```

```

        </xsd:complexContent>
      </xsd:complexType>
    </xsd:element>
    <xsd:element ref="TransactionInformation" minOccurs="0" maxOccurs="unbounded"/>
  </xsd:sequence>
</xsd:extension>
</xsd:complexContent>
</xsd:complexType>
</xsd:element>

```

2144
2145
2146
2147

6.1.27.1 element QueryResponse/RequestingBusinessActivity



namespace <http://www.untmg.org/downloads/General/approved/BPSS-v1pt10.xsd>

type restriction of [BusinessAction](#)

children [Documentation](#)

| attributes | Name | Type | Use | Default | Fixed | Annotation |
|------------|---------------------------------|--------------|----------|---------|-------|-----------------------------|
| | name | xsd:string | optional | | | |
| | nameID | GUID | required | | | |
| | isAuthorizationRequired | xsd:boolean | | false | | docume deprec ntation ed |
| | isIntelligibleCheckRequired | xsd:boolean | | false | | |
| | isNonRepudiationRequired | xsd:boolean | | false | | |
| | isNonRepudiationReceiptRequired | xsd:boolean | | false | | |
| | timeToAcknowledgeReceipt | xsd:duration | | | | |
| | timeToAcknowledgeProcessing | xsd:duration | | | | |

source

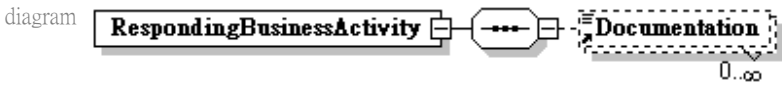
```

<xsd:element name="RequestingBusinessActivity">
  <xsd:complexType>
    <xsd:complexContent>
      <xsd:restriction base="BusinessAction">
        <xsd:sequence>
          <xsd:element ref="Documentation" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
      </xsd:restriction>
    </xsd:complexContent>
  </xsd:complexType>
</xsd:element>

```

2148
2149
2150
2151

6.1.27.2 element QueryResponse/RespondingBusinessActivity



namespace <http://www.untmg.org/downloads/General/approved/BPSS-v1pt10.xsd>

type restriction of [BusinessAction](#)

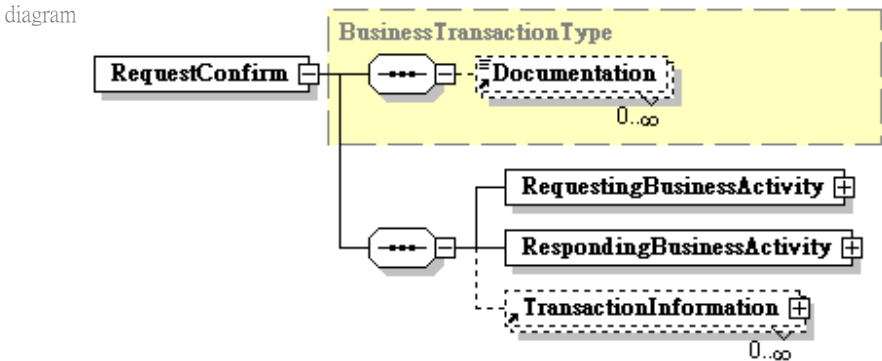
children [Documentation](#)

| attributes | Name | Type | Use | Default | Fixed | Annotation |
|------------|---------------------------------|--------------|----------|---------|-------|---------------------------------|
| | name | xsd:string | optional | | | |
| | nameID | GUID | required | | | |
| | isAuthorizationRequired | xsd:boolean | | false | | docume deprecated ntation ed |
| | isIntelligibleCheckRequired | xsd:boolean | | false | | |
| | isNonRepudiationRequired | xsd:boolean | | false | | |
| | isNonRepudiationReceiptRequired | xsd:boolean | | false | | |
| | timeToAcknowledgeReceipt | xsd:duration | | | | |
| | timeToAcknowledgeProcessing | xsd:duration | | | | |

source

```
<xsd:element name="RespondingBusinessActivity">
  <xsd:complexType>
    <xsd:complexContent>
      <xsd:restriction base="BusinessAction">
        <xsd:sequence>
          <xsd:element ref="Documentation" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
      </xsd:restriction>
    </xsd:complexContent>
  </xsd:complexType>
</xsd:element>
```

2152
2153
2154 **6.1.29 element RequestConfirm**
2155



namespace <http://www.untmg.org/downloads/General/approved/BPSS-v1pt10.xsd>

type extension of [BusinessTransactionType](#)

children [Documentation](#) [RequestingBusinessActivity](#) [RespondingBusinessActivity](#) [TransactionInformation](#)

| attributes | Name | Type | Use | Default | Fixed | Annotation |
|------------|------------------------------|-------------|----------|---------|-------|------------|
| | name | xsd:string | optional | | | |
| | nameID | GUID | required | | | |
| | pattern | xsd:anyURI | | | | |
| | isGuaranteedDeliveryRequired | xsd:boolean | | false | | |

```
source <xsd:element name="RequestConfirm" substitutionGroup="BusinessTransaction">
  <xsd:complexType>
    <xsd:complexContent>
      <xsd:extension base="BusinessTransactionType">
        <xsd:sequence>
          <xsd:element name="RequestingBusinessActivity">
            <xsd:complexType>
              <xsd:complexContent>
                <xsd:restriction base="BusinessAction">
                  <xsd:sequence>
                    <xsd:element ref="Documentation" minOccurs="0" maxOccurs="unbounded"/>
                  </xsd:sequence>
                  <xsd:attribute name="isAuthorizationRequired" default="true"/>
                  <xsd:attribute name="isAuthorizationRequired" default="true"/>
                </xsd:restriction>
              </xsd:complexContent>
            </xsd:complexType>
          </xsd:element>
          <xsd:element name="RespondingBusinessActivity">
            <xsd:complexType>
              <xsd:complexContent>
                <xsd:restriction base="BusinessAction">
                  <xsd:sequence>
                    <xsd:element ref="Documentation" minOccurs="0" maxOccurs="unbounded"/>
                  </xsd:sequence>
                  <xsd:attribute name="isAuthorizationRequired" default="true"/>
                  <xsd:attribute name="timeToAcknowledgeReceipt" default="P2H"/>
                </xsd:restriction>
              </xsd:complexContent>
            </xsd:complexType>
          </xsd:element>
          <xsd:element ref="TransactionInformation" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
      </xsd:extension>
    </xsd:complexContent>
  </xsd:complexType>
</xsd:element>
```

2156
2157
2158
2159

6.1.29.1 element RequestConfirm/RequestingBusinessActivity



namespace http://www.untmg.org/downloads/General/approved/BPSS-v1pt10.xsd

type restriction of [BusinessAction](#)

children [Documentation](#)

| attributes | Name | Type | Use | Default | Fixed | Annotation |
|------------|---------------------------------|--------------|----------|---------|-------|------------|
| | name | xsd:string | optional | | | |
| | nameID | GUID | required | | | |
| | isAuthorizationRequired | xsd:boolean | | true | | |
| | isIntelligibleCheckRequired | xsd:boolean | | false | | |
| | isNonRepudiationRequired | xsd:boolean | | false | | |
| | isNonRepudiationReceiptRequired | xsd:boolean | | false | | |
| | timeToAcknowledgeReceipt | xsd:duration | | | | |
| | timeToAcknowledgeProcessing | xsd:duration | | | | |

```
source <xsd:element name="RequestingBusinessActivity">
  <xsd:complexType>
    <xsd:complexContent>
      <xsd:restriction base="BusinessAction">
        <xsd:sequence>
          <xsd:element ref="Documentation" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:attribute name="isAuthorizationRequired" default="true"/>
        <xsd:attribute name="isAuthorizationRequired" default="true"/>
      </xsd:restriction>
    </xsd:complexContent>
  </xsd:complexType>
</xsd:element>
```

2160
2161
2162 **6.1.29.2 element RequestConfirm/RespondingBusinessActivity**
2163



namespace http://www.untmg.org/downloads/General/approved/BPSS-v1pt10.xsd

type restriction of [BusinessAction](#)

children [Documentation](#)

| attributes | Name | Type | Use | Default | Fixed | Annotation |
|------------|-----------------------------|-------------|----------|---------|-------|------------|
| | name | xsd:string | optional | | | |
| | nameID | GUID | required | | | |
| | isAuthorizationRequired | xsd:boolean | | true | | |
| | isIntelligibleCheckRequired | xsd:boolean | | false | | |
| | isNonRepudiation | xsd:boolean | | false | | |

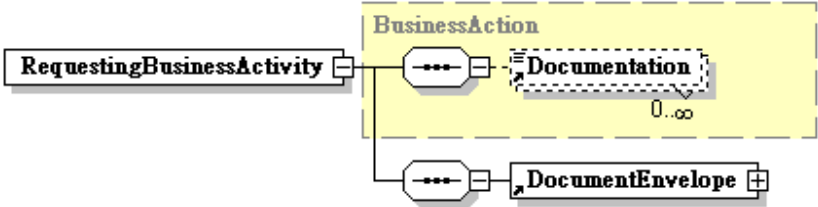
| | | |
|-----------------------------|--------------|-------|
| Required | | |
| isNonRepudiation | xsd:boolean | false |
| ReceiptRequired | | |
| timeToAcknowledgeReceipt | xsd:duration | P2H |
| timeToAcknowledgeProcessing | xsd:duration | |

```

source <xsd:element name="RespondingBusinessActivity">
  <xsd:complexType>
    <xsd:complexContent>
      <xsd:restriction base="BusinessAction">
        <xsd:sequence>
          <xsd:element ref="Documentation" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:attribute name="isAuthorizationRequired" default="true"/>
        <xsd:attribute name="timeToAcknowledgeReceipt" default="P2H"/>
      </xsd:restriction>
    </xsd:complexContent>
  </xsd:complexType>
</xsd:element>

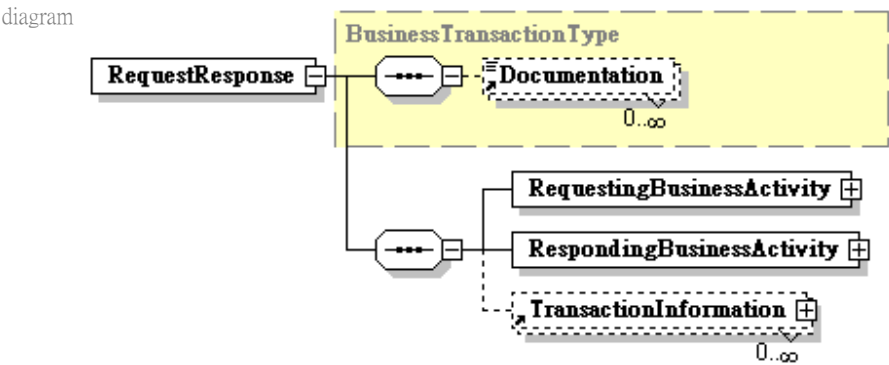
```

2164 **6.1.30 element RequestingBusinessActivity**
2165

| | | | | | |
|------------|---|--------------|----------|---------|--------------------------|
| diagram |  | | | | |
| namespace | http://www.untmg.org/downloads/General/approved/BPSS-v1pt11.xsd | | | | |
| type | extension of BusinessAction | | | | |
| children | Documentation DocumentEnvelope | | | | |
| used by | element BusinessTransaction | | | | |
| attributes | Name | Type | Use | Default | Annotation |
| | name | xsd:string | optional | | |
| | nameID | GUID | required | | |
| | isAuthorizationRequired | xsd:boolean | | false | documentation deprecated |
| | isIntelligibleCheckRequired | xsd:boolean | | false | |
| | isNonRepudiationRequired | xsd:boolean | | false | |
| | isNonRepudiationReceiptRequired | xsd:boolean | | false | |
| | timeToAcknowledgeReceipt | xsd:duration | | | |
| | timeToAcknowledgeProcessing | xsd:duration | | | |
| | retryCount | xsd:int | | | |
| source | <pre><xsd:element name="RequestingBusinessActivity"> <xsd:complexType> <xsd:complexContent> <xsd:extension base="BusinessAction"> <xsd:sequence> <xsd:element ref="DocumentEnvelope"/> </xsd:sequence> <xsd:attribute name="retryCount" type="xsd:int"/> </xsd:extension> </xsd:complexContent> </xsd:complexType> </xsd:element></pre> | | | | |

2166

2167 **6.1.31 element RequestResponse**
2168



namespace <http://www.untmg.org/downloads/General/approved/BPSS-v1pt10.xsd>

type extension of [BusinessTransactionType](#)

children [Documentation](#) [RequestingBusinessActivity](#) [RespondingBusinessActivity](#) [TransactionInformation](#)

| attributes | Name | Type | Use | Default | Fixed | Annotation |
|------------|------------------------------|-------------|----------|---------|-------|------------|
| | name | xsd:string | optional | | | |
| | nameID | GUID | required | | | |
| | pattern | xsd:anyURI | | | | |
| | isGuaranteedDeliveryRequired | xsd:boolean | | false | | |

source

