Universal Business Language v2.1

Public Review Draft 01

25 September 2010

Specification URIs:

This Version:

http://docs.oasis-open.org/ubl/prd1-UBL-2.1/UBL-2.1.html
http://docs.oasis-open.org/ubl/prd1-UBL-2.1/UBL-2.1.pdf
http://docs.oasis-open.org/ubl/prd1-UBL-2.1/UBL-2.1.xml (Authoritative)

Previous Version:

http://docs.oasis-open.org/ubl/os-UBL-2.0/UBL-2.0.html
http://docs.oasis-open.org/ubl/os-UBL-2.0/UBL-2.0.pdf
http://docs.oasis-open.org/ubl/os-UBL-2.0/UBL-2.0.xml

Latest Version:

http://docs.oasis-open.org/ubl/UBL-2.1.html
http://docs.oasis-open.org/ubl/UBL-2.1.pdf
http://docs.oasis-open.org/ubl/UBL-2.1.xml (Authoritative)

Technical Committee:

OASIS Universal Business Language TC

Chairs:

Jon Bosak, Pinax <bosak@pinax.com>
Tim McGrath <tim.mcgrath@documentengineeringservices.com>

Editors:

Jon Bosak, Pinax <bosak@pinax.com>
Tim McGrath <tim.mcgrath@documentengineeringservices.com>
G. Ken Holman, Crane Softwrights Ltd. <gkholman@CraneSoftwrights.com>

Related Work:

This specification supersedes UBL 2.0.

Declared XML Namespaces:

urn:oasis:names:specification:ubl:schema:xsd:CommonAggregateComponents-2
urn:oasis:names:specification:ubl:schema:xsd:CommonBasicComponents-2
urn:oasis:names:specification:ubl:schema:xsd:CommonExtensionComponents-2
urn:oasis:names:specification:ubl:schema:xsd:CommonSignatureComponents-2
urn:oasis:names:specification:ubl:schema:xsd:QualifiedDataTypes-2
urn:oasis:names:specification:ubl:schema:xsd:SignatureAggregateComponents-2
urn:oasis:names:specification:ubl:schema:xsd:SignatureBasicComponents-2
urn:oasis:names:specification:ubl:schema:xsd:UnqualifiedDataTypes-2

urn:oasis:names:specification:ubl:schema:xsd:ApplicationResponse-2
urn:oasis:names:specification:ubl:schema:xsd:AttachedDocument-2
urn:oasis:names:specification:ubl:schema:xsd:AwardedNotification-2
urn:oasis:names:specification:ubl:schema:xsd:BillOfLading-2
urn:oasis:names:specification:ubl:schema:xsd:CallForTenders-2
urn:oasis:names:specification:ubl:schema:xsd:Catalogue-2
urn:oasis:names:specification:ubl:schema:xsd:CatalogueDeletion-2
urn:oasis:names:specification:ubl:schema:xsd:CatalogueItemSpecificationUpdate-2
urn:oasis:names:specification:ubl:schema:xsd:CataloguePricingUpdate-2
urn:oasis:names:specification:ubl:schema:xsd:CatalogueRequest-2
urn:oasis:names:specification:ubl:schema:xsd:CertificateOfOrigin-2
urn:oasis:names:specification:ubl:schema:xsd:CommonAggregateComponents-2
urn:oasis:names:specification:ubl:schema:xsd:CommonBasicComponents-2
urn:oasis:names:specification:ubl:schema:xsd:CommonExtensionComponents-2
urn:oasis:names:specification:ubl:schema:xsd:ContractAwardNotice-2
urn:oasis:names:specification:ubl:schema:xsd:ContractNotice-2
urn:oasis:names:specification:ubl:schema:xsd:CreditNote-2
urn:oasis:names:specification:ubl:schema:xsd:DebitNote-2
urn:oasis:names:specification:ubl:schema:xsd:DespatchAdvice-2
urn:oasis:names:specification:ubl:schema:xsd:DocumentStatus-2
urn:oasis:names:specification:ubl:schema:xsd:DocumentStatusRequest-2
urn:oasis:names:specification:ubl:schema:xsd:ExceptionCriteria-2
urn:oasis:names:specification:ubl:schema:xsd:ExceptionNotification-2
urn:oasis:names:specification:ubl:schema:xsd:Forecast-2
urn:oasis:names:specification:ubl:schema:xsd:ForecastRevision-2
urn:oasis:names:specification:ubl:schema:xsd:ForwardingInstructions-2
urn:oasis:names:specification:ubl:schema:xsd:FreightInvoice-2
urn:oasis:names:specification:ubl:schema:xsd:GuaranteeCertificate-2
urn:oasis:names:specification:ubl:schema:xsd:InstructionForReturns-2
urn:oasis:names:specification:ubl:schema:xsd:InventoryReport-2
urn:oasis:names:specification:ubl:schema:xsd:Invoice-2
urn:oasis:names:specification:ubl:schema:xsd:ItemInformationRequest-2
urn:oasis:names:specification:ubl:schema:xsd:Order-2
urn:oasis:names:specification:ubl:schema:xsd:OrderCancellation-2
urn:oasis:names:specification:ubl:schema:xsd:OrderChange-2
urn:oasis:names:specification:ubl:schema:xsd:OrderResponse-2
urn:oasis:names:specification:ubl:schema:xsd:OrderResponseSimple-2
urn:oasis:names:specification:ubl:schema:xsd:PackingList-2
urn:oasis:names:specification:ubl:schema:xsd:PerformanceHistory-2
urn:oasis:names:specification:ubl:schema:xsd:PriorInformationNotice-2
urn:oasis:names:specification:ubl:schema:xsd:ProductActivity-2
urn:oasis:names:specification:ubl:schema:xsd:Quotation-2
urn:oasis:names:specification:ubl:schema:xsd:ReceiptAdvice-2
urn:oasis:names:specification:ubl:schema:xsd:Reminder-2
urn:oasis:names:specification:ubl:schema:xsd:RemittanceAdvice-2
urn:oasis:names:specification:ubl:schema:xsd:RequestForQuotation-2
urn:oasis:names:specification:ubl:schema:xsd:RetailEvent-2
urn:oasis:names:specification:ubl:schema:xsd:SelfBilledCreditNote-2
urn:oasis:names:specification:ubl:schema:xsd:SelfBilledInvoice-2
urn:oasis:names:specification:ubl:schema:xsd:Statement-2
urn:oasis:names:specification:ubl:schema:xsd:StockAvailabilityReport-2
urn:oasis:names:specification:ubl:schema:xsd:Tender-2
urn:oasis:names:specification:ubl:schema:xsd:TendererQualification-2
urn:oasis:names:specification:ubl:schema:xsd:TendererQualificationResponse-2
urn:oasis:names:specification:ubl:schema:xsd:TenderReceipt-2
urn:oasis:names:specification:ubl:schema:xsd:TradeItemLocationProfile-2
urn:oasis:names:specification:ubl:schema:xsd:TransportationStatus-2
urn:oasis:names:specification:ubl:schema:xsd:TransportExecutionPlan-2

urn:oasis:names:specification:ubl:schema:xsd:TransportExecutionStatus-2
urn:oasis:names:specification:ubl:schema:xsd:TransportOperationStatus-2
urn:oasis:names:specification:ubl:schema:xsd:UnawardedNotification-2
urn:oasis:names:specification:ubl:schema:xsd:UtilityStatement-2
urn:oasis:names:specification:ubl:schema:xsd:Waybill-2

Abstract:

This specification defines the Universal Business Language, version 2.1.

Status:

This document was last revised or approved by the UBL TC on the above date. The level of approval is also listed above. Check the current location noted above for possible later revisions of this document. This document is updated periodically on no particular schedule.

Technical Committee members should send comments on this specification to the Technical Committee's email list. Others should send comments to the Technical Committee by using the "Send A Comment" button on the Technical Committee's web page at http://www.oasis-open.org/committees/ubl/.

For information on whether any patents have been disclosed that may be essential to implementing this specification, and any offers of patent licensing terms, please refer to the Intellectual Property Rights section of the Technical Committee web page at http://www.oasis-open.org/committees/ubl/ipr.php.

See Appendix A, *Release Notes (Informative)* for more information regarding this release package.

Notices

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

OASIS requests that any OASIS Party or any other party that believes it has patent claims that would necessarily be infringed by implementations of this OASIS Committee Specification or OASIS Standard, to notify OASIS TC Administrator and provide an indication of its willingness to grant patent licenses to such patent claims in a manner consistent with the IPR Mode of the OASIS Technical Committee that produced this specification.

OASIS invites any party to contact the OASIS TC Administrator if it is aware of a claim of ownership of any patent claims that would necessarily be infringed by implementations of this specification by a patent holder that is not willing to provide a license to such patent claims in a manner consistent with the IPR Mode of the OASIS Technical Committee that produced this specification. OASIS may include such claims on its website, but disclaims any obligation to do so.

OASIS takes no position regarding the validity or scope of any intellectual property or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; neither does it represent that it has made any effort to identify any such rights. Information on OASIS' procedures with respect to rights in any document or deliverable produced by an OASIS Technical Committee can be found on the OASIS website. Copies of claims of rights made available for publication and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this OASIS Committee Specification or OASIS Standard, can be obtained from the OASIS TC Administrator. OASIS makes no representation that any information or list of intellectual property rights will at any time be complete, or that any claims in such list are, in fact, Essential Claims.

The name "OASIS" is a trademark of OASIS, the owner and developer of this specification, and should be used only to refer to the organization and its official outputs. OASIS welcomes reference to, and implementation and use of, specifications, while reserving the right to enforce its marks against misleading uses. Please see http://www.oasis-open.org/who/trademark.php for above guidance.

Table of Contents

Appendixes

1. Introduction

Since its approval as a W3C recommendation in 1998, XML has been adopted in a number of industries as a framework for the definition of the messages exchanged in electronic commerce. The widespread use of XML has led to the development of multiple industry-specific XML versions of such basic documents as purchase orders, shipping notices, and invoices.

While industry-specific data formats have the advantage of maximal optimization for their business context, the existence of different formats to accomplish the same purpose in different business domains is attended by a number of significant disadvantages as well.

- Developing and maintaining multiple versions of common business documents like purchase orders and invoices is a major duplication of effort.
- Creating and maintaining multiple adapters to enable trading relationships across domain boundaries is an even greater effort.
- The existence of multiple XML formats makes it much harder to integrate XML business messages with back-office systems.
- The need to support an arbitrary number of XML formats makes tools more expensive and trained workers harder to find.

The OASIS Universal Business Language (UBL) is intended to help solve these problems by defining a generic XML interchange format for business documents that can be extended to meet the requirements of particular industries. Specifically, UBL provides the following:

- A library of XML schemas for reusable data components such as “Address,” “Item,” and “Payment”—the common data elements of everyday business documents.
- A set of XML schemas for common business documents such as “Order,” “Despatch Advice,” and “Invoice” that are constructed from the UBL library components and can be used in generic procurement and transportation contexts.

A standard basis for XML business schemas provides the following advantages:

- Lower cost of integration, both among and within enterprises, through the reuse of common data structures.
- Lower cost of commercial software, because software written to process a given XML tag set is much easier to develop than software that can handle an unlimited number of tag sets.
- An easier learning curve, because users need master just a single library.
- Lower cost of entry and therefore quicker adoption by small and medium-size enterprises (SMEs).
- Standardized training, resulting in many skilled workers.
- A universally available pool of system integrators.
- Standardized, inexpensive data input and output tools.
- A standard target for inexpensive off-the-shelf business software.

UBL is designed to provide a universally understood and recognized commercial syntax for legally binding business documents and to operate within a standard business framework such as ISO 15000 (ebXML) to provide a complete, standards-based infrastructure that can extend the benefits of existing EDI systems to businesses of all sizes. UBL is freely available to everyone without legal encumbrance or licensing fees.

UBL schemas are modular, reusable, and extensible in XML-aware ways. As the first standard implementation of ebXML Core Components Technical Specification 2.01, the UBL Library is based on a conceptual model of information components known as Business Information Entities (BIEs). These components are assembled into specific document models such as Order and Invoice. These document assembly models are then transformed in accordance with UBL Naming and Design Rules into W3C XSD schema syntax. This approach facilitates the creation of UBL-based document types beyond those specified in this release.

1.1. Terminology

1.1.1. Terms and Definitions

Assembly model

A tree-structured model of ABIEs that can be implemented as a document schema. In this release, assembly models are provided in tabular form as spreadsheets.

Document

A set of information components that are interchanged as part of a business transaction; for example, in placing an order.

XSD schema

An XML document definition conforming to the W3C XML Schema language [XSD1] [XSD2].

The terms *Core Component (CC), Basic Core Component (BCC), Aggregate Core Component (ACC), Association Core Component (ASCC), Business Information Entity (BIE), Basic Business Information Entity (BBIE),* and *Aggregate Business Information Entity (ABIE)* are used in this specification with the meanings given in [CCTS].

The terms *Object Class, Property Term, Representation Term,* and *Qualifier* are used in this specification with the meanings given in [ISO11179].

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 [RFC2119].

1.1.2. Symbols and Abbreviations

ABIE

Aggregate Business Information Entity

ASBIE

Association Business Information Entity

BBIE

Basic Business Information Entity

BIE

Business Information Entity

CC

Core Component

CPFR

Collaborative Planning, Forecasting, and Replenishment [CPFR]

CV2

Credit Card Verification Numbering System

EDI
Electronic Data Interchange

ISO
International Organization for Standardization

NDR
UBL Naming and Design Rules (see Appendix G, *UBL 2.1 Naming and Design Rules (Informative)*)

UML
Unified Modeling Language [UML]

UN/CEFACT
United Nations Centre for Trade Facilitation and Electronic Business

UNDG
United Nations Dangerous Goods

URI
Uniform Resource Identifier

UUID
Universally Unique Identifier

XML
Extensible Markup Language [XML]

XPath
The XML Path Language

XSD
W3C XML Schema Language [XSD1] [XSD2]

1.2. Normative References

[ASN.1] *ITU-T X.680-X.683: Abstract Syntax Notation One (ASN.1) [http://www.itu.int/ITU-T/studygroups/com17/languages/X.680-X.693-0207w.zip]* , *ITU-T X.690-X.693: ASN.1 encoding rules [http://www.oasis-open.org/committees/download.php/6320/X.680-X.693-0207w.zip]*

[CCTS] *ISO/TS 15000-5:2005 Electronic Business Extensible Markup Language (ebXML)—Part 5: ebXML Core Components Technical Specification, Version 2.01 [http://www.oasis-open.org/committees/download.php/6232/CEFACT-CCTS-Version-2pt01.zip] (identical to Part 8 of the ebXML Framework)*

[CPFR] *Voluntary Interindustry Commerce Standards, Collaborative Planning, Forecasting and Replenishment Version 2.0, Global Commerce Initiative Recommended Guidelines, June 2002 [http://www.vics.org/docs/committees/cpfr/CPFR_Tabs_061802.pdf]*

[Customization] *OASIS Committee Specification 01, UBL 2 Guidelines for Customization, First Edition, 25 December 2009 [http://docs.oasis-open.org/ubl/guidelines/UBL2-Customization1.0cs01.pdf]*

[CVA] *OASIS Committee Specification 01, Context/value association using genericode 1.0, 15 April 2010 [http://docs.oasis-open.org/codelist/cs01-ContextValueAssociation-1.0/doc/context-value-association.html]*

[genericode] *OASIS Committee Specification 01, Code List Representation (Genericode) Version 1.0, 28 December 2007 [http://docs.oasis-open.org/codelist/cs-genericode-1.0/doc/oasis-code-list-representation-genericode.pdf]*