```
<xsd:element name="RequestResponse" substitutionGroup="BusinessTransaction">
  <xsd:complexType>
    <xsd:complexContent>
      <xsd:extension base="BusinessTransactionType">
        <xsd:sequence>
          <xsd:element name="RequestingBusinessActivity">
            <xsd:complexType>
              <xsd:complexContent>
                <xsd:restriction base="BusinessAction">
                  <xsd:sequence>
                    <xsd:element ref="Documentation" minOccurs="0" maxOccurs="unbounded"/>
                  </xsd:sequence>
                </xsd:restriction>
              </xsd:complexContent>
            </xsd:complexType>
          </xsd:element>
          <xsd:element name="RespondingBusinessActivity">
            <xsd:complexType>
              <xsd:complexContent>
                <xsd:restriction base="BusinessAction">
                  <xsd:sequence>
                    <xsd:element ref="Documentation" minOccurs="0" maxOccurs="unbounded"/>
                  </xsd:sequence>
                </xsd:restriction>
              </xsd:complexContent>
            </xsd:complexType>
          </xsd:element>
          <xsd:element ref="TransactionInformation" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
      </xsd:extension>
    </xsd:complexContent>
  </xsd:complexType>
</xsd:element>
```

```
</xsd:sequence>
</xsd:extension>
</xsd:complexContent>
</xsd:complexType>
</xsd:element>
```

2169
2170
2171
2172

6.1.31.1 element RequestResponse/RequestingBusinessActivity



namespace <http://www.untmg.org/downloads/General/approved/BPSS-v1pt10.xsd>

type restriction of [BusinessAction](#)

children [Documentation](#)

| attributes | Name | Type | Use | Default | Fixed | Annotation |
|------------|---------------------------------|--------------|----------|---------|-------|-----------------------------|
| | name | xsd:string | optional | | | |
| | nameID | GUID | required | | | |
| | isAuthorizationRequired | xsd:boolean | | false | | docume deprec ntation ed |
| | isIntelligibleCheckRequired | xsd:boolean | | false | | |
| | isNonRepudiationRequired | xsd:boolean | | false | | |
| | isNonRepudiationReceiptRequired | xsd:boolean | | false | | |
| | timeToAcknowledgeReceipt | xsd:duration | | | | |
| | timeToAcknowledgeProcessing | xsd:duration | | | | |

```
source <xsd:element name="RequestingBusinessActivity">
  <xsd:complexType>
    <xsd:complexContent>
      <xsd:restriction base="BusinessAction">
        <xsd:sequence>
          <xsd:element ref="Documentation" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
      </xsd:restriction>
    </xsd:complexContent>
  </xsd:complexType>
</xsd:element>
```

2173
2174
2175
2176

6.1.31.2 element RequestResponse/RespondingBusinessActivity




namespace <http://www.untmg.org/downloads/General/approved/BPSS-v1pt10.xsd>

| | | | | | | |
|--|--|--------------|----------|---------|-------|---------------------|
| type restriction of BusinessAction | | | | | | |
| children Documentation | | | | | | |
| attributes | Name | Type | Use | Default | Fixed | Annotation |
| | name | xsd:string | optional | | | |
| | nameID | GUID | required | | | |
| | isAuthorizationRequired | xsd:boolean | | false | | document deprecated |
| | isIntelligibleCheckRequired | xsd:boolean | | false | | |
| | isNonRepudiationRequired | xsd:boolean | | false | | |
| | isNonRepudiationReceiptRequired | xsd:boolean | | false | | |
| | timeToAcknowledgeReceipt | xsd:duration | | | | |
| | timeToAcknowledgeProcessing | xsd:duration | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| source | <pre><xsd:element name="RespondingBusinessActivity"> <xsd:complexType> <xsd:complexContent> <xsd:restriction base="BusinessAction"> <xsd:sequence> <xsd:element ref="Documentation" minOccurs="0" maxOccurs="unbounded"/> </xsd:sequence> </xsd:restriction> </xsd:complexContent> </xsd:complexType> </xsd:element></pre> | | | | | |

6.1.32 element RespondingBusinessActivity

| | | | | | |
|------------|---|--------------|----------|---------|--------------------------|
| diagram | | | | | |
| namespace | http://www.untmg.org/downloads/General/approved/BPSS-v1pt11.xsd | | | | |
| type | extension of BusinessAction | | | | |
| children | Documentation DocumentEnvelope | | | | |
| used by | element BusinessTransaction | | | | |
| attributes | Name | Type | Use | Default | Annotation |
| | name | xsd:string | optional | | |
| | nameID | GUID | required | | |
| | isAuthorizationRequired | xsd:boolean | | false | documentation deprecated |
| | isIntelligibleCheckRequired | xsd:boolean | | false | |
| | isNonRepudiationRequired | xsd:boolean | | false | |
| | isNonRepudiationReceiptRequired | xsd:boolean | | false | |
| | timeToAcknowledgeReceipt | xsd:duration | | | |
| | timeToAcknowledgeProcessing | xsd:duration | | | |
| source | <pre><xsd:element name="RespondingBusinessActivity"> <xsd:complexType> <xsd:complexContent> <xsd:extension base="BusinessAction"> <xsd:sequence> <xsd:element ref="DocumentEnvelope" minOccurs="0" maxOccurs="unbounded"/> </xsd:sequence> </xsd:extension> </xsd:complexContent> </xsd:complexType> </xsd:element></pre> | | | | |

2181 **6.1.33 element Start**
2182

| | | | | | |
|----------------------|--|------------|----------|----------|------------|
| diagram |  | | | | |
| namespace | http://www.untmg.org/downloads/General/approved/BPSS-v1pt11.xsd | | | | |
| children | Documentation | | | | |
| used by | element BinaryCollaboration | | | | |
| attributes | Name | Type | Use | Default | Annotation |
| | toBusinessState | xsd:string | required | | |
| | toBusinessStateIDREF | GUIDREF | | | |
| | name | xsd:string | optional | | |
| | nameID | GUID | required | | |
| identity constraints | unique | Name | Refer | Selector | Field(s) |
| | | Start-ID | | . | nameID |
| source | <pre><xsd:element name="Start"> <xsd:complexType> <xsd:sequence> <xsd:element ref="Documentation" minOccurs="0" maxOccurs="unbounded"/> </xsd:sequence> <xsd:attribute name="toBusinessState" type="xsd:string" use="required"/> <xsd:attribute name="toBusinessStateIDREF" type="GUIDREF"/> <xsd:attributeGroup ref="name"/> </xsd:complexType> <xsd:unique name="Start-ID"> <xsd:selector xpath="."/> <xsd:field xpath="nameID"/> </xsd:unique> </xsd:element></pre> | | | | |

2183

2184 **6.1.34 element SubstitutionSet**
2185

| | | | | | |
|------------|---|------------|----------|---------|------------|
| diagram | | | | | |
| namespace | http://www.untmg.org/downloads/General/approved/BPSS-v1pt11.xsd | | | | |
| children | Documentation DocumentSubstitution AttributeSubstitution | | | | |
| used by | elements Package ProcessSpecification | | | | |
| attributes | Name | Type | Use | Default | Annotation |
| | name | xsd:string | optional | | |
| | nameID | GUID | required | | |
| | applyToScope | xsd:string | required | | |
| source | <pre><xsd:element name="SubstitutionSet"> <xsd:complexType> <xsd:sequence> <xsd:element ref="Documentation" minOccurs="0" maxOccurs="unbounded"/> <xsd:element ref="DocumentSubstitution" minOccurs="0" maxOccurs="unbounded"/> <xsd:element ref="AttributeSubstitution" minOccurs="0" maxOccurs="unbounded"/> </xsd:sequence> <xsd:attributeGroup ref="name"/> <xsd:attribute name="applyToScope" type="xsd:string" use="required"/> </xsd:complexType> </xsd:element></pre> | | | | |

2186

2187

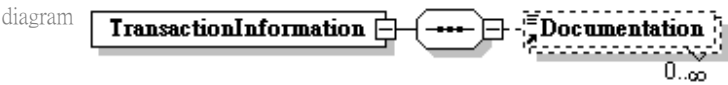
2188

6.1.35 element Success

| | | | | | |
|----------------------|--|--------------------|----------|---------------|--------------------|
| diagram | | | | | |
| namespace | http://www.untmg.org/downloads/General/approved/BPSS-v1pt11.xsd | | | | |
| children | Documentation ConditionExpression | | | | |
| used by | element BinaryCollaboration | | | | |
| attributes | Name | Type | Use | Default | Annotation |
| | name | xsd:string | optional | | |
| | nameID | GUID | required | | |
| | fromBusinessState | xsd:string | required | | |
| | fromBusinessStateIDREF | GUIDREF | | | |
| | conditionGuard | xsd:NMTOKEN | | | |
| identity constraints | unique | Name Success-ID | Refer | Selector . | Field(s) nameID |
| source | <pre><xsd:element name="Success"> <xsd:complexType> <xsd:sequence> <xsd:element ref="Documentation" minOccurs="0" maxOccurs="unbounded"/> <xsd:element ref="ConditionExpression" minOccurs="0"/> </xsd:sequence> <xsd:attributeGroup ref="name"/> <xsd:attribute name="fromBusinessState" type="xsd:string" use="required"/> <xsd:attribute name="fromBusinessStateIDREF" type="GUIDREF"/> <xsd:attribute name="conditionGuard"/> <xsd:simpleType> <xsd:restriction base="xsd:NMTOKEN"> <xsd:enumeration value="ProtocolSuccess"/> <xsd:enumeration value="AnyProtocolFailure"/> <xsd:enumeration value="RequestReceiptFailure"/> <xsd:enumeration value="RequestProcessingFailure"/> <xsd:enumeration value="ResponseReceiptFailure"/> <xsd:enumeration value="ResponseProcessingFailure"/> <xsd:enumeration value="SignalTimeout"/> <xsd:enumeration value="ResponseTimeout"/> <xsd:enumeration value="BusinessSuccess"/> <xsd:enumeration value="BusinessFailure"/> <xsd:enumeration value="Success"/> <xsd:enumeration value="Failure"/> </xsd:restriction> </xsd:simpleType> </xsd:complexType> <xsd:unique name="Success-ID"> <xsd:selector xpath="."/> <xsd:field xpath="nameID"/> </xsd:unique> </xsd:element></pre> | | | | |

2189

2190 **6.1.36 element TransactionInformation**
2191



namespace <http://www.untmg.org/downloads/General/approved/BPSS-v1pt10.xsd>

type extension of [PassInformation](#)

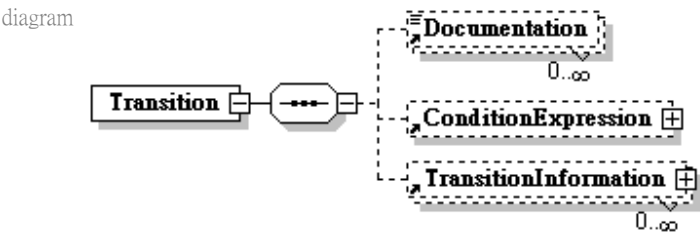
children [Documentation](#)

used by elements [BusinessDocument](#) [BusinessTransaction](#) [CommercialTransaction](#) [InformationDistribution](#) [Notification](#)
[QueryResponse](#) [RequestConfirm](#) [RequestResponse](#)

| attributes | Name | Type | Use | Default | Fixed | Annotation |
|------------|-------------------|-------------|----------|---------|-------|------------|
| | name | xsd:string | optional | | | |
| | nameID | GUID | required | | | |
| | memoryValue | xsd:string | required | | | |
| | isInternalPassing | xsd:boolean | | | | |
| | fromBusinessDocu | xsd:string | required | | | |
| | ment | | | | | |
| | fromBusinessDocu | GUIDREF | | | | |
| | ementIDREF | | | | | |

```
source <xsd:element name="TransactionInformation">
  <xsd:complexType>
    <xsd:complexContent>
      <xsd:extension base="PassInformation">
        <xsd:sequence>
          <xsd:element ref="Documentation" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:attribute name="isInternalPassing" type="xsd:boolean"/>
        <xsd:attribute name="fromBusinessDocument" type="xsd:string" use="required"/>
        <xsd:attribute name="fromBusinessDocumentIDREF" type="GUIDREF"/>
      </xsd:extension>
    </xsd:complexContent>
  </xsd:complexType>
</xsd:element>
```

2192
2193
2194 **6.1.37 element Transition**
2195



namespace <http://www.untmg.org/downloads/General/approved/BPSS-v1pt11.xsd>

children [Documentation](#) [ConditionExpression](#) [TransitionInformation](#)

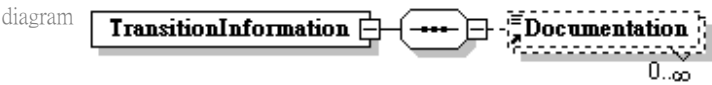
used by elements [BinaryCollaboration](#) [BusinessPartnerRole](#)

| | | | | | | |
|-------------|--|---------------|----------|---------|----------|------------|
| attributes | Name | Type | Use | Default | Fixed | Annotation |
| | name | xsd:string | optional | | | |
| | nameID | GUID | required | | | |
| | onInitiation | xsd:boolean | | false | | |
| | fromBusinessState | xsd:string | required | | | |
| | fromBusinessStateIDREF | GUIDREF | optional | | | |
| | toBusinessState | xsd:string | required | | | |
| | toBusinessStateIDREF | GUIDREF | optional | | | |
| | conditionGuard | xsd:NMTOKEN | | | | |
| identity | Name | | Refer | | Selector | Field(s) |
| constraints | unique | Transition-ID | | | . | nameID |
| source | <pre> <xsd:element name="Transition"> <xsd:complexType> <xsd:sequence> <xsd:element ref="Documentation" minOccurs="0" maxOccurs="unbounded"/> <xsd:element ref="ConditionExpression" minOccurs="0"/> <xsd:element ref="TransitionInformation" minOccurs="0" maxOccurs="unbounded"/> </xsd:sequence> <xsd:attributeGroup ref="name"/> <xsd:attribute name="onInitiation" type="xsd:boolean" default="false"/> <xsd:attribute name="fromBusinessState" type="xsd:string" use="required"/> <xsd:attribute name="fromBusinessStateIDREF" type="GUIDREF" use="optional"/> <xsd:attribute name="toBusinessState" type="xsd:string" use="required"/> <xsd:attribute name="toBusinessStateIDREF" type="GUIDREF" use="optional"/> <xsd:attribute name="conditionGuard"/> <xsd:simpleType> <xsd:restriction base="xsd:NMTOKEN"> <xsd:enumeration value="ProtocolSuccess"/> <xsd:enumeration value="AnyProtocolFailure"/> <xsd:enumeration value="RequestReceiptFailure"/> <xsd:enumeration value="RequestProcessingFailure"/> <xsd:enumeration value="ResponseReceiptFailure"/> <xsd:enumeration value="ResponseProcessingFailure"/> <xsd:enumeration value="SignalTimeout"/> <xsd:enumeration value="ResponseTimeout"/> <xsd:enumeration value="BusinessSuccess"/> <xsd:enumeration value="BusinessFailure"/> <xsd:enumeration value="Success"/> <xsd:enumeration value="Failure"/> </xsd:restriction> </xsd:simpleType> </xsd:complexType> <xsd:unique name="Transition-ID"> <xsd:selector xpath="."/> <xsd:field xpath="nameID"/> </xsd:unique> </xsd:element> </pre> | | | | | |

2197

2198

6.1.38 element TransitionInformation



namespace <http://www.untmg.org/downloads/General/approved/BPSS-v1pt11.xsd>

type extension of [PassInformation](#)

children [Documentation](#)

used by elements [BusinessDocument](#) [Transition](#)

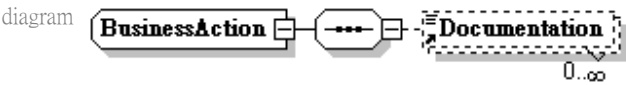
| attributes | Name | Type | Use | Default | Fixed | Annotation |
|------------|------------------------|------------|----------|---------|-------|------------|
| | name | xsd:string | optional | | | |
| | nameID | GUID | required | | | |
| | memoryValue | xsd:string | required | | | |
| | fromBusinessState | xsd:string | required | | | |
| | fromBusinessStateIDREF | GUIDREF | | | | |
| | toBusinessState | xsd:string | required | | | |
| | toBusinessStateIDREF | GUIDREF | | | | |

```
source <xsd:element name="TransitionInformation">
  <xsd:complexType>
    <xsd:complexContent>
      <xsd:extension base="PassInformation">
        <xsd:sequence>
          <xsd:element ref="Documentation" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:attribute name="fromBusinessState" type="xsd:string" use="required"/>
        <xsd:attribute name="fromBusinessStateIDREF" type="GUIDREF"/>
        <xsd:attribute name="toBusinessState" type="xsd:string" use="required"/>
        <xsd:attribute name="toBusinessStateIDREF" type="GUIDREF"/>
      </xsd:extension>
    </xsd:complexContent>
  </xsd:complexType>
</xsd:element>
```

2199

2200 6.1.39 complexType BusinessAction

2201



namespace <http://www.untmg.org/downloads/General/approved/BPSS-v1pt11.xsd>

children [Documentation](#)

used by elements [RequestingBusinessActivity](#) [CommercialTransaction/RequestingBusinessActivity](#)
[InformationDistribution/RequestingBusinessActivity](#)
[Notification/RequestingBusinessActivity](#)
[QueryResponse/RequestingBusinessActivity](#)
[RequestConfirm/RequestingBusinessActivity](#)
[RequestResponse/RequestingBusinessActivity](#) [RespondingBusinessActivity](#)
[CommercialTransaction/RespondingBusinessActivity](#)
[QueryResponse/RespondingBusinessActivity](#)
[RequestConfirm/RespondingBusinessActivity](#)
[RequestResponse/RespondingBusinessActivity](#)

| attributes | Name | Type | Use | Default | Fixed | Annotation |
|------------|---------------------------------|--------------|----------|---------|-------|---------------------|
| | name | xsd:string | optional | | | |
| | nameID | GUID | required | | | |
| | isAuthorizationRequired | xsd:boolean | | false | | document deprecated |
| | isIntelligibleCheckRequired | xsd:boolean | | false | | |
| | isNonRepudiationRequired | xsd:boolean | | false | | |
| | isNonRepudiationReceiptRequired | xsd:boolean | | false | | |
| | timeToAcknowledgeReceipt | xsd:duration | | | | |
| | timeToAcknowledgeProcessing | xsd:duration | | | | |

source

```
<xsd:complexType name="BusinessAction">
  <xsd:sequence>
    <xsd:element ref="Documentation" minOccurs="0" maxOccurs="unbounded"/>
  </xsd:sequence>
  <xsd:attributeGroup ref="name"/>
  <xsd:attribute name="isAuthorizationRequired" type="xsd:boolean" default="false">
    <xsd:annotation>
      <xsd:documentation>deprecated</xsd:documentation>
    </xsd:annotation>
  </xsd:attribute>
  <xsd:attribute name="isIntelligibleCheckRequired" type="xsd:boolean" default="false"/>
  <xsd:attribute name="isNonRepudiationRequired" type="xsd:boolean" default="false"/>
  <xsd:attribute name="isNonRepudiationReceiptRequired" type="xsd:boolean" default="false"/>
  <xsd:attribute name="timeToAcknowledgeReceipt" type="xsd:duration"/>
  <xsd:attribute name="timeToAcknowledgeProcessing" type="xsd:duration"/>
</xsd:complexType>
```

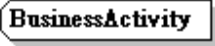
2202

2203

2204

2205 **6.1.40 complexType BusinessActivity**

2206

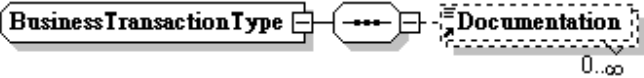
| | | | | | |
|------------|---|------------|----------|---------|------------|
| diagram |  | | | | |
| namespace | http://www.untmg.org/downloads/General/approved/BPSS-v1pt11.xsd | | | | |
| used by | elements BusinessTransactionActivity CollaborationActivity | | | | |
| attributes | Name | Type | Use | Default | Annotation |
| | name | xsd:string | optional | | |
| | nameID | GUID | required | | |
| | fromRole | xsd:string | required | | |
| | fromRoleIDREF | GUIDREF | | | |
| | toRole | xsd:string | required | | |
| | toRoleIDREF | GUIDREF | | | |
| | beginsWhen | xsd:string | | | |
| | endsWhen | xsd:string | | | |
| | preCondition | xsd:string | | | |
| | postCondition | xsd:string | | | |
| source | <pre><xsd:complexType name="BusinessActivity"> <xsd:attributeGroup ref="name"/> <xsd:attribute name="fromRole" type="xsd:string" use="required"/> <xsd:attribute name="fromRoleIDREF" type="GUIDREF"/> <xsd:attribute name="toRole" type="xsd:string" use="required"/> <xsd:attribute name="toRoleIDREF" type="GUIDREF"/> <xsd:attribute name="beginsWhen" type="xsd:string"/> <xsd:attribute name="endsWhen" type="xsd:string"/> <xsd:attribute name="preCondition" type="xsd:string"/> <xsd:attribute name="postCondition" type="xsd:string"/> </xsd:complexType></pre> | | | | |

2207

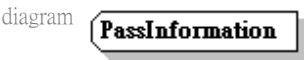
2208

2209 **6.1.41 complexType BusinessTransactionType**

2210

| | | | | | |
|------------|---|-------------|----------|---------|-------|
| diagram |  | | | | |
| namespace | http://www.untmg.org/downloads/General/approved/BPSS-v1pt11.xsd | | | | |
| children | Documentation | | | | |
| used by | elements BusinessTransaction CommercialTransaction InformationDistribution Notification QueryResponse RequestConfirm RequestResponse | | | | |
| attributes | Name | Type | Use | Default | Fixed |
| | name | xsd:string | optional | | |
| | nameID | GUID | required | | |
| | pattern | xsd:anyURI | | | |
| | isGuaranteedDeliv | xsd:boolean | | false | |
| | eryRequired | | | | |
| source | <pre><xsd:complexType name="BusinessTransactionType"> <xsd:sequence> <xsd:element ref="Documentation" minOccurs="0" maxOccurs="unbounded"/> </xsd:sequence> <xsd:attributeGroup ref="name"/> <xsd:attribute name="pattern" type="xsd:anyURI"/> <xsd:attribute name="isGuaranteedDeliveryRequired" type="xsd:boolean" default="false"/> </xsd:complexType></pre> | | | | |

2211 **6.1.42 complexType PassInformation**
2212



namespace <http://www.untmg.org/downloads/General/approved/BPSS-v1pt11.xsd>


used by elements [TransactionInformation](#) [TransitionInformation](#)

| attributes | Name | Type | Use | Default | Fixed | Annotation |
|------------|--------------------|------------|----------|---------|-------|------------|
| | name | xsd:string | optional | | | |
| | nameID | GUID | required | | | |
| | expressionLanguage | xsd:string | optional | | | |
| | memoryValue | xsd:string | required | | | |

source