[ISO11179] *ISO/IEC 11179-1:1999 Information technology — Specification and standardization of data elements — Part 1: Framework for the specification and standardization of data elements [http://www.oasis-open.org/committees/download.php/6233/c002349_ISO_IEC_11179-1_1999%28E%29.pdf]*

[RELAX NG] *ISO/IEC 19757-2, Information technology — Document Schema Definition Language (DSDL) — Part 2: Regular-grammar-based validation — RELAX NG [http://standards.iso.org/ittf/PubliclyAvailableStandards/c037605_ISO_IEC_19757-2_2003(E).zip]* , *Information technology — Document Schema Definition Language (DSDL) — Part 2: Regular-grammar-based validation — RELAX NG AMENDMENT 1: Compact Syntax [http://standards.iso.org/ittf/PubliclyAvailableStandards/c040774_ISO_IEC_19757-2_2003_Amd_1_2006(E).zip]*

[RFC2119] *Key words for use in RFCs to Indicate Requirement Levels [http://www.faqs.org/rfcs/rfc2119.html]*

[SCH] *Document Schema Definition Languages (DSDL) — Part 3: Rule-based validation (Schematron) [http://standards.iso.org/ittf/PubliclyAvailableStandards/c040833_ISO_IEC_19757-3_2006(E).zip]*

[UML] *Unified Modeling Language Version 1.5 (formal/03-03-01) [http://www.oasis-open.org/committees/download.php/6240/03-03-01.zip]*

[XAdES] *XML Advanced Electronic Signatures. ETSI TS 101 903 V1.2.2 (2004-04) [http://uri.etsi.org/01903/v1.2.2/ts_101903v010202p.pdf]*

[XML] *Extensible Markup Language (XML) 1.0 (Second Edition), W3C Recommendation 6 October 2000 [http://www.w3.org/TR/2000/REC-xml-20001006]*

[xmldsig] *XML-Signature Syntax and Processing. W3C Recommendation 12 February 2002 [http://www.w3.org/TR/2002/REC-xmldsig-core-20020212/]*

[XSD1] *XML Schema Part 1: Structures. Second Edition. W3C Recommendation 28 October 2004 [http://www.w3.org/TR/2004/REC-xmlschema-1-20041028/]*

[XSD2] *XML Schema Part 2: Datatypes. Second Edition. W3C Recommendation 28 October 2004 [http://www.w3.org/TR/2004/REC-xmlschema-2-20041028/]*

[XSLT] *XSL Transformations (XSLT) Version 1.0, W3C Recommendation 16 November 1999 [http://www.w3.org/TR/1999/REC-xslt-19991116]*

1.3. Non-Normative References

[CPFRoverview] *CPFR: An Overview, 18 May 2004 [http://www.vics.org/docs/standards/CPFR_Overview_US-A4.pdf]*

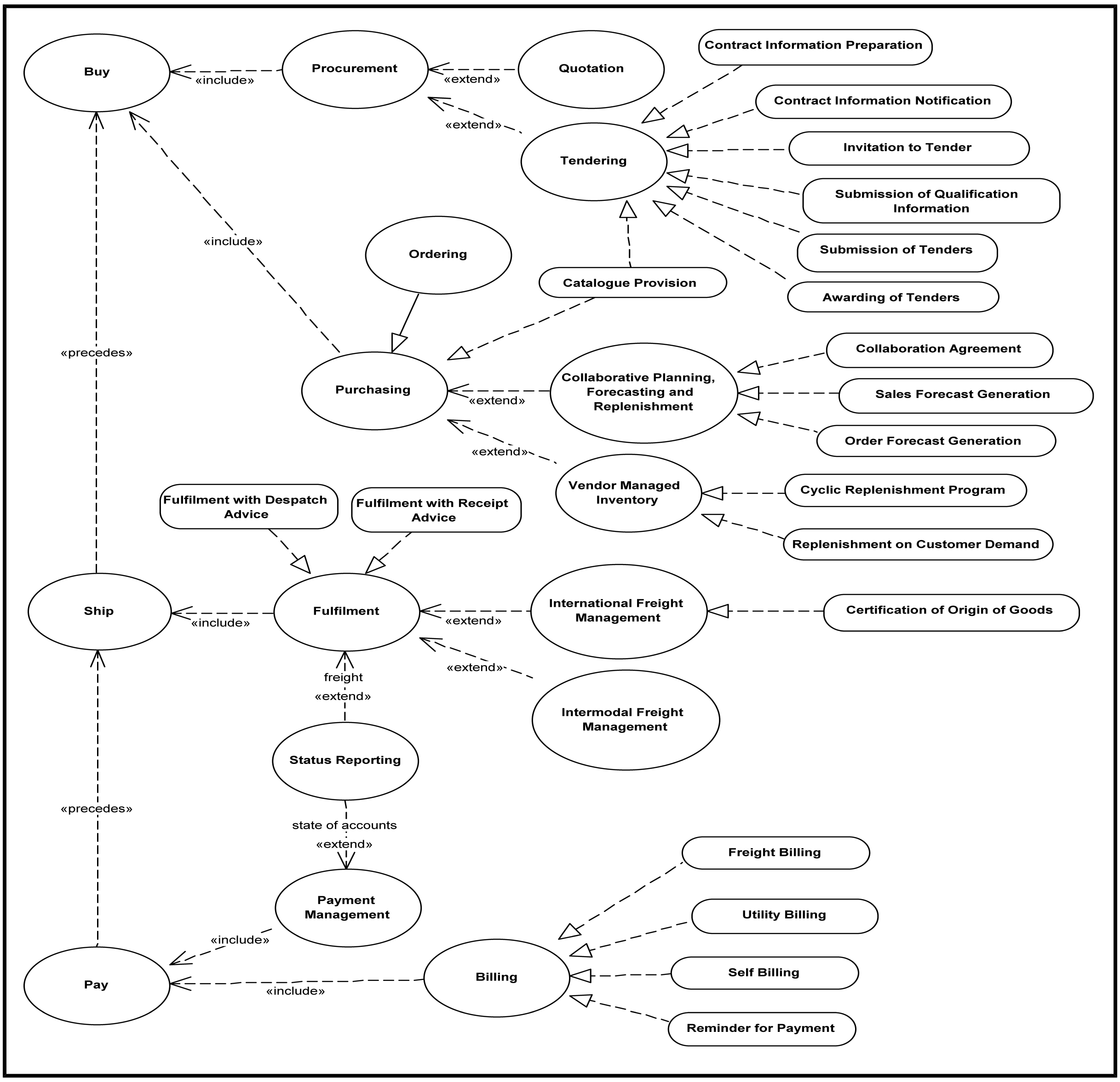
[eBiz-TCF] *Reference Architecture of eBusiness in Textile Clothing and Footwear Sector [http://spring.bologna.enea.it/ebiz/defaultebiz.asp?versione=DOWNSTREAM]*

2. UBL 2.1 Context of Use

The processes described in this section, and the business rules associated with them, define a context for the use of UBL 2.1 business documents. They are normative insofar as they provide semantics for the UBL document schemas, but they should not be construed as limiting the application of those schemas.

UBL 2.1 extends the supply chain (including the commercial collaborations of international trade) to include documents for collaborative forecasting, planning and scheduling; vendor managed inventory; utility billing; tendering; and intermodal freight management. The following diagram illustrates the process context assumed by UBL 2.1 documents.

Figure 1. UBL 2.1 Use Case



It is important to note that, as with previous UBL releases, the UBL 2.1 library is designed to support the construction of a wide variety of document types beyond those provided in the 2.1 package. It is expected that implementers will develop their own customized document types and components and that more UBL document types will be added as the library evolves. For guidance in customizing UBL document types, see the UBL Guidelines for Customization [Customization].

2.1. General Business Requirements

This section describes some of the requirements and general business rules that are assumed for collaborations and document exchanges in UBL 2.1.

2.1.1. Items

- An item may be a product or a service
- Items may have multiple classifications
- A contract may influence prices
- An item may be part of another item
- An item may have a price per unit and an order unit
- An item may reference pictures and documents
- An item may have a validity period
- An item may refer to other relevant or necessary items

2.1.2. Item Identification

One of the following identifiers may be used to identify each Item (for example, a product):

- Buyer’s Item Identification, or
- Seller’s Item Identification, or
- Manufacturer’s Item Identification, or
- Catalogue Item Identification, or
- Item Identification according to a system promulgated by a standards body.

The Item may be further distinguished by the specification of Measurement(s) or Physical Attribute(s). This enables specification of the following kinds of item:

- Item Requiring Description

  This is an item that is not identified by an unambiguous machine-processable product code and requires additional descriptive information to precisely identify it.

- Customer Defined Item

  This is an item that the customer describes according to his need, and in the specification of which the customer may make some reference to comparable “standard” items.

- Item Requiring Measurements

  This is an item for which it is necessary to specify one or more measurements as part of the descriptive specification of the item.

2.1.3. Item Instances

Certain Items may be identified and ordered as individual, unique objects—for example, a specific car rather than a make and model of a car. This form of identification may also be needed for product tracing (e.g., perishable goods) or because of the nature of the commodity (e.g., used, collectible, specialized, or rare).

In data modeling terms, an Item Instance is an extension of an Item.

2.1.4. Item Pricing

For any given Item, price ranges by amount, quantity, location, etc., are specified by the Seller during the sourcing stage. They are not repeated back to the Seller during Ordering; only the active price is specified.

In some cases, the Buyer may not know the Item Price, in which case it is not specified. This makes a detailed response from the Seller necessary; see Order Response.

2.1.5. Hazardous Items

Although ordered items may include Hazardous items, it is not necessary to specify information related to Hazardous status at the order stage. The Buyer may not be aware of the nature of the Item. Indication of the Hazardous nature of the Item, and any relevant information, would be indicated in the Despatch Advice and Transportation documents.

2.1.6. Parties

In UBL, a party is defined as an individual, a group, or a body having a role in a business function.

Dependent on the business process, a Party may play various roles in the document exchange.

2.1.7. Multilingual Text

Some textual components, such as Notes and Description, may be specified in several languages. Each should be a separate occurrence of the component, using the language attribute to define its presentation. However, multiple occurrences of the same textual components should not be in the same language.

2.2. Overview of Business Processes

Following from UBL 2.0, the UBL 2.1 documents and library support an increased range of different business processes. These processes (with the additions in 2.1 shown in underlined boldface) can be categorized as follows:

- Procurement
  - Sourcing
    - Pre-Award
      - **Tendering**
      - Catalogue
    - Post-Award
      - **Catalogue**
      - **Quotation**
  - Ordering
  - Fulfilment
  - Billing
    - **Freight Billing**
    - **Utility Billing**
  - **Payment**
  - Replenishment
    - **Collaborative Planning, Forecasting and Replenishment**
    - **Vendor Managed Inventory**
      - **Cyclic Replenishment Program**

Transportation

International Freight Management

Intermodal Freight Management

Freight Status Reporting

International Trade

Certification of Origin of Goods

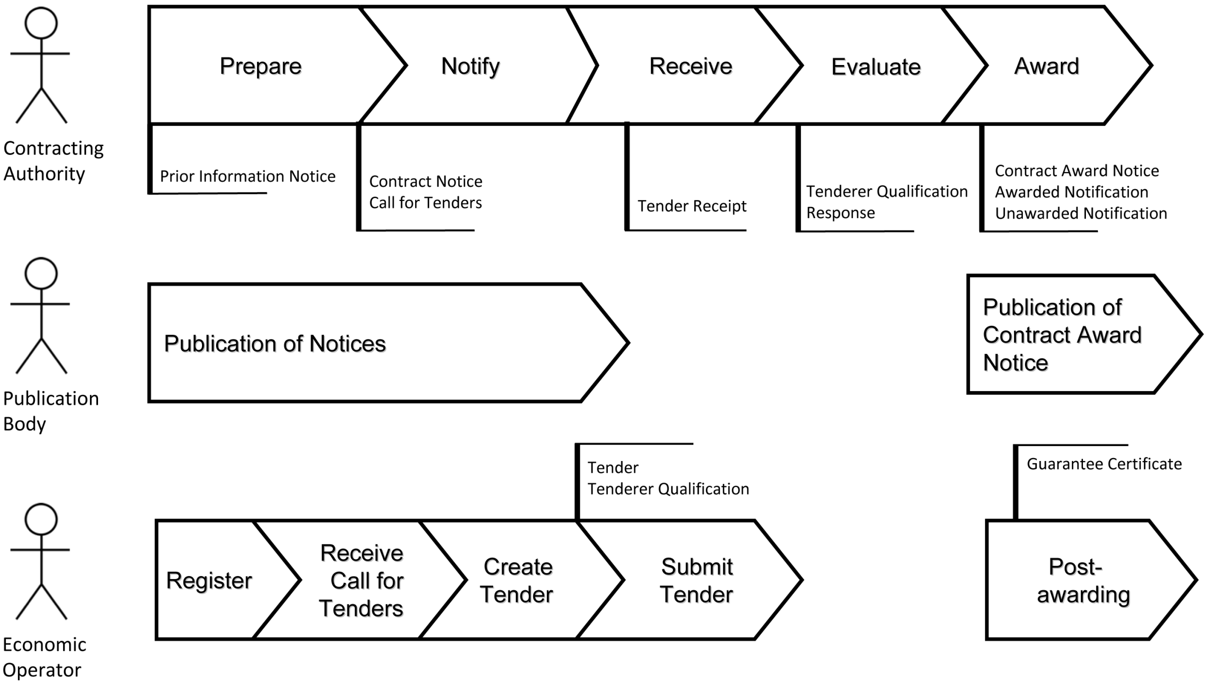
The following sections contain the formal business process descriptions:

Section 2.3, “Tendering”
Section 2.4, “Catalogue”
Section 2.5, “Quotation”
Section 2.6, “Ordering”
Section 2.7, “Fulfilment”
Section 2.8, “Billing”
Section 2.9, “Freight Billing”
Section 2.10, “Utility Billing”
Section 2.11, “Payment”
Section 2.12, “Collaborative Planning, Forecasting, and Replenishment”
Section 2.13, “Vendor Managed Inventory”
Section 2.14, “International Freight Management”
Section 2.15, “Intermodal Freight Management”
Section 2.16, “Freight Status Reporting”
Section 2.17, “Certification of Origin of Goods”

2.3. Tendering

Tendering is the case where a contracting authority (the Originator) initiates a procurement project to buy goods, services, or works during a specified period, as shown in the following diagram.