```
<xsd:complexType name="PassInformation">
  <xsd:attributeGroup ref="name"/>
  <xsd:attribute name="expressionLanguage" type="xsd:string" use="optional"/>
  <xsd:attribute name="memoryValue" type="xsd:string" use="required"/>
</xsd:complexType>
```

2213
2214 **6.1.43 complexType RoleType**
2215
2216

| | | | | | |
|------------|--|------------|----------|---------|------------|
| diagram |  | | | | |
| namespace | http://www.untmg.org/downloads/General/approved/BPSS-v1pt11.xsd | | | | |
| children | Documentation | | | | |
| used by | element BinaryCollaboration/Role | | | | |
| attributes | Name | Type | Use | Default | Annotation |
| | name | xsd:string | optional | | |
| | nameID | GUID | required | | |
| source | <pre><xsd:complexType name="RoleType"> <xsd:sequence> <xsd:element ref="Documentation" minOccurs="0"/> </xsd:sequence> <xsd:attributeGroup ref="name"/> </xsd:complexType></pre> | | | | |

2217

2218 6.1.44 simpleType GUID

2219

| | |
|-----------|--|
| namespace | http://www.untmg.org/downloads/General/approved/BPSS-v1pt11.xsd |
| type | xsd:string |
| used by | attribute Namespace/@nameID attributeGroup name |
| source | <code><xsd:simpleType name="GUID"> <xsd:restriction base="xsd:string"/> </xsd:simpleType></code> |

2220

2221

2222 6.1.45 simpleType GUIDREF

2223

namespace http://www.untmg.org/downloads/General/approved/BPSS-v1pt11.xsd

type **xsd:string**

used by attributes [CollaborationActivity/@binaryCollaborationIDREF](#) [Attachment/@businessDocumentIDREF](#)
[DocumentEnvelope/@businessDocumentIDREF](#) [BusinessRules/@businessDocumentIDREF](#)
[BusinessTransactionActivity/@businessTransactionIDREF](#)
[BusinessRules/@documentEnvelopeIDREF](#)
[TransactionInformation/@fromBusinessDocuementIDREF](#)
[TransitionInformation/@fromBusinessStateIDREF](#) [Transition/@fromBusinessStateIDREF](#)
[Success/@fromBusinessStateIDREF](#) [Failure/@fromBusinessStateIDREF](#)
[BusinessActivity/@fromRoleIDREF](#) [BinaryCollaboration/@initiatingRoleIDREF](#)
[DocumentSubstitution/@originalBusinessDocumentID](#) [Performs/@roleIDREF](#)
[Start/@toBusinessStateIDREF](#) [Transition/@toBusinessStateIDREF](#)
[TransitionInformation/@toBusinessStateIDREF](#) [BusinessActivity/@toRoleIDREF](#)

source `<xsd:simpleType name="GUIDREF">
 <xsd:restriction base="xsd:string"/>
</xsd:simpleType>`

2224

2225

2226

6.1.46 attributeGroup documentSecurity

| | | | | | |
|------------|---|--|-----|---------|------------|
| namespace | http://www.untmg.org/downloads/General/approved/BPSS-v1pt11.xsd | | | | |
| used by | elements | Attachment DocumentEnvelope | | | |
| attributes | Name | Type | Use | Default | Annotation |
| | isAuthenticated | xsd:NMTOKEN | | | |
| | isConfidential | xsd:NMTOKEN | | | |
| | isTamperDetectable | xsd:NMTOKEN | | | |
| source | <pre><xsd:attributeGroup name="documentSecurity"> <xsd:attribute name="isAuthenticated"> <xsd:simpleType> <xsd:restriction base="xsd:NMTOKEN"> <xsd:enumeration value="none"/> <xsd:enumeration value="transient"/> <xsd:enumeration value="persistent"/> <xsd:enumeration value="transient-and-persistent"/> </xsd:restriction> </xsd:simpleType> </xsd:attribute> <xsd:attribute name="isConfidential"> <xsd:simpleType> <xsd:restriction base="xsd:NMTOKEN"> <xsd:enumeration value="none"/> <xsd:enumeration value="transient"/> <xsd:enumeration value="persistent"/> <xsd:enumeration value="transient-and-persistent"/> </xsd:restriction> </xsd:simpleType> </xsd:attribute> <xsd:attribute name="isTamperDetectable"> <xsd:simpleType> <xsd:restriction base="xsd:NMTOKEN"> <xsd:enumeration value="none"/> <xsd:enumeration value="transient"/> <xsd:enumeration value="persistent"/> <xsd:enumeration value="transient-and-persistent"/> </xsd:restriction> </xsd:simpleType> </xsd:attribute> </xsd:attributeGroup></pre> | | | | |

2227

2228 **6.1.47 attributeGroup name**
2229

namespace <http://www.untmg.org/downloads/General/approved/BPSS-v1pt11.xsd>

used by elements [Attachment](#) [BinaryCollaboration](#) [BusinessDocument](#) [BusinessPartnerRole](#) [BusinessRules](#) [Decision](#)
[DocumentEnvelope](#) [Failure](#) [Fork](#) [Include](#) [Join](#) [MultiPartyCollaboration](#) [Package](#) [Performs](#)
[ProcessSpecification](#) [Start](#) [SubstitutionSet](#) [Success](#) [Transition](#)

complexTypes [BusinessAction](#) [BusinessActivity](#) [BusinessTransactionType](#) [PassInformation](#) [RoleType](#)

| | | | | | | |
|------------|--------|------------|----------|---------|-------|------------|
| attributes | Name | Type | Use | Default | Fixed | Annotation |
| | name | xsd:string | optional | | | |
| | nameID | GUID | required | | | |

source `<xsd:attributeGroup name="name">
<xsd:attribute name="name" type="xsd:string" use="optional"/>
<xsd:attribute name="nameID" use="required">
<xsd:simpleType>
 <xsd:restriction base="GUID"/>
</xsd:simpleType>
</xsd:attribute>
</xsd:attributeGroup>`

2230

2231 6.2 XML to UML cross-reference

2232 The following is a table that references the XML element names in the XSD to their
2233 counterpart classes in the UML specification schema.

| XML Element | UML Class |
|-------------------------------|---|
| Attachment | Attachment |
| Role | 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 |
| Decision | Decision |
| Fork | Fork |
| Start | Start |
| Success | Success |
| Join | Join |
| Transition | Transition |
| BusinessAction | BusinessAction |
| BusinessActivity | BusinessActivity |

| | |
|------------------------|------------------------|
| DocumentSecurity | DocumentSecurity |
| BusinessRules | BusinessRules |
| PassInformation | PassInformation |
| TransactionInformation | TransactionInformation |
| TransitionInformation | TransitionInformation |

2234 The following classes in the UML specification schema are abstract, and do not have an
 2235 element equivalent in theSchema. Only their concrete subtypes are in the Schema

- 2236 ● BusinessState
- 2237 ● CompletionState
- 2238 ● BusinessActivity

2239 **6.3 Scoped Name Reference**

2240 The structure of ebXML Business Process Specification Schema encourages re-use. A
 2241 BPSS instance can include another BPSS instance by reference.

2242 In addition the contents of a BPSS instance can be arranged in a recursive package structure.
 2243 The ebXMLProcessSpecification element is a package container, so it can contain packages
 2244 within it. Package in itself is also a package container, so it can contain further packages
 2245 within it.

2246 Packages function as namespaces as per below.

2247 Finally a Package, at any level can have PackageContent. Types of Package Content are
 2248 BusinessDocument, BusinessTransaction, BinaryCollaboration, MultiPartyCollaboration.

2249 Package Content is always uniquely named within a package. Lower level elements are
 2250 uniquely named within their parent PackageContent.

2251 Each Package Content type is a built-in context provider for the Logical Model for the
 2252 Business Document definitions referenced by this ebXML ProcessSpecification.

2253 Within an ebXML BPSS instance the following applies to naming:

2254 Specification elements reference other specification elements by name through the use of
 2255 attributes. The design pattern is that elements have a name attribute and other elements that
 2256 reference the named elements do so through an attribute defined as the lowerCamelCase
 2257 version of the referenced element (e.g. Role has attribute name while Performs, which
 2258 references Role, has attribute role). Two types of attributes are provided for names and
 2259 references, XML GUID/GUIDREF based and plain text. Each named element has a
 2260 required name attribute and an optional nameID attribute. Referencing elements have

lowerCamelCase and lowerCamelCaseIDREF attributes for the referenced element. XML GUID/GUIDREF functionality requires all IDs to be globally unique and that all GUIDREFs point to a defined GUID value. Plain text attributes do not have this capability and may result in duplicate names. To unambiguously identify a referenced element using plain text attribute in the referencing attribute it is strongly recommended that XPath syntax be used. However, this is not enforced in the Schema.

The purpose of providing both solutions is to facilitate creation of BPSS instance documents directly in XML and to support future development tools that can automatically assign machine readable nameIDs and references. Both styles can be used simultaneously, in which case the GUID and GUIDREF versions provide the unambiguous referencing and the plain text versions are used to provide meaningful names. Examples of named elements and references:

```

2273 <Package name="ebXMLOrdering">
2274   <BinaryCollaboration
2275     name="OrderCollaboration"
2276     nameID="b112">
2277     <Role name="buyer" nameID="r224"/>
2278     <Role name="seller" nameID="r225"/>
2279   </BinaryCollaboration>
2280 </Package>
2281 <!--the XPath approach -->
2282 <Performs
2283   Role='//Package[@name="OAGOrdering"]/BinaryCollaboration[
2284     @name="OrderCollaboration"]/ Role[@name="buyer"]' />
2285
2286 <!--Combination approach -->
2287 <Performs Role="buyer" RoleIDREF="r224"/>

```

It is not required to use the full path specification as shown above, other forms of XPath expressions could be used as long as they resolve to a single reference. For example if buyer was unique to the document then the XPath could have been:

```

2291 <Performs Role='//Role[@name="buyer"]' />
2292 Relative paths are also allowed for example:
2293 <BusinessTransactionActivity fromRole='../ Role[@name="buyer"]' ... />

```

2294 **6.4 Sample XML document against above Schema**

2295 Provided in Appendix A

2296 7.0 Business signal structures

2297 The ebXML Message Service Specification signal structures provide business service state
 2298 alignment infrastructure, including unique message identifiers and digests used to meet the
 2299 basic process alignment requirements. The business signal payload structures provided
 2300 herein are optional and normative and are intended to provide business and legal semantic
 2301 to the business signals. Since signals do not differ in structure from business transaction to
 2302 business transaction, they are defined once and for all, and their definition is implied by the
 2303 conjunction of the Business Process Specification Schema and Message Service
 2304 Specification. Here are the Schemas for business signal payload for
 2305 ReceiptAcknowledgment and for ProcessingAcknowledgement and Exception.

2306 An Exception message would be sent in lieu of a ReceiptAcknowledgement signal or an
 2307 ProcessingAcknowledgment signal and would indicate a corresponding negative
 2308 ReceiptAcknowledgement or negative ProcessingAcknowledgement. On the other hand,
 2309 sending a ReceiptAcknowledgment or ProcessingAcknowledgement message as defined
 2310 below would indicate a positive signal.

2311 7.1 Signal Schema

```

2312 <?xml version="1.0" encoding="UTF-8"?>
2313 <!-- By Himagiri Mukkamala(himagiri@sybase.com) .
2314 This schema has the element definitions for the signal messages used in the run time execution of BPSS-->
2315 <xsd:schema targetNamespace="http://www.untmg.org/BusinessProcess/BPSS_SIGNALS"
2316 xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
2317 xmlns:bpssignal="http://www.untmg.org/BusinessProcess/BPSS_SIGNALS"
2318 xmlns:xlink="http://www.w3.org/1999/xlink" xmlns="http://www.untmg.org/BusinessProcess/BPSS_SIGNALS"
2319 elementFormDefault="qualified" attributeFormDefault="qualified" version="2.0">
2320   <xsd:import namespace="http://www.w3.org/1999/xlink" schemaLocation="http://www.oasis-
2321 open.org/committees/ebxml-msg/schema/xlink.xsd"/>
2322   <xsd:annotation>
2323     <xsd:documentation>
2324       The version of digital signature specification supported is identified using the namespace and schemalocation
2325       denoted below </xsd:documentation>
2326     </xsd:annotation>
2327     <xsd:import namespace="http://www.w3.org/2000/09/xmldsig#" schemaLocation="http://www.w3.org/TR/xmldsig-
2328 core/xmldsig-core-schema.xsd"/>
2329     <xsd:simpleType name="non-empty-string">
2330       <xsd:restriction base="xsd:string">
2331         <xsd:minLength value="1"/>
2332       </xsd:restriction>
2333     </xsd:simpleType>
2334     <xsd:complexType name="PartyInfoType">
2335       <xsd:simpleContent>
2336         <xsd:extension base="non-empty-string">
2337           <xsd:attribute name="type" type="non-empty-string"/>
2338         </xsd:extension>
2339       </xsd:simpleContent>
2340     </xsd:complexType>
2341     <xsd:complexType name="RoleType">
2342       <xsd:annotation>
2343         <xsd:documentation>
2344           This type defines the structure for Role Definition.
2345         </xsd:documentation>

```