Figure 2. The Tendering Process

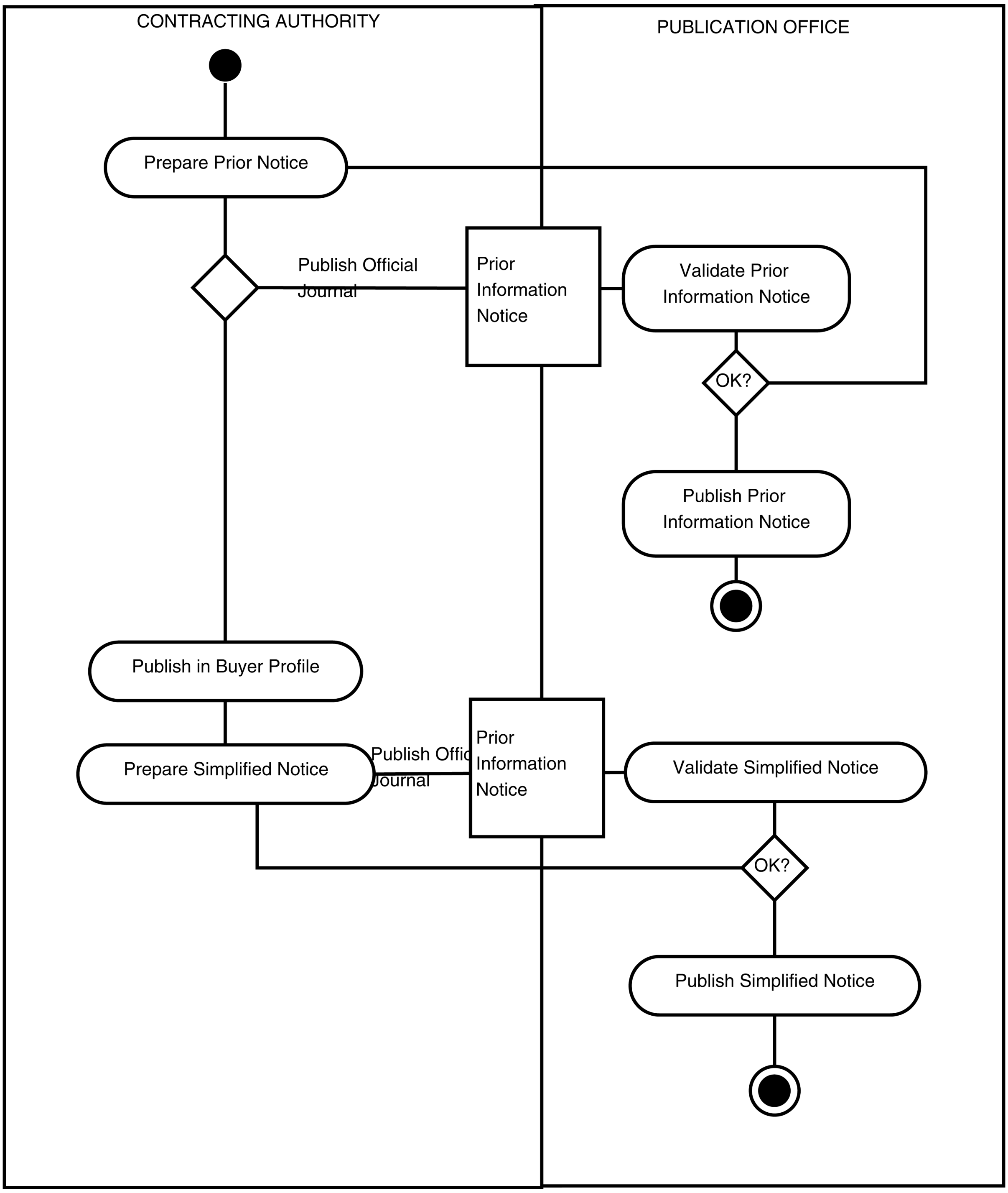


2.3.1. Contract Information Preparation

The Tendering process optionally begins with publication of a Prior Information Notice prepared by a Contracting Authority to *declare the intention* to buy goods, services, or works during a specified period.

The purpose of this step (if implemented) is to reduce preparation time when an actual Contract Notice is published (see Section 2.3.2, “Contract Information Notification”).

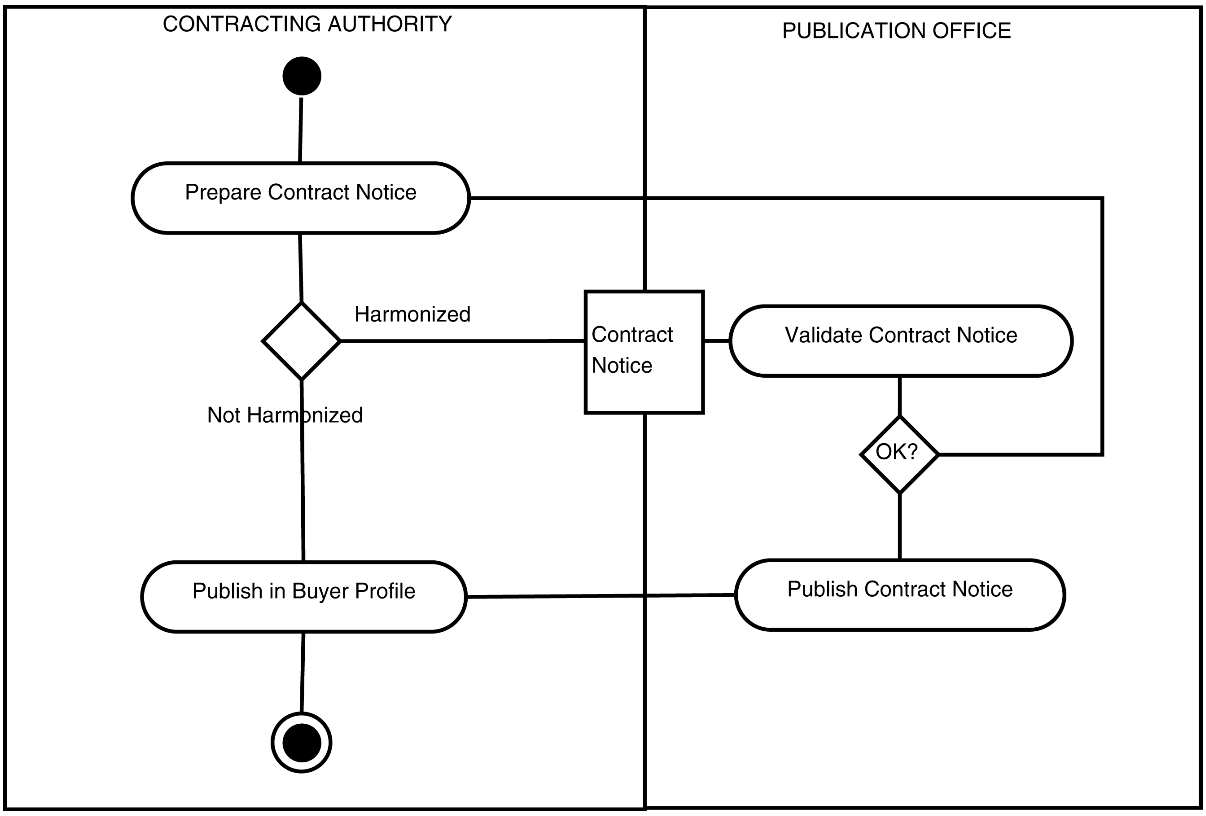
Figure 3. Contract Information Preparation



2.3.2. Contract Information Notification

The process of Notification includes the publication by the Contracting Authority of a Contract Notice to *announce* the project to buy goods, services, or works. The details shown here are specific to the EU, which requires contracts over a certain amount (Harmonized contracts) to be published in the Official Journal of the EU. Other tendering contexts will differ in their publication requirements.

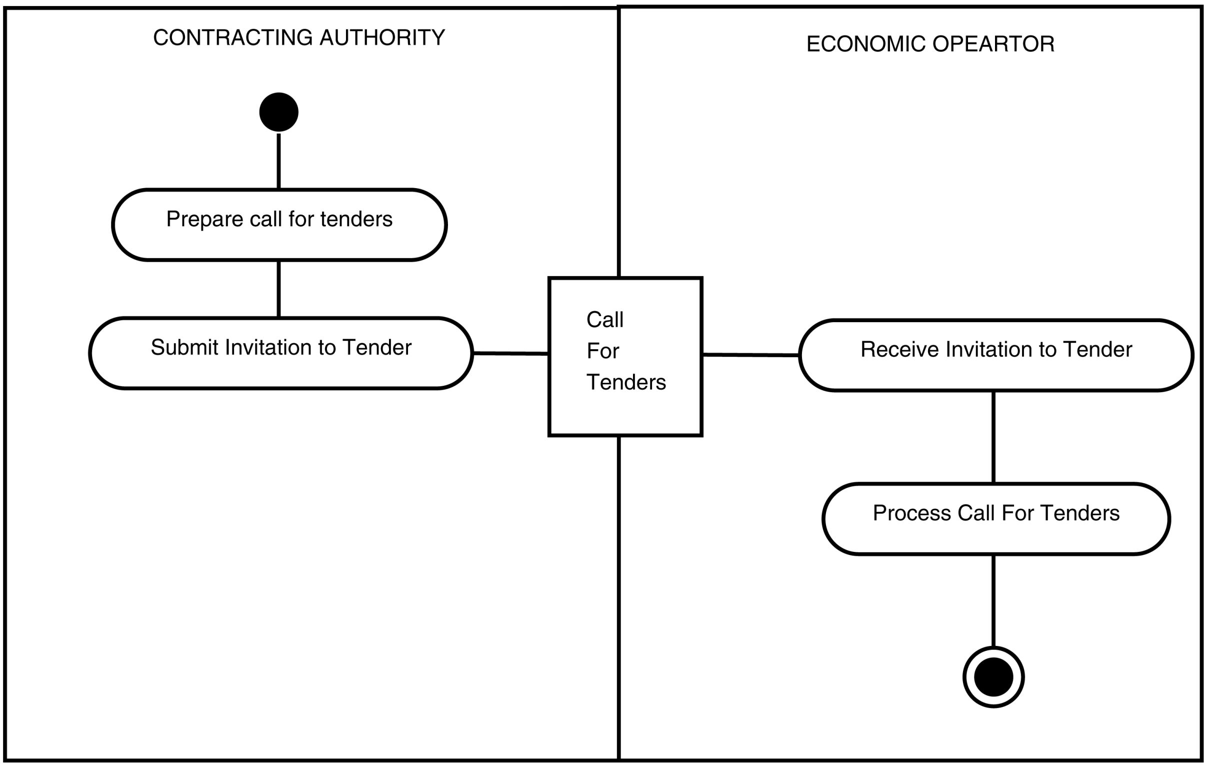
Figure 4. Contract Information Notification



2.3.3. Invitation to Tender

In some procedures, the Contracting Authority invites economic operators to participate in a contest by sending them an invitation to tender using a Call for Tenders to *define* the procurement project to buy goods, services, or works during a specified period. The Call For Tenders may be sent jointly with an unstructured letter of invitation to tender.

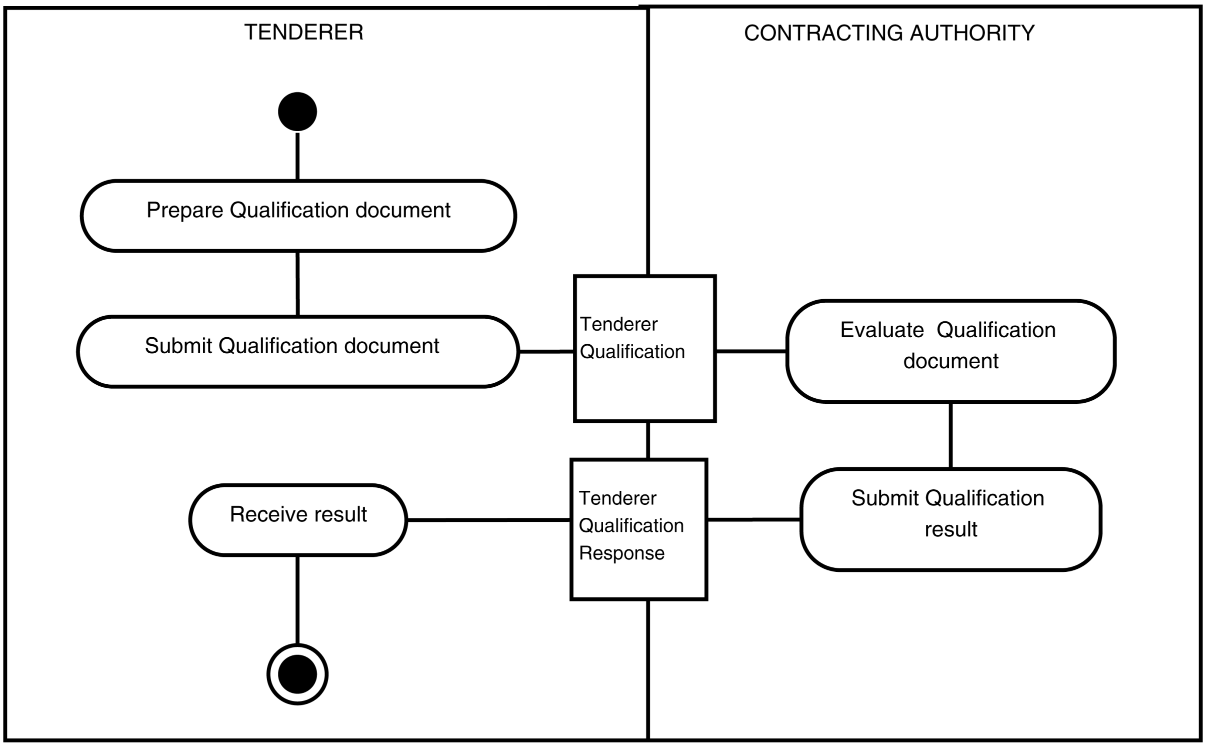
Figure 5. Invitation to Tender



2.3.4. Submission of Qualification Information

The economic operator sends a Tenderer Qualification to the Contracting Authority to *define its own situation or status* relating to the requirements of the Contracting Authority for a specific tendering process. The Contracting Authority uses the Tenderer Qualification Response to notify the Tenderer of its *admission to or exclusion from the tendering process.*

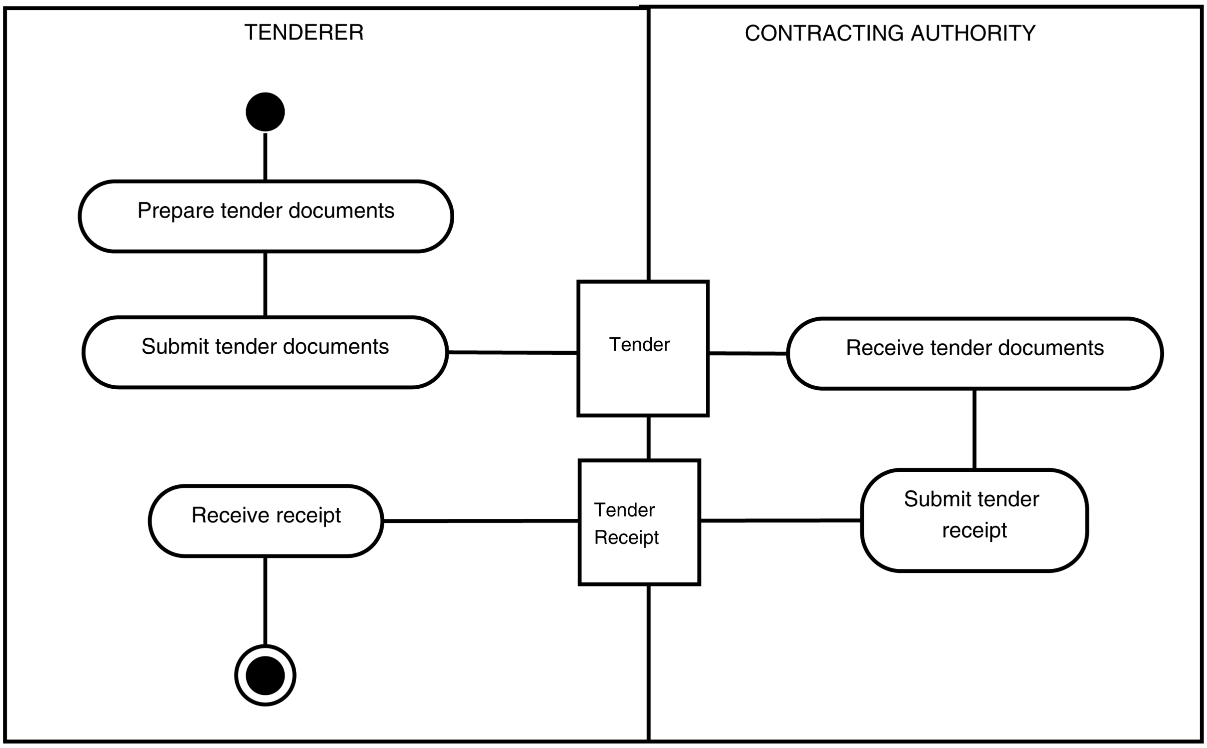
Figure 6. Submission of Qualification Information


TENDERER
CONTRACTING AUTHORITY
Prepare Qualification document
Submit Qualification document
Tenderer Qualification
Evaluate Qualification document
Receive result
Tenderer Qualification Response
Submit Qualification result

2.3.5. Submission of Tenders

A Tenderer submits one or more Tender documents that offer a tender to the Contracting Authority for bid. The Contracting Authority responds with a Tender Receipt to *notify the reception of the tender* for a tendering process. The date and time of the Tender Receipt are significant, because tendering procedures usually have tight deadlines for tender presentation.

Figure 7. Submission of Tenders

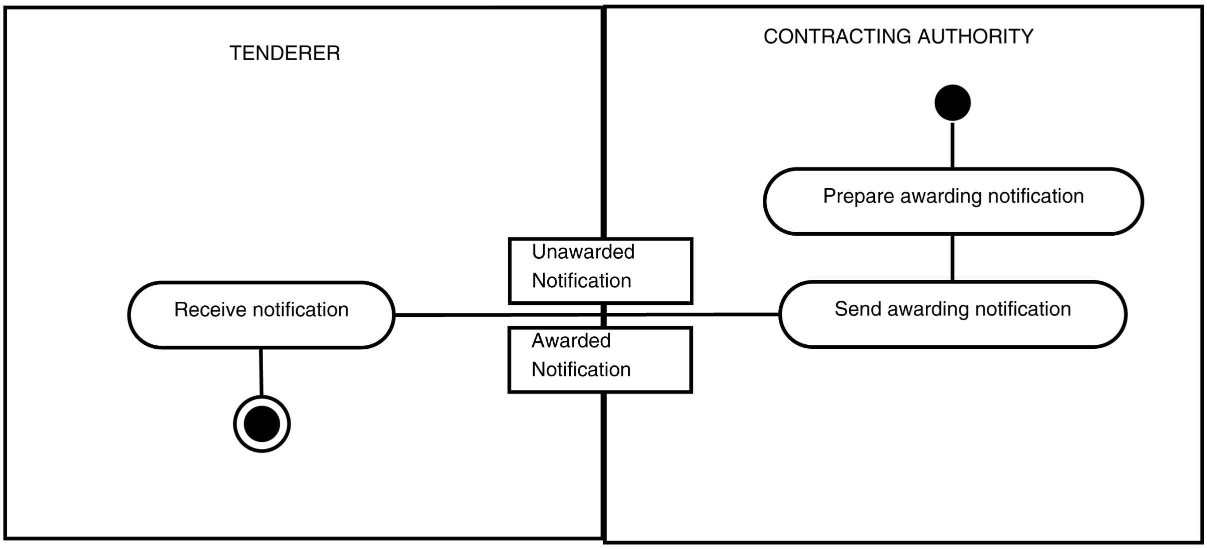


2.3.6. Awarding of Tenders

The awarding of tenders takes place in three phases.

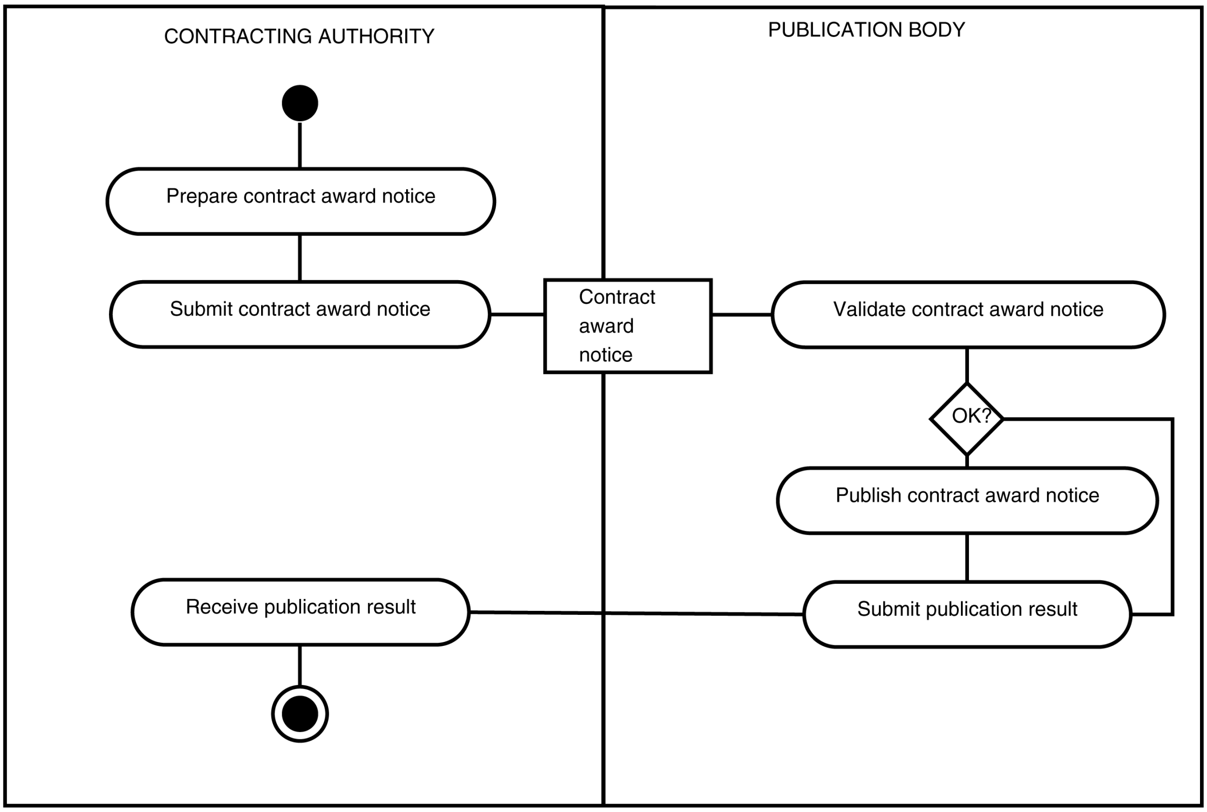
First, the Contracting Authority *notifies each tenderer of its success or failure* in winning the contract, using the Awarded Notification document to communicate the contract award to the winning tenderer or the Unawarded Notification document to communicate that the contract has been awarded to another tenderer.

Figure 8. Award Notification



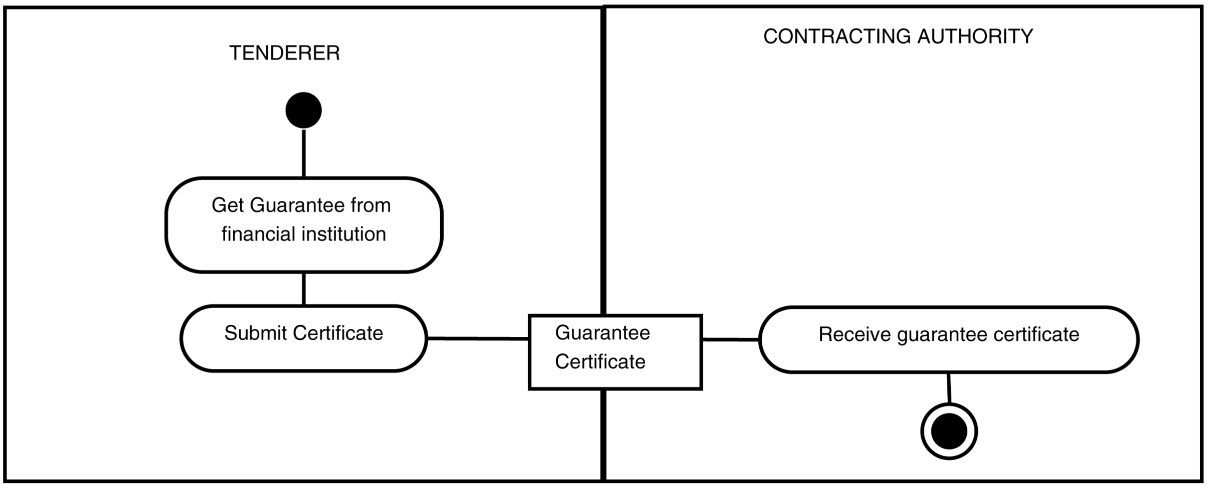
Second, the Contracting Authority causes a Contract Award Notice to be published to *announce the awarding* of a procurement project.

Figure 9. Award Publication


CONTRACTING AUTHORITY
PUBLICATION BODY
Prepare contract award notice
Submit contract award notice
Contract award notice
Validate contract award notice
OK?
Publish contract award notice
Receive publication result
Submit publication result

Finally, the Tenderer sends a Guarantee Certificate to *notify the deposit of a guarantee.*

Figure 10. Guarantee Deposit


TENDERER
CONTRACTING AUTHORITY
Get Guarantee from financial institution
Submit Certificate
Guarantee Certificate
Receive guarantee certificate

2.4. Catalogue

Document types in these processes are Catalogue Request, Application Response, Catalogue Item Specification

Update, Catalogue Pricing Update, and Catalogue Deletion.

2.4.1. Catalogue Business Rules

- Any conditions specified in the contract shall overrule those stated in the common Catalogue.

- A Catalogue exchange shall be between one Provider and one Receiver Party.
- A classification system may have its own set of properties.
- A classification scheme shall have metadata.
- A Catalogue may have a validity period.
- A Catalogue should include item classifications.
- Classification schemes should include standard and specific properties.
- A Catalogue may refer to the lot (sub-section) of a contract.
- A Catalogue may explicitly specify the framework contract reference.
- A Catalogue may refer to a DPS contract number.
- When a Catalogue item is updated, the item shall be replaced in the Catalogue.
- When a Catalogue item is updated, historical information about replaced or updated items must be available to reconcile with outstanding transactions.
- Prices may be updated independently of other Catalogue information.
- Catalogue distribution may be Provider or Receiver Party initiated.
- If a Receiver initiates a request for a Catalogue, they may request an entire Catalogue or only updates to either pricing or item specification details.
- Whether Receiver Party initiated or not, the decision to issue a new Catalogue or update an existing one shall be at the discretion of the Provider Party.
- If an updated Catalogue is issued, then an action code shall define the status of the items in the Catalogue.

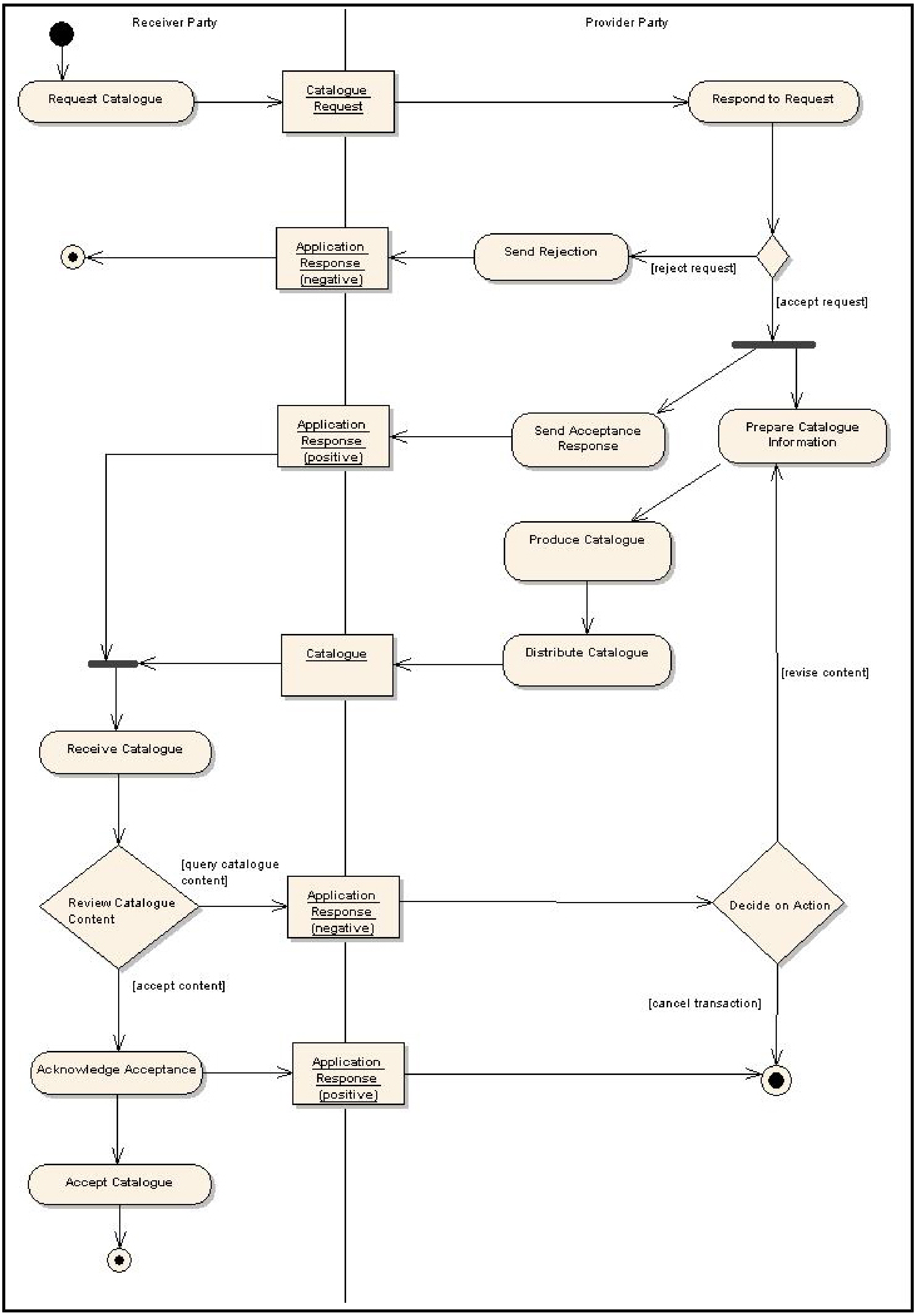
2.4.2. Catalogue Provision

A Catalogue is defined as a document produced by a party in the procurement chain that describes items and prices. Catalogue provision is the case where a Provider sends information regarding items available for purchase to a Receiver. This may be on request or unsolicited. Because they are only potential purchasers, a Receiver may never become a Customer Party.

2.4.2.1. Create Catalogue

The process of creating a Catalogue is shown in the following diagram.