```

2346     </xsd:annotation>
2347     <xsd:attribute name="name" type="non-empty-string" use="required"/>
2348     <xsd:attributeGroup ref="xlink.grp"/>
2349 </xsd:complexType>
2350 <xsd:attributeGroup name="xlink.grp">
2351     <xsd:attribute ref="xlink:type" fixed="simple"/>
2352     <xsd:attribute ref="xlink:href" use="required"/>
2353 </xsd:attributeGroup>
2354 <xsd:complexType name="ProcessSpecificationType">
2355     <xsd:attribute name="version" type="non-empty-string"/>
2356     <xsd:attribute name="name" type="non-empty-string"/>
2357     <xsd:attributeGroup ref="xlink.grp"/>
2358     <xsd:attribute name="nameID" type="xsd:anyURI"/>
2359 </xsd:complexType>
2360 <xsd:complexType name="SignalIdentificationInformation">
2361     <xsd:sequence>
2362         <xsd:element name="OriginalMessageIdentifier" type="bpssignal:non-empty-string"/>
2363         <xsd:element name="OriginalDocumentIdentifier" type="bpssignal:non-empty-string" minOccurs="0"/>
2364         <xsd:element name="FromPartyInfo" type="bpssignal:PartyInfoType"/>
2365         <xsd:element name="ToPartyInfo" type="bpssignal:PartyInfoType"/>
2366         <xsd:element name="FromRole" type="bpssignal:RoleType"/>
2367         <xsd:element name="ToRole" type="bpssignal:RoleType"/>
2368         <xsd:element name="OriginalMessageDateTime" type="xsd:dateTime"/>
2369         <xsd:element name="ThisMessageDateTime" type="xsd:dateTime"/>
2370         <xsd:element name="ProcessSpecificationInfo" type="bpssignal:ProcessSpecificationType"/>
2371     </xsd:sequence>
2372 </xsd:complexType>
2373 <xsd:element name="Exception">
2374     <xsd:complexType>
2375         <xsd:complexContent>
2376             <xsd:extension base="bpssignal:SignalIdentificationInformation">
2377                 <xsd:sequence>
2378                     <xsd:element name="ExceptionType">
2379                         <xsd:complexType>
2380                             <xsd:choice>
2381                                 <xsd:element name="ReceiptException">
2382                                     <xsd:simpleType>
2383                                         <xsd:restriction base="xsd:string">
2384                                             <xsd:enumeration value="Syntax"/>
2385                                             <xsd:enumeration value="Authorization"/>
2386                                             <xsd:enumeration value="Signature"/>
2387                                             <xsd:enumeration value="Sequence"/>
2388                                         </xsd:restriction>
2389                                     </xsd:simpleType>
2390                                 </xsd:element>
2391                                 <xsd:element name="ProcessingException">
2392                                     <xsd:simpleType>
2393                                         <xsd:restriction base="xsd:string">
2394                                             <xsd:enumeration value="Business"/>
2395                                             <xsd:enumeration value="Performance"/>
2396                                         </xsd:restriction>
2397                                     </xsd:simpleType>
2398                                 </xsd:element>
2399                                 <xsd:element name="GeneralException">
2400                                     <xsd:simpleType>
2401                                         <xsd:restriction base="xsd:string"/>
2402                                     </xsd:simpleType>
2403                                 </xsd:element>
2404                             </xsd:choice>
2405                         </xsd:complexType>
2406                     </xsd:element>
2407                     <xsd:element name="Reason" type="bpssignal:non-empty-string"/>

```



```

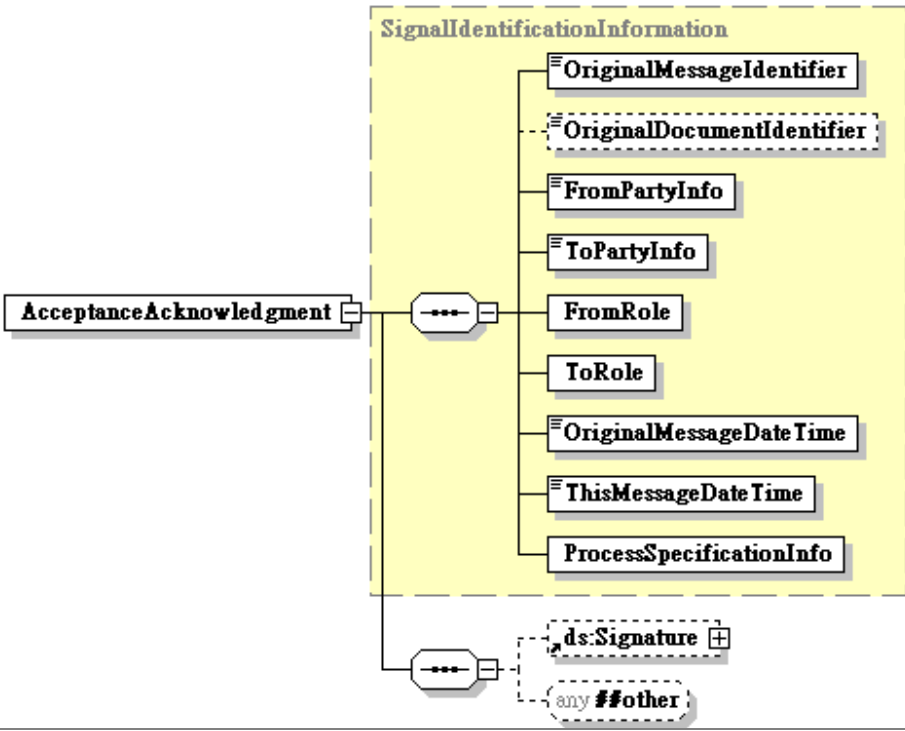
2408         <xsd:element name="ExceptionMessage" type="bpssignal:non-empty-string" minOccurs="0"/>
2409         <xsd:any namespace="##other" minOccurs="0"/>
2410     </xsd:sequence>
2411 </xsd:extension>
2412 </xsd:complexContent>
2413 </xsd:complexType>
2414 </xsd:element>
2415 <xsd:element name="ReceiptAcknowledgment">
2416     <xsd:complexType>
2417         <xsd:complexContent>
2418             <xsd:extension base="bpssignal:SignalIdentificationInformation">
2419                 <xsd:sequence>
2420                     <xsd:element ref="bpssignal:NonRepudiationInformation" minOccurs="0"/>
2421                     <xsd:element ref="ds:Signature" minOccurs="0"/>
2422                     <xsd:any namespace="##other" minOccurs="0"/>
2423                 </xsd:sequence>
2424             </xsd:extension>
2425         </xsd:complexContent>
2426     </xsd:complexType>
2427 </xsd:element>
2428 <xsd:element name="NonRepudiationInformation">
2429     <xsd:complexType>
2430         <xsd:sequence>
2431             <xsd:element ref="bpssignal:MessagePartNRInformation" maxOccurs="unbounded"/>
2432         </xsd:sequence>
2433     </xsd:complexType>
2434 </xsd:element>
2435 <xsd:element name="MessagePartNRInformation">
2436     <xsd:complexType>
2437         <xsd:choice>
2438             <xsd:element name="MessagePartIdentifier" type="bpssignal:non-empty-string"/>
2439             <xsd:element ref="ds:Reference"/>
2440         </xsd:choice>
2441     </xsd:complexType>
2442 </xsd:element>
2443 <xsd:element name="ProcessingAcknowledgment">
2444     <xsd:annotation>
2445         <xsd:documentation/>
2446     </xsd:annotation>
2447     <xsd:complexType>
2448         <xsd:complexContent>
2449             <xsd:extension base="bpssignal:SignalIdentificationInformation">
2450                 <xsd:sequence>
2451                     <xsd:element ref="ds:Signature" minOccurs="0"/>
2452                     <xsd:any namespace="##other" minOccurs="0"/>
2453                 </xsd:sequence>
2454             </xsd:extension>
2455         </xsd:complexContent>
2456     </xsd:complexType>
2457 </xsd:element>
2458 </xsd:schema>

```

2459 **7.2 Documentation for Signal Schema**

| | | |
|---|---|----------------------------------|
| Elements | Complex types | Simple types |
| ProcessingAcknowledgment | PartyInfoType | non-empty-string |
| Exception | ProcessSpecificationType | |
| MessagePartNRInformation | RoleType | |
| NonRepudiationInformation | SignalIdentificationInformation | |
| ReceiptAcknowledgment | | |

2460
2461 **7.2.1 element ProcessingAcknowledgment**

| | |
|-----------|---|
| diagram |  |
| namespace | http://www.untmg.org/BusinessProcess/BPSS_SIGNALS |
| type | extension of SignalIdentificationInformation |
| children | OriginalMessageIdentifier OriginalDocumentIdentifier FromPartyInfo ToPartyInfo FromRole ToRole OriginalMessageDateTime ThisMessageDateTime ProcessSpecificationInfo ds:Signature any ##other |
| source | <pre><xsd:element name="ProcessingAcknowledgment"> <xsd:annotation> <xsd:documentation/> </xsd:annotation> <xsd:complexType> <xsd:complexContent> <xsd:extension base="bpssignal:SignalIdentificationInformation"> <xsd:sequence> <xsd:element ref="ds:Signature" minOccurs="0"/> <xsd:any namespace="##other" minOccurs="0"/> </xsd:sequence> </xsd:extension> </xsd:complexContent> </xsd:complexType> </xsd:element></pre> |

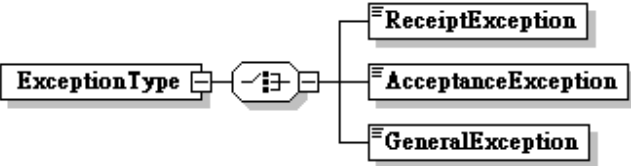
2462 7.2.2 element Exception

| | |
|-----------|---|
| diagram | <p>The diagram shows the structure of the Exception element. It is a complex type that extends SignalIdentificationInformation. The Exception element contains a sequence of elements: OriginalMessageIdentifier, OriginalDocumentIdentifier, FromPartyInfo, ToPartyInfo, FromRole, ToRole, OriginalMessageDateTime, ThisMessageDateTime, and ProcessSpecificationInfo. Additionally, it contains a choice of elements: ExceptionType, Reason, ExceptionMessage, and any ##other.</p> |
| namespace | http://www.untmg.org/BusinessProcess/BPSS_SIGNALS |
| type | extension of SignalIdentificationInformation |
| children | OriginalMessageIdentifier OriginalDocumentIdentifier FromPartyInfo ToPartyInfo FromRole ToRole OriginalMessageDateTime ThisMessageDateTime ProcessSpecificationInfo ExceptionType Reason ExceptionMessage |
| source | <pre> <xsd:element name="Exception"> <xsd:complexType> <xsd:complexContent> <xsd:extension base="bpssignal:SignalIdentificationInformation"> <xsd:sequence> <xsd:element name="ExceptionType"> <xsd:complexType> <xsd:choice> <xsd:element name="ReceiptException"> <xsd:simpleType> <xsd:restriction base="xsd:string"> <xsd:enumeration value="Syntax"/> <xsd:enumeration value="Authorization"/> <xsd:enumeration value="Signature"/> <xsd:enumeration value="Sequence"/> </xsd:restriction> </xsd:simpleType> </xsd:element> <xsd:element name="ProcessingException"> <xsd:simpleType> <xsd:restriction base="xsd:string"> <xsd:enumeration value="Business"/> <xsd:enumeration value="Performance"/> </xsd:restriction> </xsd:simpleType> </xsd:element> </xsd:choice> </xsd:complexType> </xsd:element> </xsd:sequence> </xsd:extension> </xsd:complexContent> </xsd:complexType> </xsd:element> </pre> |

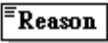
```
</xsd:restriction>
</xsd:simpleType>
</xsd:element>
<xsd:element name="GeneralException">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string"/>
  </xsd:simpleType>
</xsd:element>
</xsd:choice>
</xsd:complexType>
</xsd:element>
<xsd:element name="Reason" type="bpssignal:non-empty-string"/>
<xsd:element name="ExceptionMessage" type="bpssignal:non-empty-string" minOccurs="0"/>
<xsd:any namespace="##other" minOccurs="0"/>
</xsd:sequence>
</xsd:extension>
</xsd:complexContent>
</xsd:complexType>
</xsd:element>
```

2463
2464
2465

7.2.2.1 element Exception/ExceptionType

| | |
|-----------|---|
| diagram |  |
| namespace | http://www.untmg.org/BusinessProcess/BPSS_SIGNALS |
| children | ReceiptException ProcessingException GeneralException |
| source | <pre><xsd:element name="ExceptionType"> <xsd:complexType> <xsd:choice> <xsd:element name="ReceiptException"> <xsd:simpleType> <xsd:restriction base="xsd:string"> <xsd:enumeration value="Syntax"/> <xsd:enumeration value="Authorization"/> <xsd:enumeration value="Signature"/> <xsd:enumeration value="Sequence"/> </xsd:restriction> </xsd:simpleType> </xsd:element> <xsd:element name="ProcessingException"> <xsd:simpleType> <xsd:restriction base="xsd:string"> <xsd:enumeration value="Business"/> <xsd:enumeration value="Performance"/> </xsd:restriction> </xsd:simpleType> </xsd:element> <xsd:element name="GeneralException"> <xsd:simpleType> <xsd:restriction base="xsd:string"/> </xsd:simpleType> </xsd:element> </xsd:choice> </xsd:complexType> </xsd:element></pre> |


2466 **7.2.2.2 element Exception/Reason**

| | |
|-----------|---|
| diagram |  |
| namespace | http://www.untmg.org/BusinessProcess/BPSS_SIGNALS |
| type | non-empty-string |
| facets | minLength 1 |
| source | <xsd:element name="Reason" type="bpssignal:non-empty-string"/> |

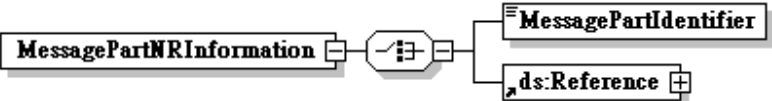
2467

2468

2469 **7.2.2.3 element Exception/ExceptionMessage**

| | |
|-----------|--|
| diagram |  |
| namespace | http://www.untmg.org/BusinessProcess/BPSS_SIGNALS |
| type | non-empty-string |
| facets | minLength 1 |
| source | <xsd:element name="ExceptionMessage" type="bpssignal:non-empty-string" minOccurs="0"/> |

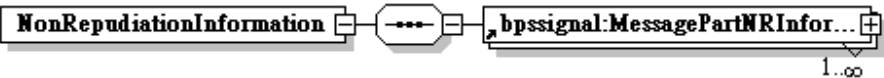
2470 **7.2.3 element MessagePartNRInformation**

| | |
|-----------|---|
| diagram |  |
| namespace | http://www.untmg.org/BusinessProcess/BPSS_SIGNALS |
| children | MessagePartIdentifier ds:Reference |
| used by | element NonRepudiationInformation |
| source | <pre> <xsd:element name="MessagePartNRInformation"> <xsd:complexType> <xsd:choice> <xsd:element name="MessagePartIdentifier" type="bpssignal:non-empty-string"/> <xsd:element ref="ds:Reference"/> </xsd:choice> </xsd:complexType> </xsd:element> </pre> |

2471

2472

2473 **7.2.4 element NonRepudiationInformation**

| | |
|-----------|--|
| diagram |  |
| namespace | http://www.untmg.org/BusinessProcess/BPSS_SIGNALS |
| children | MessagePartNRInformation |
| used by | element ReceiptAcknowledgmen t |
| source | <pre> <xsd:element name="NonRepudiationInformation"> <xsd:complexType> <xsd:sequence> <xsd:element ref="bpssignal:MessagePartNRInformation" maxOccurs="unbounded"/> </xsd:sequence> </xsd:complexType> </xsd:element> </pre> |

2474 **7.2.5 element ReceiptAcknowledgment**

| | |
|-----------|---|
| diagram | |
| namespace | http://www.untmg.org/BusinessProcess/BPSS_SIGNALS |
| type | extension of SignalIdentificationInformation |
| children | OriginalMessageIdentifier OriginalDocumentIdentifier FromPartyInfo ToPartyInfo FromRole ToRole OriginalMessageDateTime ThisMessageDateTime ProcessSpecificationInfo NonRepudiationInformation ds:Signature |
| source | <pre> <xsd:element name="ReceiptAcknowledgment"> <xsd:complexType> <xsd:complexContent> <xsd:extension base="bpssignal:SignalIdentificationInformation"> <xsd:sequence> <xsd:element ref="bpssignal:NonRepudiationInformation" minOccurs="0"/> <xsd:element ref="ds:Signature" minOccurs="0"/> <xsd:any namespace="##other" minOccurs="0"/> </xsd:sequence> </xsd:extension> </xsd:complexContent> </xsd:complexType> </xsd:element> </pre> |

2475 **7.2.6 complexType PartyInfoType**

| | | | | | | |
|------------|---|------|-----|---------|-------|------------|
| diagram | PartyInfoType | | | | | |
| namespace | http://www.untmg.org/BusinessProcess/BPSS_SIGNALS | | | | | |
| type | extension of non-empty-string | | | | | |
| used by | elements SignalIdentificationInformation/FromPartyInfo SignalIdentificationInformation/ToPartyInfo | | | | | |
| facets | minLength 1 | | | | | |
| attributes | Name | Type | Use | Default | Fixed | Annotation |
| type | non-empty-string | | | | | |
| source | <pre><xsd:complexType name="PartyInfoType"> <xsd:simpleContent> <xsd:extension base="non-empty-string"> <xsd:attribute name="type" type="non-empty-string"/> </xsd:extension> </xsd:simpleContent> </xsd:complexType></pre> | | | | | |

2476

2477

2478 **7.2.7 complexType ProcessSpecificationType**

| | | | | | | |
|------------|---|------|-----|---------|-------|------------|
| diagram | ProcessSpecificationType | | | | | |
| namespace | http://www.untmg.org/BusinessProcess/BPSS_SIGNALS | | | | | |
| used by | element SignalIdentificationInformation/ProcessSpecificationInfo | | | | | |
| attributes | Name | Type | Use | Default | Fixed | Annotation |
| version | non-empty-string | | | | | |
| name | non-empty-string | | | | | |
| xlink:type | xsd:NMTOKEN | | | | | |
| xlink:href | xsd:anyURI | | | | | |
| nameID | xsd:anyURI | | | | | |
| source | <pre><xsd:complexType name="ProcessSpecificationType"> <xsd:attribute name="version" type="non-empty-string"/> <xsd:attribute name="name" type="non-empty-string"/> <xsd:attributeGroup ref="xlink.grp"/> <xsd:attribute name="nameID" type="xsd:anyURI"/> </xsd:complexType></pre> | | | | | |

2479 **7.2.8 complexType RoleType**

| | | | | | | |
|------------|---|--|----------|---------|--------|------------|
| diagram | <div><div>Role Type</div><p>This type defines the structure for Role Definition.</p></div> | | | | | |
| namespace | http://www.untmg.org/BusinessProcess/BPSS_SIGNALS | | | | | |
| used by | elements | SignalIdentificationInformation/FromRole SignalIdentificationInformation/ToRole | | | | |
| attributes | Name | Type | Use | Default | Fixed | Annotation |
| | name | non-empty-string | required | | | |
| | xlink:type | xsd:NMTOKEN | | | simple | |
| | xlink:href | xsd:anyURI | required | | | |
| annotation | documentation | This type defines the structure for Role Definition. | | | | |
| source | <xsd:complexType name="RoleType"> <xsd:annotation> <xsd:documentation> This type defines the structure for Role Definition. </xsd:documentation> </xsd:annotation> <xsd:attribute name="name" type="non-empty-string" use="required"/> <xsd:attributeGroup ref="xlink.grp"/> </xsd:complexType> | | | | | |

2480 **7.2.9 complexType SignalIdentificationInformation**

| | |
|-----------|---|
| diagram | |
| namespace | http://www.untmg.org/BusinessProcess/BPSS_SIGNALS |
| children | OriginalMessageIdentifier OriginalDocumentIdentifier FromPartyInfo ToPartyInfo FromRole ToRole OriginalMessageDateTime ThisMessageDateTime ProcessSpecificationInfo |
| used by | elements ProcessingAcknowledgment Exception ReceiptAcknowledgment |
| source | <pre><xsd:complexType name="SignalIdentificationInformation"> <xsd:sequence> <xsd:element name="OriginalMessageIdentifier" type="bpssignal:non-empty-string"/> <xsd:element name="OriginalDocumentIdentifier" type="bpssignal:non-empty-string" minOccurs="0"/> <xsd:element name="FromPartyInfo" type="bpssignal:PartyInfoType"/> <xsd:element name="ToPartyInfo" type="bpssignal:PartyInfoType"/> <xsd:element name="FromRole" type="bpssignal:RoleType"/> <xsd:element name="ToRole" type="bpssignal:RoleType"/> <xsd:element name="OriginalMessageDateTime" type="xsd:dateTime"/> <xsd:element name="ThisMessageDateTime" type="xsd:dateTime"/> <xsd:element name="ProcessSpecificationInfo" type="bpssignal:ProcessSpecificationType"/> </xsd:sequence> </xsd:complexType></pre> |

2481
2482
2483

7.2.9.1 element SignalIdentificationInformation/OriginalMessageIdentifier


| | |
|-----------|--|
| diagram | |
| namespace | http://www.untmg.org/BusinessProcess/BPSS_SIGNALS |
| type | non-empty-string |
| facets | minLength 1 |
| source | <pre><xsd:element name="OriginalMessageIdentifier" type="bpssignal:non-empty-string"/></pre> |

2484
2485
2486

7.2.9.2 element SignalIdentificationInformation/OriginalDocumentIdentifier

| | |
|-----------|---|
| diagram | |
| namespace | http://www.untmg.org/BusinessProcess/BPSS_SIGNALS |
| type | non-empty-string |
| facets | minLength 1 |
| source | <pre><xsd:element name="OriginalDocumentIdentifier" type="bpssignal:non-empty-string" minOccurs="0"/></pre> |


2487 **7.2.9.3 element SignalIdentificationInformation/FromPartyInfo**

| | | | | | | |
|------------|---|------------------|-----|---------|-------|------------|
| diagram |  | | | | | |
| namespace | http://www.untmg.org/BusinessProcess/BPSS_SIGNALS | | | | | |
| type | PartyInfoType | | | | | |
| facets | minLength 1 | | | | | |
| attributes | Name | Type | Use | Default | Fixed | Annotation |
| | type | non-empty-string | | | | |
| source | <xsd:element name="FromPartyInfo" type="bpssignal:PartyInfoType"/> | | | | | |

2488

2489


2490 **7.2.9.4 element SignalIdentificationInformation/ToPartyInfo**

| | | | | | | |
|------------|---|------------------|-----|---------|-------|------------|
| diagram |  | | | | | |
| namespace | http://www.untmg.org/BusinessProcess/BPSS_SIGNALS | | | | | |
| type | PartyInfoType | | | | | |
| facets | minLength 1 | | | | | |
| attributes | Name | Type | Use | Default | Fixed | Annotation |
| | type | non-empty-string | | | | |
| source | <xsd:element name="ToPartyInfo" type="bpssignal:PartyInfoType"/> | | | | | |

2491

2492

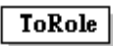
2493 **7.2.9.5 element SignalIdentificationInformation/FromRole**

| | | | | | | |
|------------|---|------------------|----------|---------|--------|------------|
| diagram |  | | | | | |
| namespace | http://www.untmg.org/BusinessProcess/BPSS_SIGNALS | | | | | |
| type | RoleType | | | | | |
| attributes | Name | Type | Use | Default | Fixed | Annotation |
| | name | non-empty-string | required | | | |
| | xlink:type | xsd:NMTOKEN | | | simple | |
| | xlink:href | xsd:anyURI | required | | | |
| source | <xsd:element name="FromRole" type="bpssignal:RoleType"/> | | | | | |

2494

2495

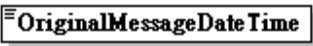
2496 **7.2.9.6 element SignalIdentificationInformation/ToRole**

| | | | | | | |
|------------|---|------------------|----------|---------|--------|------------|
| diagram |  | | | | | |
| namespace | http://www.untmg.org/BusinessProcess/BPSS_SIGNALS | | | | | |
| type | RoleType | | | | | |
| attributes | Name | Type | Use | Default | Fixed | Annotation |
| | name | non-empty-string | required | | | |
| | xlink:type | xsd:NMTOKEN | | | simple | |
| | xlink:href | xsd:anyURI | required | | | |
| source | <xsd:element name="ToRole" type="bpssignal:RoleType"/> | | | | | |

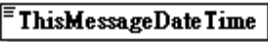
2497

2498

2499 **7.2.9.7 element SignalIdentificationInformation/OriginalMessageDateTime**

| | | | | | | |
|-----------|---|--|--|--|--|--|
| diagram |  | | | | | |
| namespace | http://www.untmg.org/BusinessProcess/BPSS_SIGNALS | | | | | |
| type | xsd:dateTime | | | | | |
| source | <xsd:element name="OriginalMessageDateTime" type="xsd:dateTime"/> | | | | | |

2500 **7.2.9.8 element SignalIdentificationInformation/ThisMessageDateTime**

| | |
|-----------|---|
| diagram |  |
| namespace | http://www.untmg.org/BusinessProcess/BPSS_SIGNALS |
| type | xsd:dateTime |
| source | <xsd:element name="ThisMessageDateTime" type="xsd:dateTime"/> |

2501
2502
2503

7.2.9.9 element SignalIdentificationInformation/ProcessSpecificationInfo

| | | | | | | |
|------------|--|------------------|----------|---------|--------|------------|
| diagram | <div><div>ProcessSpecificationInfo</div></div> | | | | | |
| namespace | http://www.untmg.org/BusinessProcess/BPSS_SIGNALS | | | | | |
| type | ProcessSpecificationType | | | | | |
| attributes | Name | Type | Use | Default | Fixed | Annotation |
| | version | non-empty-string | | | | |
| | name | non-empty-string | | | | |
| | xlink:type | xsd:NMTOKEN | | | simple | |
| | xlink:href | xsd:anyURI | required | | | |
| | nameID | xsd:anyURI | | | | |
| source | <xsd:element name="ProcessSpecificationInfo" type="bpssignal:ProcessSpecificationType"/> | | | | | |

2504 **7.2.10 simpleType non-empty-string**

| | | | |
|-----------|--|---|--|
| namespace | http://www.untmg.org/BusinessProcess/BPSS_SIGNALS | | |
| type | restriction of xsd:string | | |
| used by | elements | Exception/ExceptionMessage MessagePartNRInformation/MessagePartIdentifier SignalIdentificationInformation/OriginalDocumentIdentifier SignalIdentificationInformation/OriginalMessageIdentifier Exception/Reason | |
| | complexType | PartyInfoType | |
| | attributes | RoleType/@name ProcessSpecificationType/@name PartyInfoType/@type ProcessSpecificationType/@version | |
| facets | minLength | 1 | |
| source | <xsd:simpleType name="non-empty-string"> <xsd:restriction base="xsd:string"> <xsd:minLength value="1"/> </xsd:restriction> </xsd:simpleType> | | |

8.0 Production Rules

This section provides a set of production rules, defining the mapping from the UML version of the *Business Process Specification Schema* to the XML version.

The primary purpose for these production rules is to govern the one-time generation of the schema version of the *Business Process Specification Schema* from the UML Class Diagram version of *Business Process Specification Schema*.

The Class Diagram version of *Business Process Specification Schema* is not intended for the direct creation of ebXML Business Process Specifications. However, if a *Business Process Specification* was in fact (programmatically) created as an instance of this class diagram, the production rules would also provide the prescriptive definition necessary to translate a such an instance into a XML Specification Document conformant with theSchema. The production rules are defined for concrete classes, abstract classes, aggregate associations, specialization associations and unidirectional associations.

1. Classes are rendered as XML elements.
2. Class attributes are rendered as XML attributes. NOTE: occurrence requirements (required vs optional) and default values for attributes are not modeled.
3. Specialization classes (classes that inherit from another class) are rendered as XML elements including all attributes and aggregate associations from the base class. Repeated attributes are normalized to a single occurrence.
4. Abstract classes are not rendered in the XML Schema. Abstract classes are inherited from and represent a form of collection. A class that aggregates an abstract class, essentially aggregates “any of each” of the specialization classes.
5. An aggregate association renders the aggregated class as an XML child element with appropriate cardinality.
6. A unidirectional association defines an attribute in the originating class of the same name as the class the association points to. This type of attribute is called a “reference attribute” and contains the name of the class it points to. The referenced class must have a “name” attribute.
7. A class attribute data type, that has a class of the same name with stereotype <<Enumeration>> is rendered as an XML attribute enumeration. The Enumeration class does not have an explicit association.
8. A class attribute data type (e.g. Time, URI, Boolean) that has no corresponding class definition is rendered as a string in theSchema. In the XML Schema version these data types are mapped as:

- 2539 ● Time - xsd:duration
- 2540 ● URI - xsd:anyURI
- 2541 ● Boolean - xsd:boolean

- 2542 9. Each class is given an optional “Documentation*” element which is intended for
2543 annotation of the specification instances. This is not modeled.

Appendix A: Sample XML Business Process Specification Schema Instance