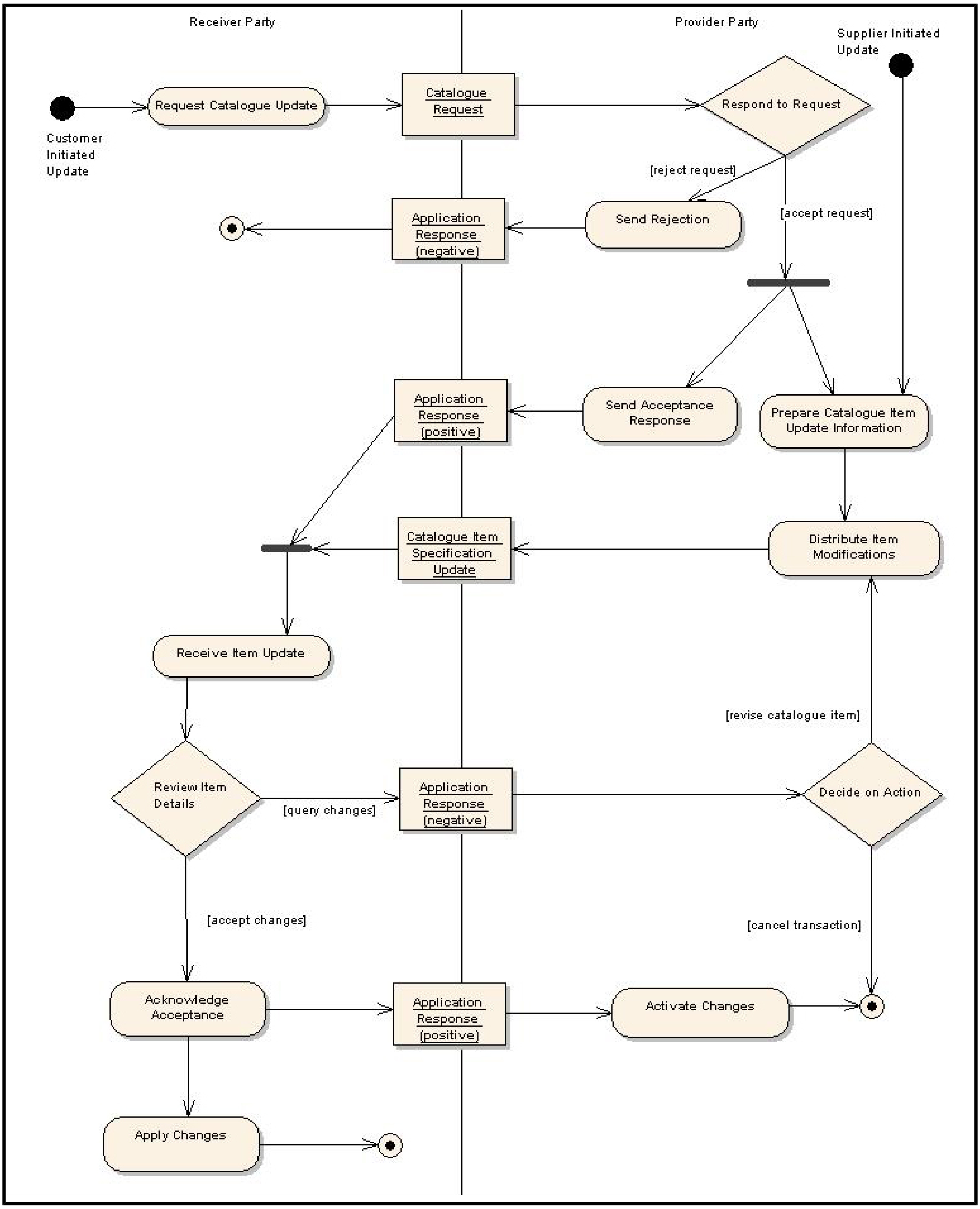
Figure 11. Create Catalogue Process



2.4.2.2. Update Catalogue Item Specification

The process of updating a Catalogue Item Specification is shown in the following diagram.

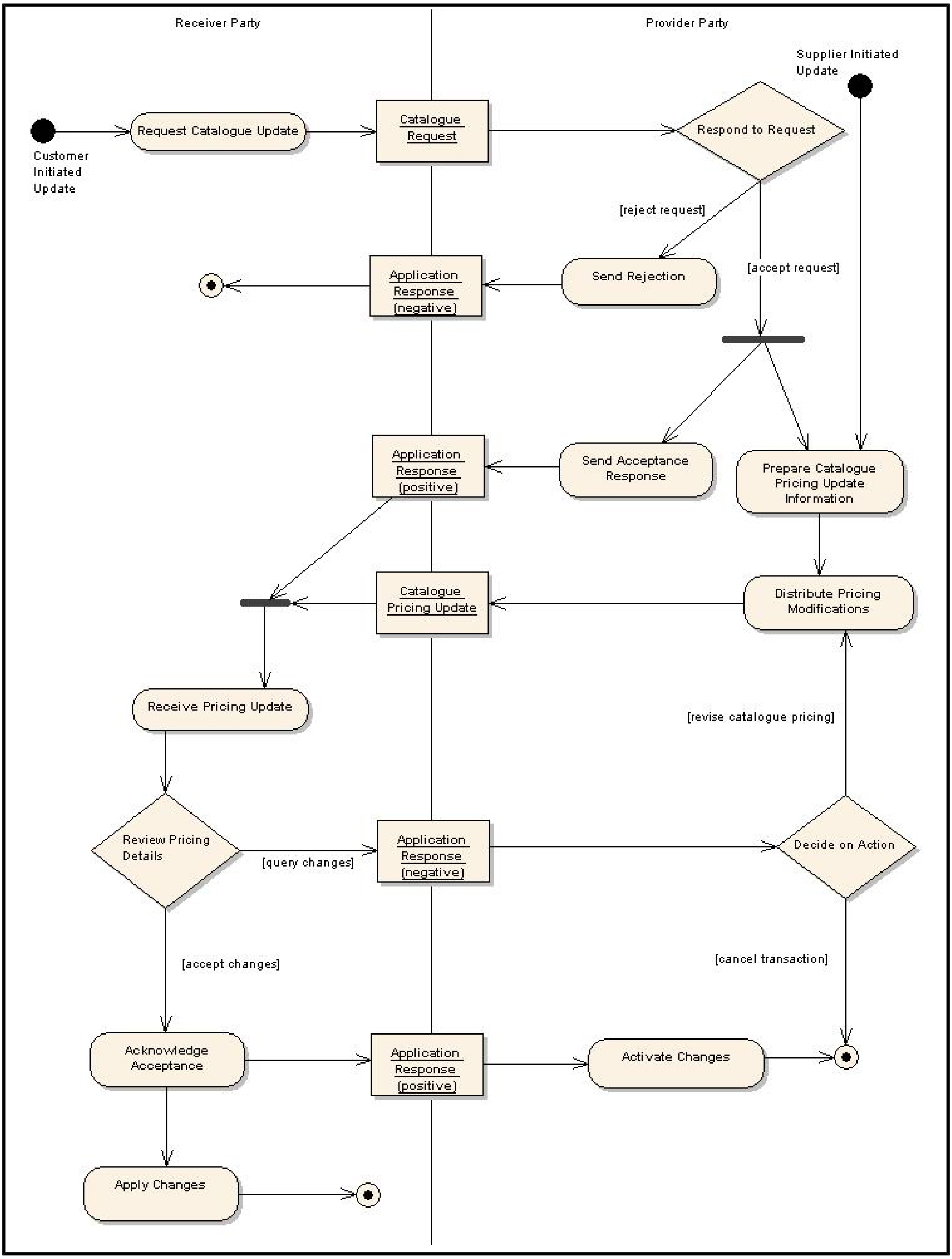
Figure 12. Update Item Specification Process



2.4.2.3. Update Catalogue Pricing

The process of updating Catalogue pricing is shown in the following diagram.

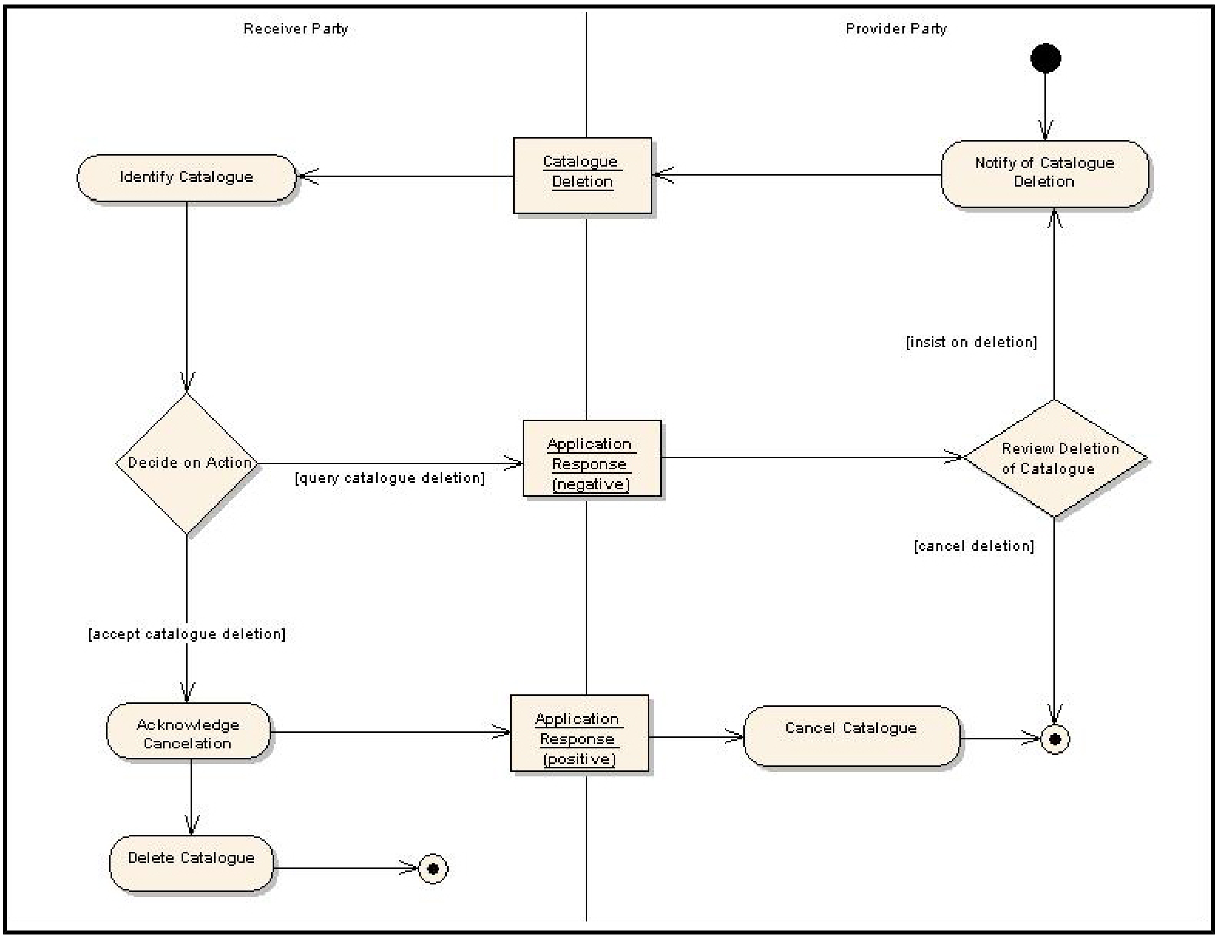
Figure 13. Update Catalogue Pricing Process



2.4.2.4. Delete Catalogue

Deletion of a Catalogue is shown in the following diagram.

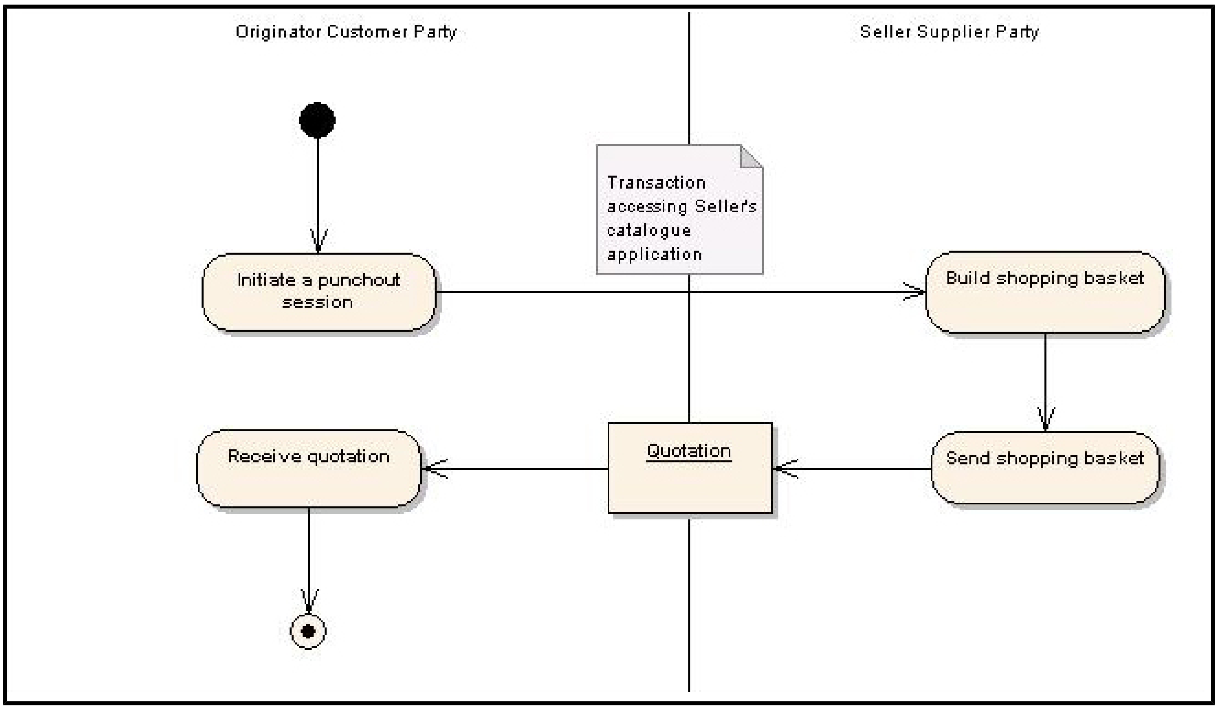
Figure 14. Delete Catalogue Process



2.4.2.5. Punchout

Punchout is a technological innovation whereby an Originator is able to directly access a Seller’s application from within their own procurement application.