```

<?xml version="1.0" encoding="UTF-8"?>
<ProcessSpecification name="Simple" version="1.1" nameID="Simple-2434134"
xmlns="http://www.untmg.org/downloads/General/approved/BPSS-v1pt11.xsd"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.untmg.org/downloads/General/approved/BPSS-v1pt11.xsd
C:\projects\bpss\bpss_1.1\ebBPSS1.08b.xsd">
  <!-- Business Documents -->
  <BusinessDocument name="Catalog Request" specificationLocation="http://www.xyx.com/CatalogReq.xsd"/>
  <BusinessDocument name="Catalog" specificationLocation="http://www.xyx.com/Catalog.xsd"/>
  <BusinessDocument name="Purchase Order" specificationLocation="http://www.xyx.com/PO.xsd"/>
  <BusinessDocument name="PO Acknowledgement" specificationLocation="http://www.xyx.com/POAck.xsd"/>
  <BusinessDocument name="Credit Request" specificationLocation="http://www.xyx.com/CreditReq.xsd"/>
  <BusinessDocument name="Credit Confirm" specificationLocation="http://www.xyx.com/CreditCon.xsd"/>
  <BusinessDocument name="ASN" specificationLocation="http://www.xyx.com/CatalogASN.xsd"/>
  <BusinessDocument name="CreditAdvice" specificationLocation="http://www.xyx.com/CreditAdvice.xsd"/>
  <BusinessDocument name="DebitAdvice" specificationLocation="http://www.xyx.com/DebitAdvice.xsd"/>
  <BusinessDocument name="Invoice" specificationLocation="http://www.xyx.com/Invoice.xsd"/>
  <BusinessDocument name="Payment" specificationLocation="http://www.xyx.com/Payment.xsd"/>
  <BusinessDocument name="Inventory Report Request" specificationLocation="http://www.xyx.com/InvReq.xsd"/>
  <BusinessDocument name="Inventory Report" specificationLocation="http://www.xyx.com/InvRep.xsd"/>
  <Package name="Ordering">
    <!-- Here are all the Business Transactions needed -->
    <BusinessTransaction name="Catalog Request">
      <RequestingBusinessActivity name="RequestCatalog">
        <DocumentEnvelope businessDocument="Catalog Request"/>
      </RequestingBusinessActivity>
      <RespondingBusinessActivity name="SendCatalog">
        <DocumentEnvelope isPositiveResponse="true" businessDocument="Catalog"/>
      </RespondingBusinessActivity>
    </BusinessTransaction>
    <BusinessTransaction name="Create Order">
      <RequestingBusinessActivity name="SendOrder" isNonRepudiationRequired="true"
timeToAcknowledgeReceipt="P2D" timeToAcknowledgeProcessing="P3D">
        <DocumentEnvelope businessDocument="Purchase Order"/>
      </RequestingBusinessActivity>
      <RespondingBusinessActivity name="SendPOAcknowledgement" isNonRepudiationRequired="true"
timeToAcknowledgeReceipt="P5D">
        <DocumentEnvelope isPositiveResponse="true" businessDocument="PO
Acknowledgement"/>
      </RespondingBusinessActivity>
    </BusinessTransaction>
    <BusinessTransaction name="Check Credit ">
      <RequestingBusinessActivity name="CreditCheck">
        <DocumentEnvelope businessDocument="Credit Request"/>
      </RequestingBusinessActivity>
      <RespondingBusinessActivity name="ConfirmCredit">
        <DocumentEnvelope isPositiveResponse="true" businessDocument="Credit Confirm"/>
      </RespondingBusinessActivity>
    </BusinessTransaction>
    <BusinessTransaction name="Notify of advance shipment">
      <RequestingBusinessActivity name="AdvanceShipmentNotification">
        <DocumentEnvelope businessDocument="ASN"/>
      </RequestingBusinessActivity>
      <RespondingBusinessActivity name="ASNResponse"/>
    </BusinessTransaction>
    <BusinessTransaction name="Process Credit Payment">
      <RequestingBusinessActivity name="CreditPaymentProcess">

```



```

2603     <DocumentEnvelope businessDocument="CreditAdvice"/>
2604 </RequestingBusinessActivity>
2605 <RespondingBusinessActivity name="CreditPaymentProcessResponse">
2606     <DocumentEnvelope isPositiveResponse="true" businessDocument="DebitAdvice"/>
2607 </RespondingBusinessActivity>
2608 </BusinessTransaction>
2609 <BusinessTransaction name="Process Payment">
2610     <RequestingBusinessActivity name="PaymentProcess">
2611         <DocumentEnvelope businessDocument="Invoice"/>
2612     </RequestingBusinessActivity>
2613     <RespondingBusinessActivity name="SendPayment">
2614         <DocumentEnvelope isPositiveResponse="true" businessDocument="Payment"/>
2615     </RespondingBusinessActivity>
2616 </BusinessTransaction>
2617 <BusinessTransaction name="Request Inventory Report">
2618     <RequestingBusinessActivity name="">
2619         <DocumentEnvelope businessDocument="Inventory Report Request"/>
2620     </RequestingBusinessActivity>
2621     <RespondingBusinessActivity name="Inventory Report">
2622         <DocumentEnvelope businessDocument="Inventory Report"/>
2623     </RespondingBusinessActivity>
2624 </BusinessTransaction>
2625 <!-- Now the Binary Collaborations -->
2626 <BinaryCollaboration name="Request Catalog" initiatingRoleID="1122B1">
2627     <Role name="requestor" nameID="1122B1"/>
2628     <Role name="provider" nameID="2211A1"/>
2629     <Start toBusinessState="Catalog Request"/>
2630     <BusinessTransactionActivity name="Catalog Request" businessTransaction="Catalog Request"
2631 fromRole="requestor" toRole="provider"/>
2632     <Success fromBusinessState="Catalog Request" conditionGuard="Success"/>
2633     <Failure fromBusinessState="Catalog Request" conditionGuard="Failure"/>
2634 </BinaryCollaboration>
2635 <BinaryCollaboration name="Firm Order" timeToPerform="P2D" initiatingRoleID="1122B2">
2636     <Documentation>timeToPerform = Period: 2 days from start of transaction</Documentation>
2637     <Role name="buyer" nameID="1122B2"/>
2638     <Role name="seller" nameID="1122B3"/>
2639     <Start toBusinessState="Create Order"/>
2640     <BusinessTransactionActivity name="Create Order" businessTransaction="Create Order" fromRole="buyer"
2641 toRole="seller"/>
2642     <Success fromBusinessState="Create Order" conditionGuard="Success"/>
2643     <Failure fromBusinessState="Create Order" conditionGuard="Failure"/>
2644 </BinaryCollaboration>
2645 <BinaryCollaboration name="Product Fulfillment" timeToPerform="P5D" initiatingRoleID="1122B2">
2646     <Documentation>timeToPerform = Period: 5 days from start of transaction</Documentation>
2647     <Role name="buyer" nameID="1122B2"/>
2648     <Role name="seller" nameID="1122B3"/>
2649     <Start toBusinessState="Create Order"/>
2650     <BusinessTransactionActivity name="Create Order" businessTransaction="Create Order" fromRole="buyer"
2651 toRole="seller"/>
2652     <BusinessTransactionActivity name="Notify shipment" businessTransaction="Notify of advance
2653 shipment" fromRole="seller" toRole="buyer"/>
2654     <Success fromBusinessState="Notify shipment" conditionGuard="Success"/>
2655     <Failure fromBusinessState="Notify shipment" conditionGuard="Failure"/>
2656     <Transition fromBusinessState="Create Order" toBusinessState="Notify shipment"/>
2657 </BinaryCollaboration>
2658 <BinaryCollaboration name="Inventory Status" initiatingRoleID="1122B1">
2659     <Role name="requestor" nameID="1122B1"/>
2660     <Role name="provider" nameID="2211A1"/>
2661     <Start toBusinessState="Inventory Report Request"/>
2662     <BusinessTransactionActivity name="Inventory Report Request" businessTransaction="Inventory
2663 Report Request" fromRole="requestor" toRole="provider"/>
2664     <Success fromBusinessState="Inventory Report Request" conditionGuard="Success"/>

```