Figure 15. Punchout Sourcing Process


Originator Customer Party
Seller Supplier Party
Transaction accessing Seller's catalogue application
Initiate a punchout session
Build shopping basket
Receive quotation
Quotation
Send shopping basket

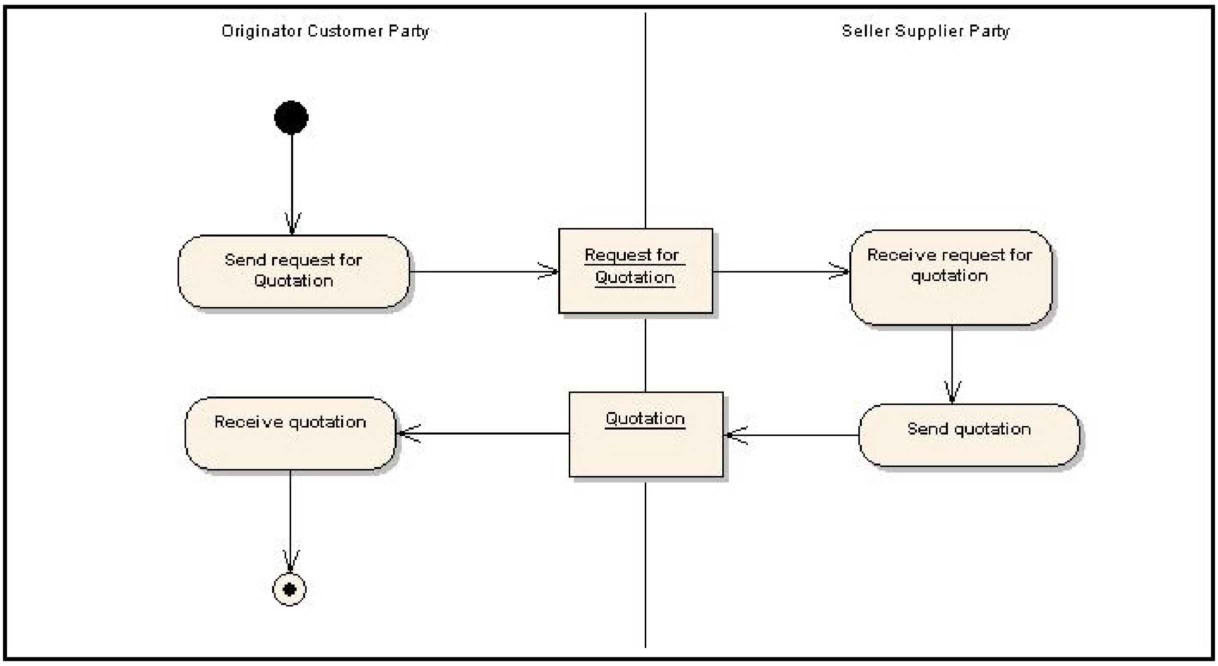
The Originators leave ("punch out" from) their system and interact with the Seller's catalogue to locate and order products, while their application transparently gathers pertinent information.

While conceptually the punchout request is a form of Request for Quote, the exchange transaction is tightly coupled to the specific catalogue application and is considered outside the scope of UBL.

2.5. Quotation

Less formally defined than a tender, a quotation process is the case where the Originator asks for a quotation, as shown in the following diagram.

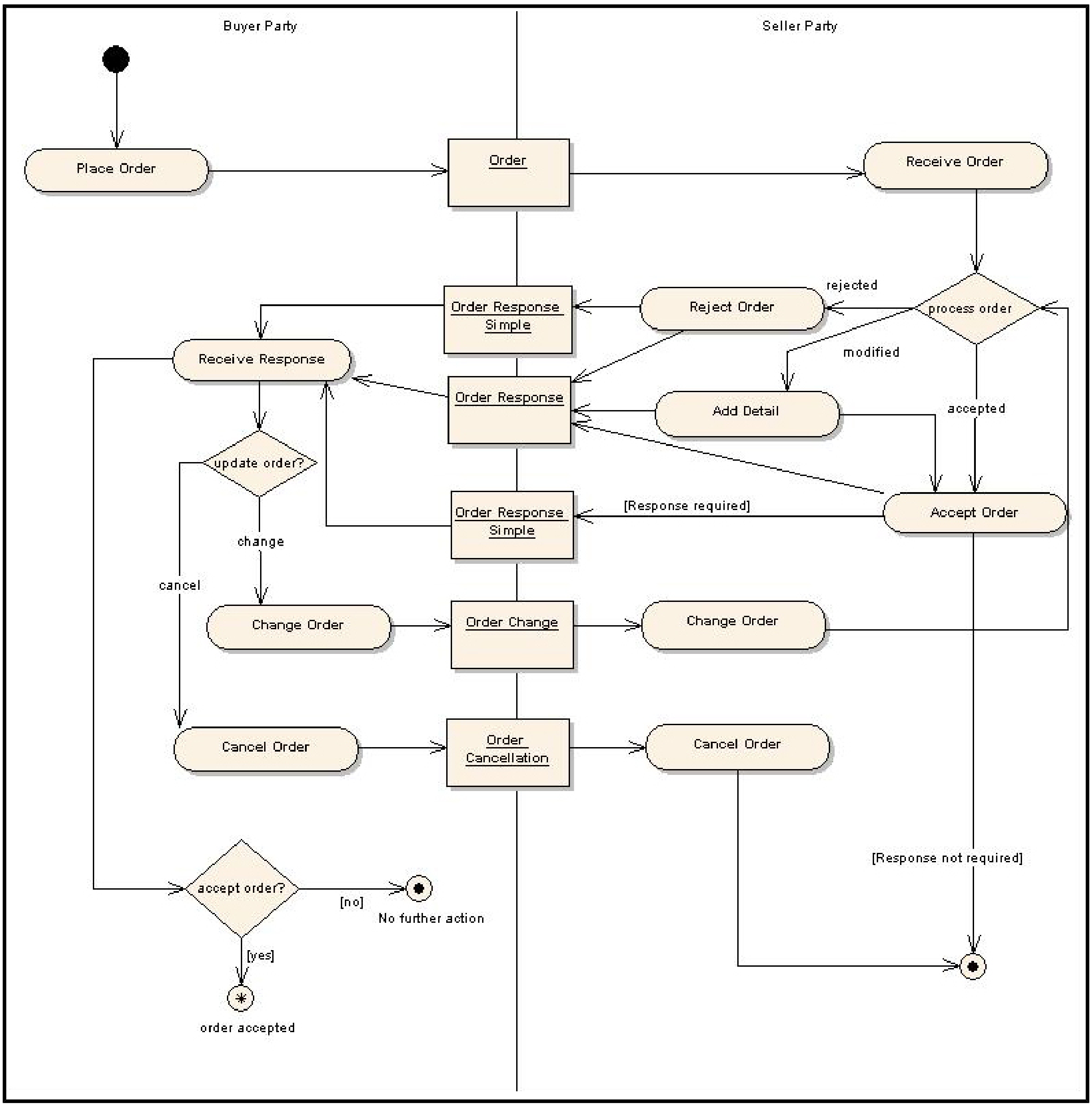
Figure 16. Quotation Process



2.6. Ordering

Ordering is the collaboration that creates a contractual obligation between the Seller Supplier Party and the Buyer Customer Party. Document types in these processes are Order, Order Response, Order Response Simple, Order Change, and Order Cancellation.

Figure 17. Ordering Process



2.6.1. Ordering Business Rules

- The Order may specify allowance and charge instructions (e.g., freight, documentation, etc.) that identify the type of charge and who pays which charges. The Order may be placed “on account” against a trading credit account held by the Seller, or against a credit/debit card account, or against a direct debit agreement. The Order allows for an overall currency defining a default for all pricing and also a specific currency to be used for Invoicing. Within an Order, additional currencies may be specified both for individual item pricing and for any allowances or charges.

- Trade discount may be specified at the Order level. The Buyer may not know the trade discount, in which case it is not specified. This makes a detailed response from the Seller necessary; see Order Response (Section 2.6.3).
- The Order provides for multiple Order Lines.
- The Order may specify delivery terms, while the Order Line may provide instructions for delivery.
- The Buyer may indicate potential acceptable alternatives.

2.6.2. Order Response Simple

The Order Response Simple is the means by which the Seller confirms receipt of the Order from the Buyer, indicating either commitment to fulfil without change or that the Order has been rejected.

2.6.3. Order Response

Proposed changes to an Order by the Seller are accomplished through the full Order Response document.

The Order Response proposes to replace the original Order. It reflects the entire new state of an order transaction.

It also is the means by which the Seller confirms or supplies Order-related details to the Buyer that were not available to, or specified by, the Buyer at the time of ordering. These may include:

- Delivery date, offered by the Seller if not specifically requested by the Buyer
- Prices
- Discounts
- Charges
- Item Classification codes

The Seller may advise on replacements, substitutes, or other necessary changes using the Order Response.

2.6.4. Order Change

The Buyer may change an established Order in two ways, subject to the legal contract or trading partner agreement: first, by sending an Order Change, or second, by sending an Order Cancellation (see Section 2.6.5, “Order Cancellation”) followed by a new, complete replacement Order.

An Order Change reflects the entire current state of an order transaction.

Buyers may initiate a change to a previously accepted order for various reasons, such as changing ordered items, quantity, delivery date, ship-to address, etc. Suppliers may accept or reject the Order Change using either Order Response or Order Response Simple.

2.6.5. Order Cancellation

At any point of the process, a Buyer may cancel an established order transaction using the Order Cancellation document. Legal contracts, trading partner agreements, and business rules will determine the point at which an Order Cancellation will be ignored (e.g., at the point of manufacture or the initiation of the delivery process). Given the agreements and rules, an Order Cancellation may or may not be an automated business transaction. The terms and conditions of contract formation for business commitments will dictate which, if any, of these restrictions or guidelines will apply.

2.7. Fulfilment

Fulfilment is the collaboration in which the goods or services are transferred from the Despatch Party to the Delivery Party.

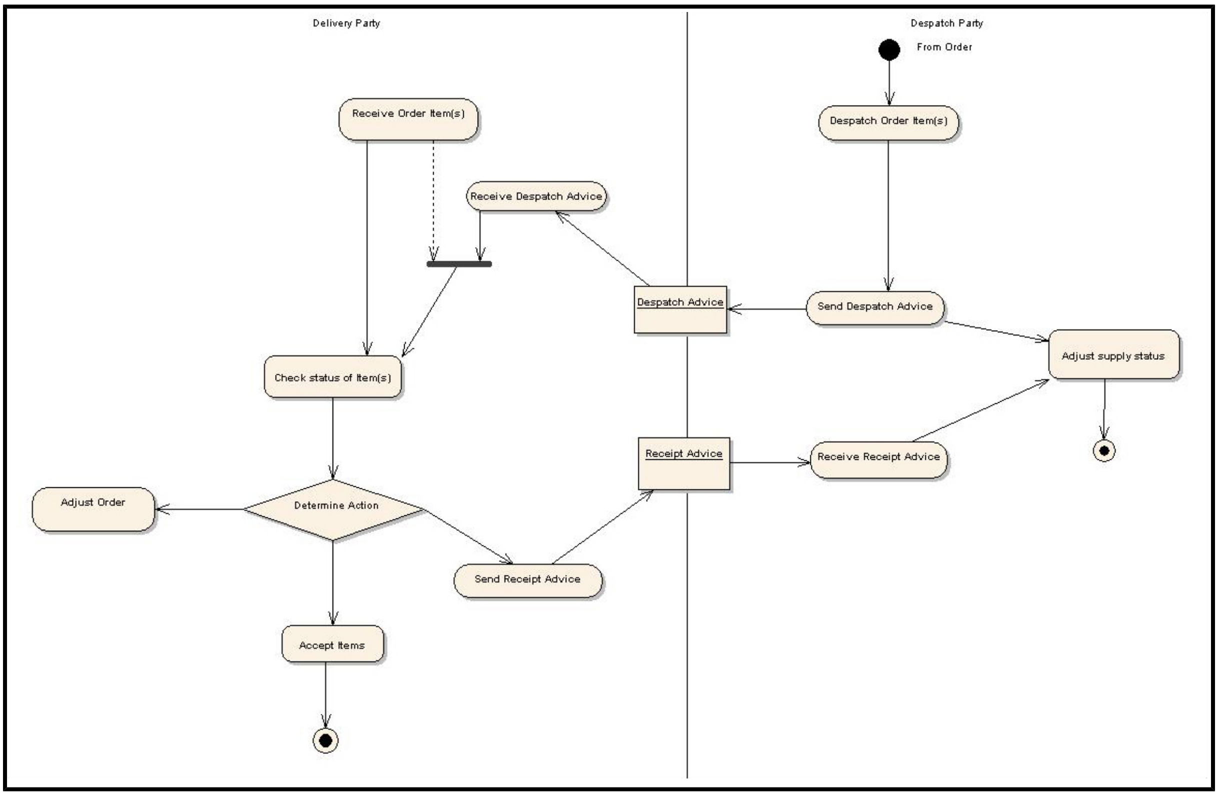
Document types in these processes are Despatch Advice, Receipt Advice, Order Cancellation, and Order Change.

In common practice, fulfilment is either supported by a proactive Despatch Advice from the Despatch Party or by a reactive Receipt Advice from the Delivery Party.

If the Customer is not satisfied with the goods or services, they may then cancel or change the order (see Section 2.6, “Ordering”).

The Seller may have a fulfilment (or customer) service dealing with anomalies.

Figure 18. Fulfilment with Despatch Advice Process



2.7.1. Despatch Advice Business Rules

The Despatch Advice is sent by the Despatch Party to the Delivery Party to confirm shipment of items.

The Despatch Advice provides for two situations:

1. Organization of the delivery set of items by Transport Handling Unit(s) so that the Receiver can check the Transport Handling Unit and then the contained items. Quantities of the same item on the same Order Line may be separated into different Transport Handling Units and hence appear on separate Despatch Lines within a Transport Handling Unit.

2. Organization of the delivery set of items by Despatch Line, annotated by the Transport Handling Unit in which they are placed, to facilitate checking against the Order. For convenience, any Order Line split over multiple Transport Handling Units will result in a Despatch Line for each Transport Handling Unit they are contained in.

Additionally, in either case, the Despatch Advice may advise:

- Full Despatch—advising the Recipient and/or Buyer that all the items on the order will be, or are being, delivered in one complete consignment on a given date.
- Partial Despatch—advising the Recipient and/or Buyer that the items on the order will be, or are being, partially delivered in a consignment on a given date.

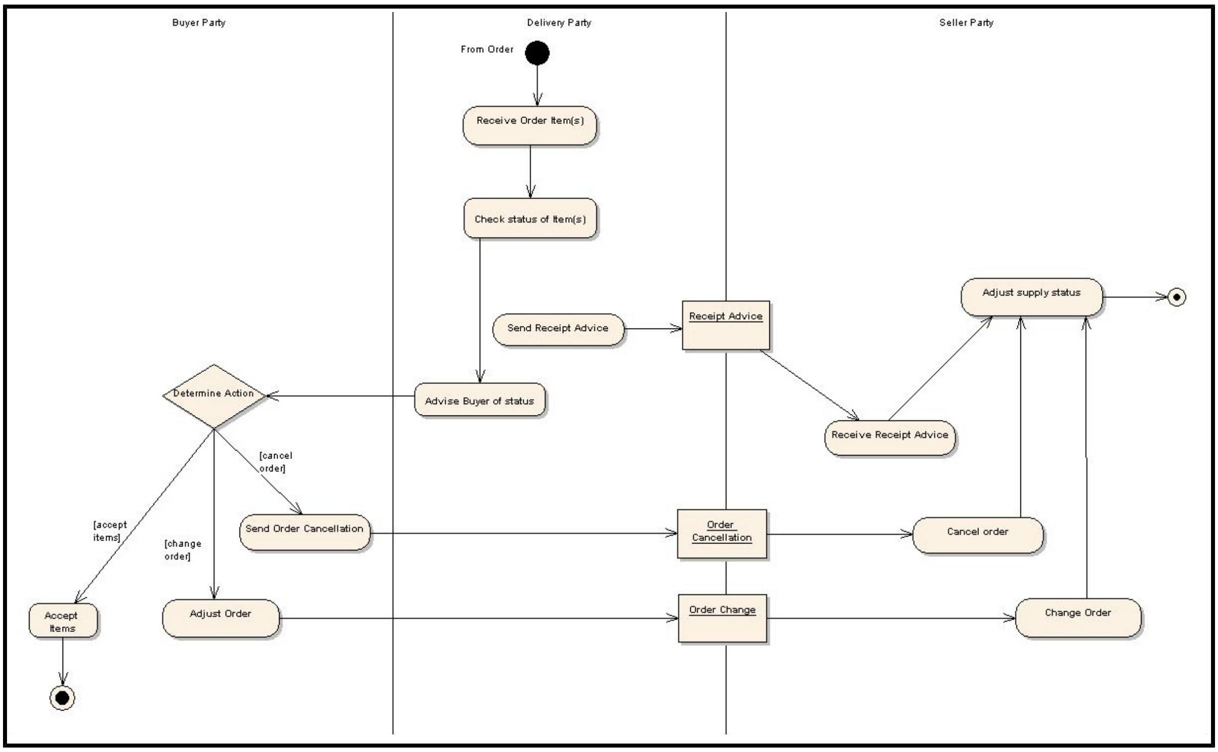
Despatch Lines of the Despatch Advice need not correspond one-to-one with Order Lines, and are linked by a reference. The information structure of the Despatch Advice may result in multiple Despatch Lines from one Order Line. Equally, partial despatch may result in some Order Lines not being matched by any Line in a Despatch Advice.

Within a Despatch Advice, an Item may also indicate the Country of Origin and the Hazardous nature of the Item.

2.7.2. Receipt Advice Business Rules

The Receipt Advice is sent by the Delivery Party to the Despatch Party to confirm receipt of items and is capable of reporting shortages or damaged items.

Figure 19. Fulfilment with Receipt Advice Process



The Receipt Advice provides for two situations. For ease of processing claimed receipt against claimed delivery, it must be organised in the same way as the corresponding Despatch Advice:

1. Indication of receipt by Transport Handling Unit(s) and contained Receipt Lines one-to-one with the Despatch Advice as detailed by the Seller party, or
2. Indication of receipt by Receipt Lines annotated by Transport Handling Unit, one-to-one with the Despatch Advice as detailed by the Seller party.

The Receipt Advice allows the Delivery Party to state any shortages from the claimed despatch quantity and to state any quantities rejected for a given reason.

2.8. Billing

In the Billing process, a request is made for payment for goods or services that have been ordered, received, or consumed. In practice, there are several ways in which goods or services may be billed.

Document types in these processes are Invoice, Credit Note, Debit Note, and Application Response.

For UBL 2.1, we assume the following billing methods:

1. Traditional Billing
    a. Using Credit Note
    b. Using Debit Note
2. Self Billing (also known as billing on receipt)
    a. Using Credit Note
    b. Using Self Billed Credit Note

2.8.1. Billing Business Rules

The Invoice is normally issued on the basis of one despatch event triggering one invoice. An Invoice may also be issued for pre-payment on a whole or partial basis. The possibilities are:

- Prepayment invoice (payment expected)
- Pro-forma invoice (pre advice, payment not expected)
- Normal Invoice, on despatch for despatched items
- Invoice after return of Receipt Advice

The Invoice only contains the information that is necessary for invoicing purposes. It does not reiterate any information already established in the Order, Order Change, Order Response, Despatch Advice, or Receipt Advice that is not necessary when invoicing. If necessary, the Invoice refers to the Order, Despatch Advice, or Receipt Advice by a Reference for those documents.

Taxation on the Invoice allows for compound taxes, the sequence of calculation being implied by the sequence of information repeated in the data stream (e.g., Energy tax, with VAT—Value Added Tax—superimposed).

Charges may be specified either as a lump sum or by percentage applied to the whole Invoice value prior to calculation of taxes. Such charges cover:

- Packaging
- Delivery/postage
- Freight
- Documentation

Each Invoice Line refers to any related Order Line(s) and may also refer to the Despatch Line and/or Receipt Line.

2.8.2. Traditional Billing

Traditional billing is where the supplier invoices the customer when the goods are delivered or the services provided. In this case, the invoice may be created at the time of despatch or when the Delivery Party acknowledges that the goods have been received (using a Receipt Advice).

When there are discrepancies between the Despatch Advice, Receipt Advice, and/or the Invoice and the goods actually received, or the goods are rejected for quality reasons, the customer may send an Application Response or a Debit Note to the supplier. The supplier may then issue a Credit Note or another Invoice as required.

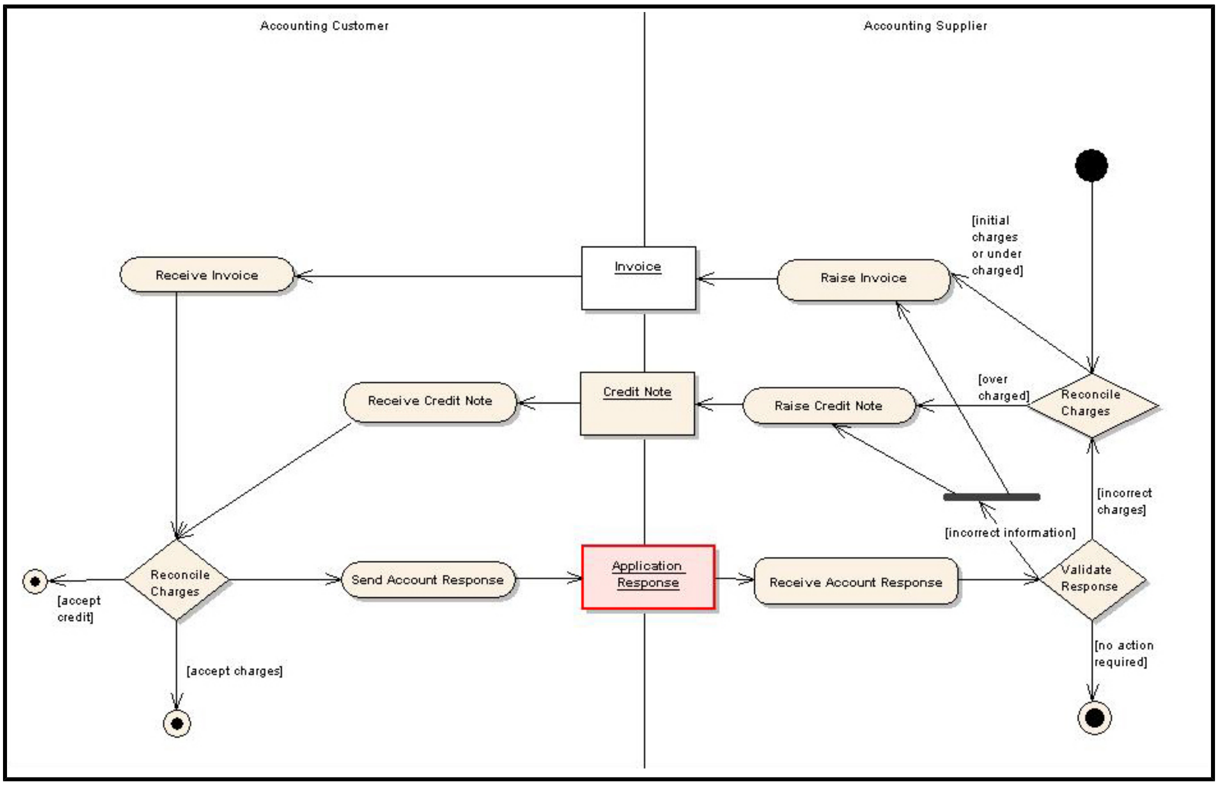
A Credit Note or Debit Note may also be issued in the case of retrospective price change.

Credit Notes or Debit Notes may be also issued after the Billing collaboration (as part of the Payment collaboration).

2.8.2.1. Billing Using Credit Notes

Billing using Credit Notes is shown in the following diagram.

Figure 20. Billing with Credit Note Process

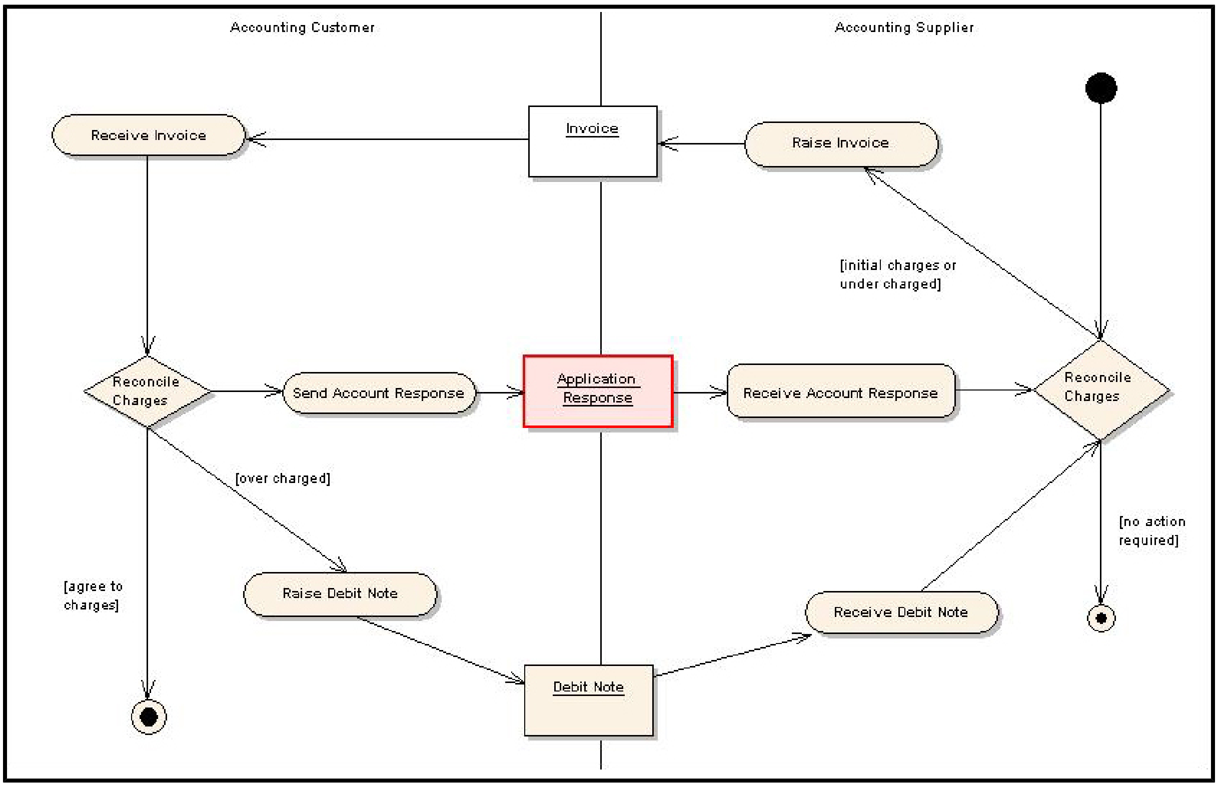


When using Credit Notes, the Supplier (in their Accounting role) is responsible for specifying the tax requirements.

2.8.2.2. Billing Using Debit Notes

Billing using Debit Notes is shown in the following diagram.

Figure 21. Billing with Debit Note Process



When using Debit Notes, both the Supplier (in their Accounting role) and the Customer (in their Accounting role) are responsible for providing taxation information.

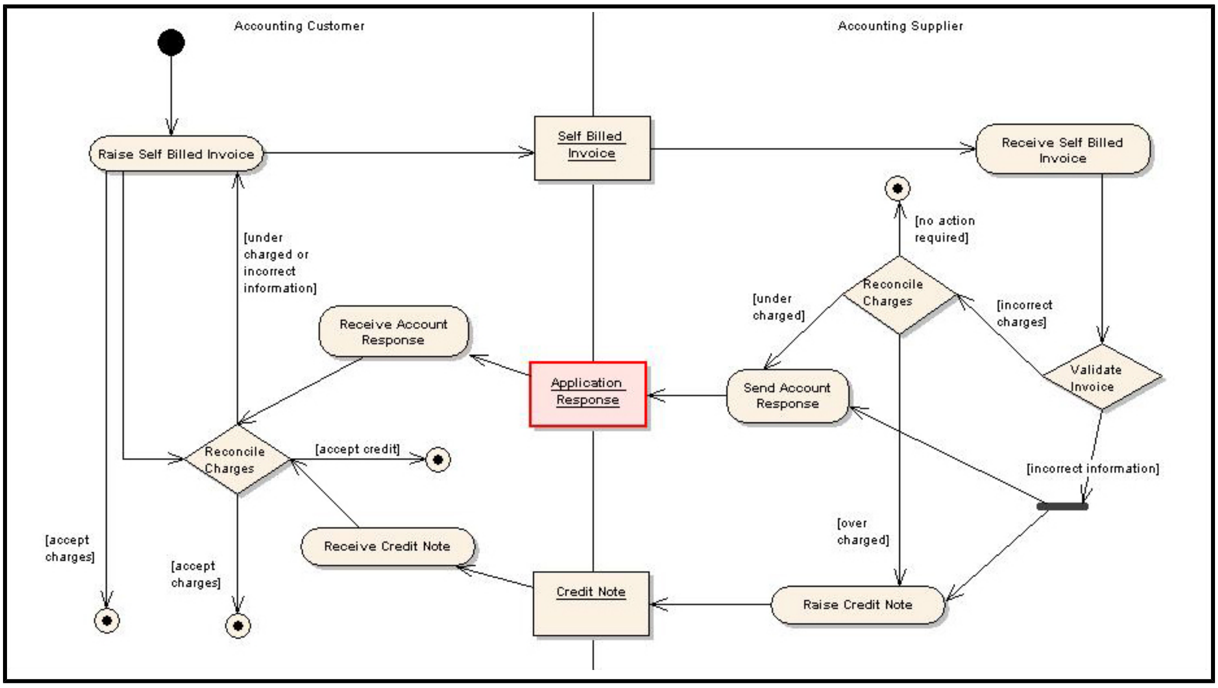
2.8.3. Self Billing

A self billing process is where a Customer “invoices” itself, *in the name and on behalf of* the Supplier, and provides the Supplier with a copy of the self billed invoice.

2.8.3.1. Self Billing Using Credit Notes

Self Billing using Credit Notes is shown in the following diagram.

Figure 22. Self Billing with Credit Note Process

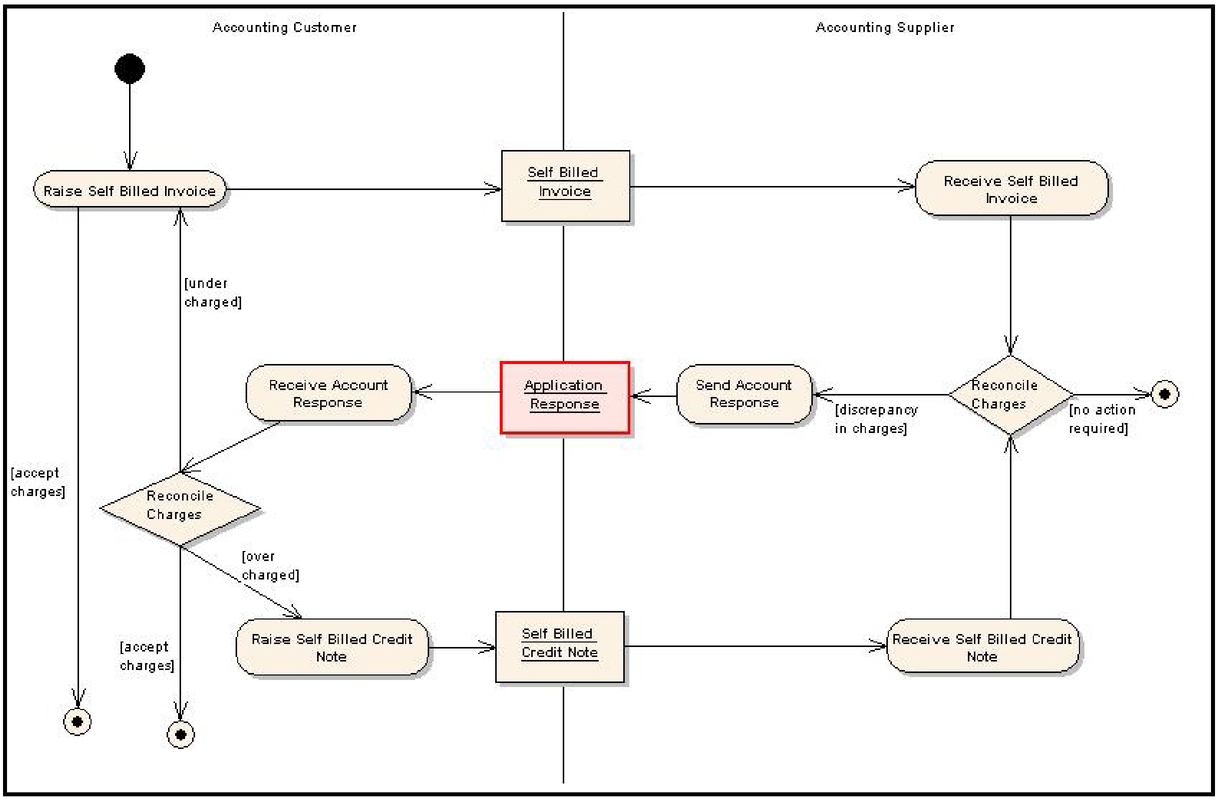


If the Supplier finds that the Self Billed Invoice is incorrect, e.g., wrong quantities or wrong prices, or if the goods have not been invoiced at all, it may send an Application Response or a Credit Note to the Customer. The customer may then verify whether the adjustment is acceptable or not and consequently issue another Self Billed Invoice or a Self Billed Credit Note.