```

2665         <Failure fromBusinessState=" Inventory Report Request " conditionGuard="Failure"/>
2666     </BinaryCollaboration>
2667     <BinaryCollaboration name="Credit Inquiry" initiatingRoleID="9122B1">
2668         <Role name="creditor" nameID="9122B1"/>
2669         <Role name="credit service" nameID="8122B1"/>
2670         <Start toBusinessState="Check Credit"/>
2671         <BusinessTransactionActivity name="Check Credit" businessTransaction="Check Credit"
2672 fromRole="creditor" toRole="credit service"/>
2673         <Success fromBusinessState="Check Credit" conditionGuard="Success"/>
2674         <Failure fromBusinessState="Check Credit" conditionGuard="Failure"/>
2675     </BinaryCollaboration>
2676     <BinaryCollaboration name="Credit Payment" initiatingRoleID="6122B1">
2677         <Role name="payee" nameID="6122B1"/>
2678         <Role name="payor" nameID="7122B1"/>
2679         <Start toBusinessState="Process Credit Payment"/>
2680         <BusinessTransactionActivity name="Process Credit Payment" businessTransaction="Process
2681 Credit Payment" fromRole="payee" toRole="payor"/>
2682         <Success fromBusinessState="Process Credit Payment" conditionGuard="Success"/>
2683         <Failure fromBusinessState="Process Credit Payment" conditionGuard="Failure"/>
2684     </BinaryCollaboration>
2685     <!-- A compound BinaryCollaboration for illustration purposes-->
2686     <BinaryCollaboration name="Credit Charge" initiatingRoleID="8132B1">
2687         <Role name="charger" nameID="8132B1"/>
2688         <Role name="credit service" nameID="8122B1"/>
2689         <Start toBusinessState="Credit Inquiry"/>
2690         <CollaborationActivity name="Credit Inquiry" binaryCollaboration="Credit Inquiry" fromRole="charger"
2691 toRole="credit service"/>
2692         <CollaborationActivity name="Credit Payment" binaryCollaboration="Credit Payment" fromRole="charger"
2693 toRole="payor"/>
2694         <Success fromBusinessState="Credit Payment" conditionGuard="Success"/>
2695         <Failure fromBusinessState="Credit Payment" conditionGuard="Failure"/>
2696         <Transition fromBusinessState="Credit Inquiry" toBusinessState="Credit Payment"/>
2697     </BinaryCollaboration>
2698     <BinaryCollaboration name="Fulfillment Payment" initiatingRoleID="6122B1">
2699         <Role name="payee" nameID="6122B1"/>
2700         <Role name="payor" nameID="7122B1"/>
2701         <Start toBusinessState="Process Payment"/>
2702         <BusinessTransactionActivity name="Process Payment" businessTransaction="Process
2703 Payment" fromRole="payee" toRole="payor"/>
2704         <Success fromBusinessState="Process Payment" conditionGuard="Success"/>
2705         <Failure fromBusinessState="Process Payment" conditionGuard="Failure"/>
2706     </BinaryCollaboration>
2707     <!-- First the overall MultiParty Collaboration -->
2708     <MultiPartyCollaboration name="DropShip">
2709         <BusinessPartnerRole name="Customer">
2710             <Performs role="requestor" roleIDREF="1122B1"/>
2711             <Performs role="buyer" roleIDREF="1122B2"/>
2712             <Transition fromBusinessState="Catalog Request" toBusinessState="Create Order"/>
2713         </BusinessPartnerRole>
2714         <BusinessPartnerRole name="Retailer">
2715             <Performs role="provider" roleIDREF="2211A1"/>
2716             <Performs role="seller" roleIDREF="1122B3"/>
2717             <Performs role="creditor" roleIDREF="9122B1"/>
2718             <Performs role="buyer" roleIDREF="1122B2"/>
2719             <Performs role="payee" roleIDREF="6122B1"/>
2720             <Performs role="payor" roleIDREF="7122B1"/>
2721             <Performs role="requestor" roleIDREF="1122B1"/>
2722             <Transition fromBusinessState="Create Order" toBusinessState="Check Credit"/>
2723             <Transition fromBusinessState="Check Credit" toBusinessState="Credit Payment"/>
2724         </BusinessPartnerRole>
2725         <BusinessPartnerRole name="DropShip Vendor">
2726             <Performs role="seller" roleIDREF="1122B3"/>

```

```
2727     <Performs role="payee" roleIDREF="6122B1"/>
2728     <Performs role="provider" roleIDREF="2211A1"/>
2729   </BusinessPartnerRole>
2730   <BusinessPartnerRole name="Credit Authority">
2731     <Performs role="credit service" roleIDREF="8122B1"/>
2732     <Performs role="payor" roleIDREF="7122B1"/>
2733   </BusinessPartnerRole>
2734 </MultiPartyCollaboration>
2735 </Package>
2736 </ProcessSpecification>
```

2737 Appendix B: Sample XML Signals

```

2738 <?xml version="1.0" encoding="UTF-8"?>
2739 <bpssignal:ReceiptAcknowledgment xmlns:bpssignal="http://www.untmg.org/BusinessProcess/BPSS_SIGNALS"
2740 xmlns:ds="http://www.w3.org/2000/09/xmldsig#" xmlns:xsd="http://www.w3.org/2001/XMLSchema-instance"
2741 xmlns:xlink="http://www.w3.org/1999/xlink"
2742 xsd:schemaLocation="http://www.untmg.org/BusinessProcess/BPSS_SIGNALS BPSS_Signals.xsd">
2743   <bpssignal:OriginalMessageIdentifier>MessageIdentifier-1</bpssignal:OriginalMessageIdentifier>
2744   <bpssignal:FromPartyInfo bpssignal:type="DUNS.com">PartyA</bpssignal:FromPartyInfo>
2745   <bpssignal:ToPartyInfo bpssignal:type="DUNS.com">PartyB</bpssignal:ToPartyInfo>
2746   <bpssignal:FromRole bpssignal:name="Buyer" xlink:type="simple"
2747   xlink:href="http://www.rosettanet.org/processes/3A4.xml#Buyer"/>
2748   <bpssignal:ToRole bpssignal:name="Seller" xlink:type="simple"
2749   xlink:href="http://www.rosettanet.org/processes/3A4.xml#Seller"/>
2750   <bpssignal:OriginalMessageDateTime>2002-03-05T19:00:00</bpssignal:OriginalMessageDateTime>
2751   <bpssignal:ThisMessageDateTime>2002-03-05T20:00:00</bpssignal:ThisMessageDateTime>
2752   <bpssignal:ProcessSpecificationInfo bpssignal:version="2.0" bpssignal:name="PIP3A4RequestPurchaseOrder"
2753   xlink:type="simple" xlink:href="http://www.rosettanet.org/processes/3A4.xml"
2754   bpssignal:nameID="urn:icann:rosettanet.org:bpid:3A4$2.0"/>
2755   <bpssignal:NonRepudiationInformation>
2756     <bpssignal:MessagePartNRInformation>
2757       <ds:Reference ds:URI="cid://Message-Part-1">
2758         <ds:DigestMethod ds:Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
2759         <ds:DigestValue>R0lGODlhcgGSALMAAAQCAEMmCZtuMFQxDS8bd012</ds:DigestValue>
2760       </ds:Reference>
2761     </bpssignal:MessagePartNRInformation>
2762     <bpssignal:MessagePartNRInformation>
2763       <ds:Reference ds:URI="cid://Message-Part-2">
2764         <ds:DigestMethod ds:Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
2765         <ds:DigestValue>afde1AbcgGSALMAAAQCAEMmCZtuMFQxDS8be</ds:DigestValue>
2766       </ds:Reference>
2767     </bpssignal:MessagePartNRInformation>
2768     <bpssignal:MessagePartNRInformation>
2769       <bpssignal:MessagePartIdentifier>Message-Part-3</bpssignal:MessagePartIdentifier>
2770     </bpssignal:MessagePartNRInformation>
2771   </bpssignal:NonRepudiationInformation>
2772 </bpssignal:ReceiptAcknowledgment>
2773 <?xml version="1.0" encoding="UTF-8"?>
2774 <bpssignal:ProcessingAcknowledgment xmlns:bpssignal="http://www.untmg.org/BusinessProcess/BPSS_SIGNALS"
2775 xmlns:xsd="http://www.w3.org/2001/XMLSchema-instance" xmlns:xlink="http://www.w3.org/1999/xlink"
2776 xsd:schemaLocation="http://www.untmg.org/BusinessProcess/BPSS_SIGNALS BPSS_Signals.xsd">
2777   <bpssignal:OriginalMessageIdentifier>MessageIdentifier-1</bpssignal:OriginalMessageIdentifier>
2778   <bpssignal:FromPartyInfo bpssignal:type="DUNS.com">PartyA</bpssignal:FromPartyInfo>
2779   <bpssignal:ToPartyInfo bpssignal:type="DUNS.com">PartyB</bpssignal:ToPartyInfo>
2780   <bpssignal:FromRole bpssignal:name="Buyer" xlink:type="simple"
2781   xlink:href="http://www.rosettanet.org/processes/3A4.xml#Buyer"/>
2782   <bpssignal:ToRole bpssignal:name="Seller" xlink:type="simple"
2783   xlink:href="http://www.rosettanet.org/processes/3A4.xml#Seller"/>
2784   <bpssignal:OriginalMessageDateTime>2002-03-05T19:00:00</bpssignal:OriginalMessageDateTime>
2785   <bpssignal:ThisMessageDateTime>2002-03-05T20:00:00</bpssignal:ThisMessageDateTime>
2786   <bpssignal:ProcessSpecificationInfo bpssignal:version="2.0" bpssignal:name="PIP3A4RequestPurchaseOrder"
2787   xlink:type="simple" xlink:href="http://www.rosettanet.org/processes/3A4.xml"
2788   bpssignal:nameID="urn:icann:rosettanet.org:bpid:3A4$2.0"/>
2789 </bpssignal:ProcessingAcknowledgment>
2790 <?xml version="1.0" encoding="UTF-8"?>
2791 <bpssignal:Exception xmlns:bpssignal="http://www.untmg.org/BusinessProcess/BPSS_SIGNALS"
2792 xmlns:xsd="http://www.w3.org/2001/XMLSchema-instance" xmlns:xlink="http://www.w3.org/1999/xlink"
2793 xsd:schemaLocation="http://www.untmg.org/BusinessProcess/BPSS_SIGNALS
2794 BPSS_Signals.xsd">

```

```

2795     <bpssignal:OriginalMessageIdentifier>MessageIdentifier-1</bpssignal:OriginalMessageIdentifier>
2796     <bpssignal:FromPartyInfo bpssignal:type="DUNS.com">PartyA</bpssignal:FromPartyInfo>
2797     <bpssignal:ToPartyInfo bpssignal:type="DUNS.com">PartyB</bpssignal:ToPartyInfo>
2798     <bpssignal:FromRole bpssignal:name="Buyer" xlink:type="simple"
2799     xlink:href="http://www.rosettanet.org/processes/3A4.xml#Buyer"/>
2800     <bpssignal:ToRole bpssignal:name="Seller" xlink:type="simple"
2801     xlink:href="http://www.rosettanet.org/processes/3A4.xml#Seller"/>
2802     <bpssignal:OriginalMessageDateTime>2002-03-05T19:00:00</bpssignal:OriginalMessageDateTime>
2803     <bpssignal:ThisMessageDateTime>2002-03-05T20:00:00</bpssignal:ThisMessageDateTime>
2804     <bpssignal:ProcessSpecificationInfo bpssignal:version="2.0" bpssignal:name="PIP3A4RequestPurchaseOrder"
2805     xlink:type="simple" xlink:href="http://www.rosettanet.org/processes/3A4.xml"
2806     bpssignal:nameID="urn:icann:rosettanet.org:bpid:3A4$2.0"/>
2807     <bpssignal:ExceptionType>
2808         <bpssignal:ReceiptException>Signature</bpssignal:ReceiptException>
2809     </bpssignal:ExceptionType>
2810     <bpssignal:Reason>State transition failure</bpssignal:Reason>
2811     <bpssignal:ExceptionMessage>Signature Validation Failed for request
2812     message</bpssignal:ExceptionMessage>
2813 </bpssignal:Exception>

```

2814 **Copyright Statement**

2815 Copyright © UN/CEFACT 2005. All Rights Reserved.

2816 This document and translations of it may be copied and furnished to others, and derivative
2817 works that comment on or otherwise explain it or assist in its implementation may be
2818 prepared, copied, published and distributed, in whole or in part, without restriction of any
2819 kind, provided that the above copyright notice and this paragraph are included on all such
2820 copies and derivative works. However, this document itself may not be modified in any way,
2821 such as by removing the copyright notice or references to UN/CEFACT except as required
2822 to translate it into languages other than English.

2823 The limited permissions granted above are perpetual and will not be revoked by
2824 UN/CEFACT or its successors or assigns.

2825 This document and the information contained herein is provided on an "AS IS" basis and
2826 UN/CEFACT DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING
2827 BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION
2828 HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF
2829 MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.