2.8.3.2. Self Billing Using Self Billed Credit Notes

Self Billing using Self Billed Credit Notes is shown in the following diagram.

Figure 23. Self Billing with Self Billed Credit Note Process

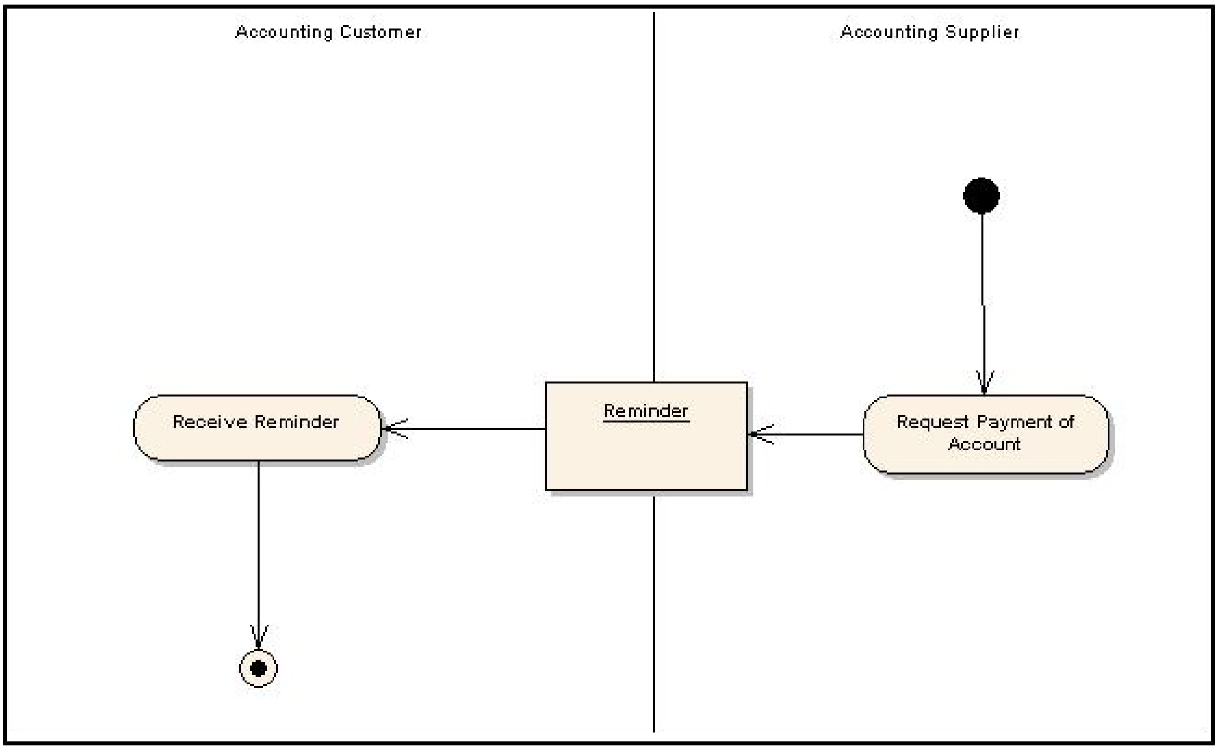


When using Self Billed Credit Notes, the Customer is raising the Self Billed Credit Note *in the name and on behalf of* the Supplier. Therefore the Supplier and the Customer are still both responsible for providing taxation information.

2.8.4. Reminder for Payment

A Reminder may be used to notify the Customer of accounts due to be paid.

Figure 24. Reminder for Payment Process



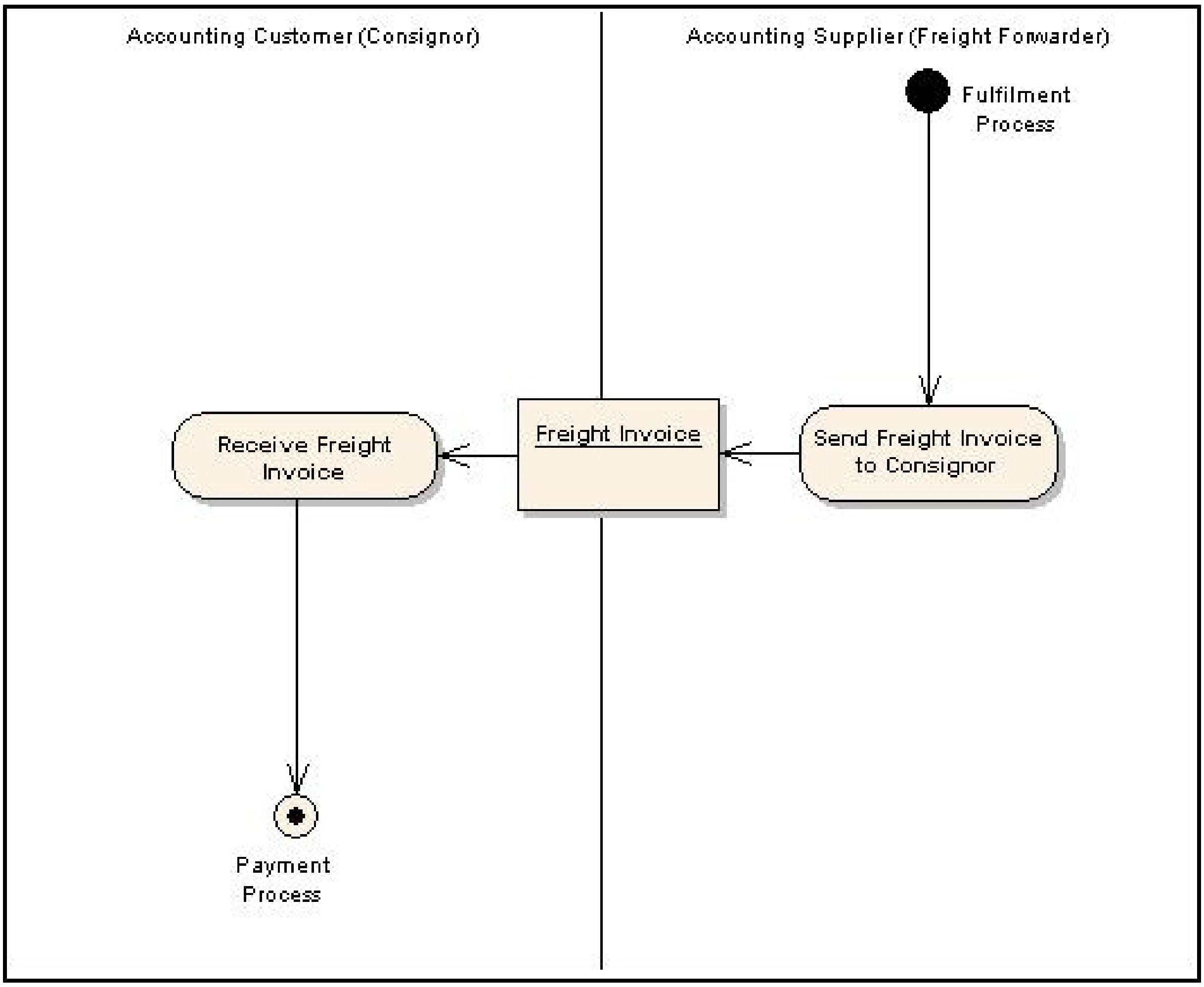
2.9. Freight Billing

An extension of the Billing process is that of Freight Billing. This represents the billing process between the Transport Service Buyer and Transport Service Provider through the use of an Invoice for freight charges.

The Transport Service Provider initiates the process of billing the Transport Service Buyer for logistic services.

The Freight Invoice lists the charges incurred in order to fulfil the agreed service.

Figure 25. Freight Billing Process

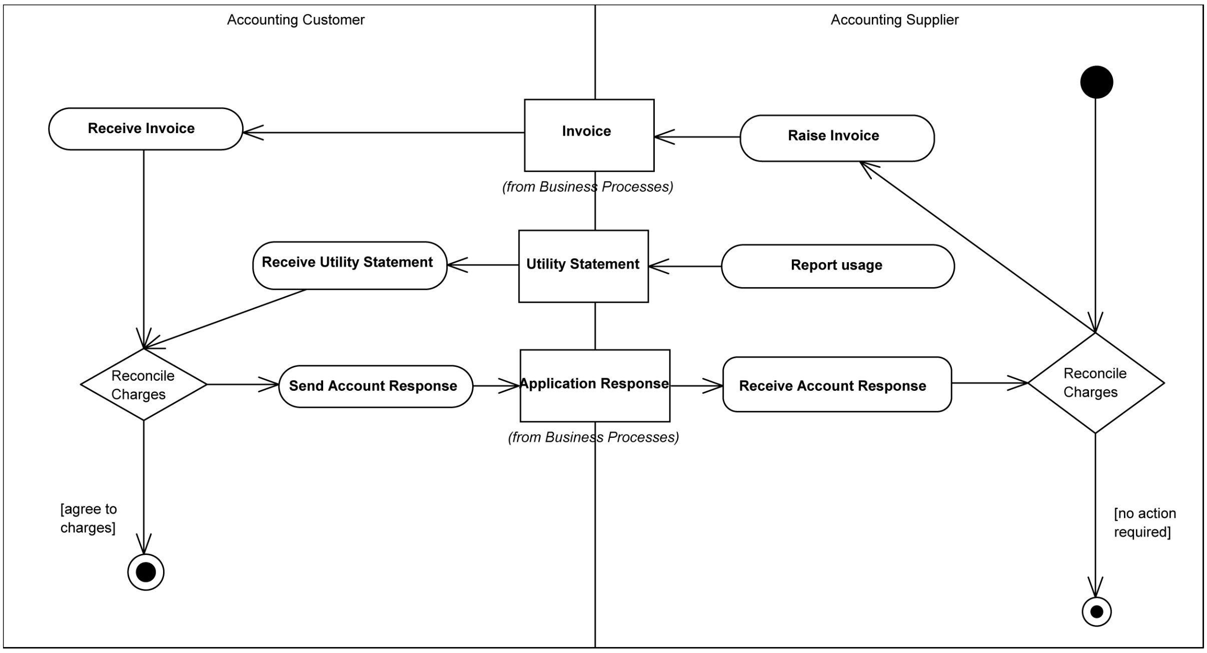

Accounting Customer (Consignor)
Accounting Supplier (Freight Forwarder)
Fulfilment Process
Send Freight Invoice to Consignor
Freight Invoice
Receive Freight Invoice
Payment Process

2.10. Utility Billing

This process defines the billing process for invoicing between suppliers of utilities (including electricity, gas, water, and telephony services) and private and public customers.

The Utility Statement supplements an Invoice with information about consumption of the utility’s services.

Figure 26. Utility Billing Process

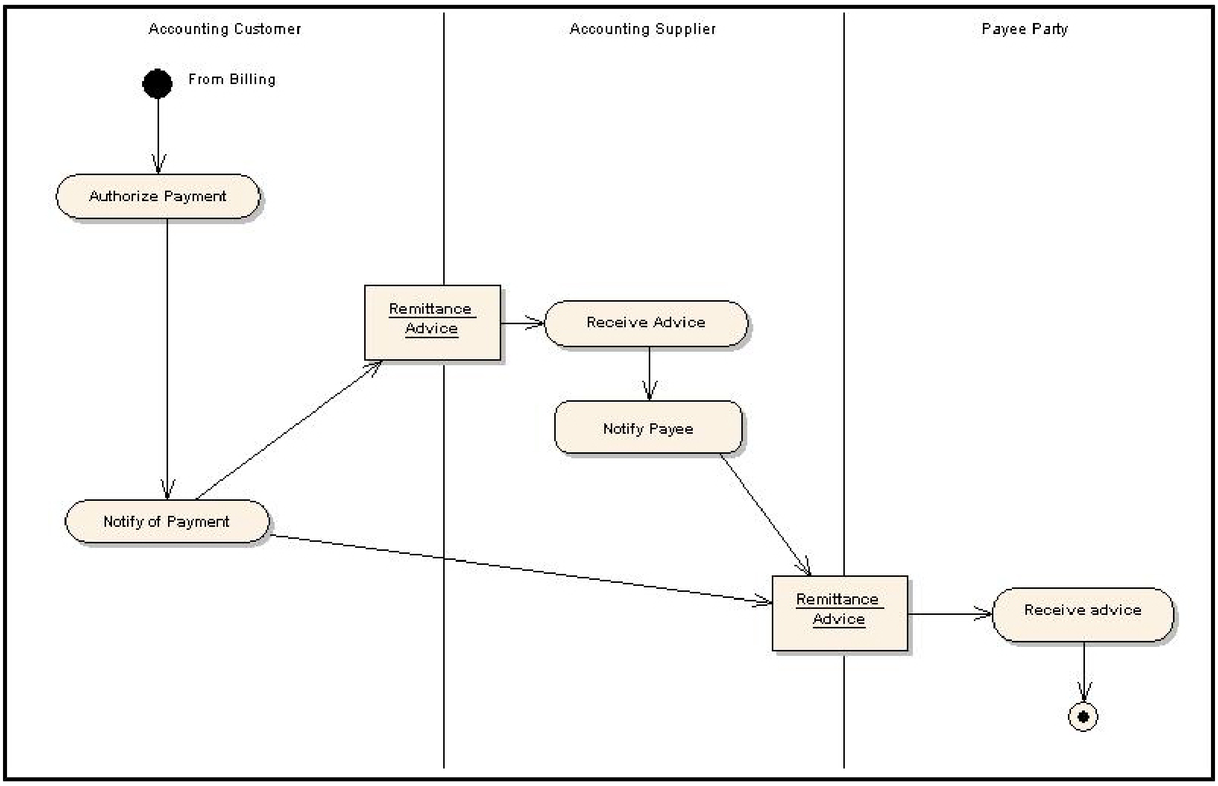


2.11. Payment

In the payment process, the Payee (who is most often the Accounting Customer) is notified of any funds transferred, against the account of the Accounting Supplier, using a Remittance Advice.

The document type in this process is the Remittance Advice.

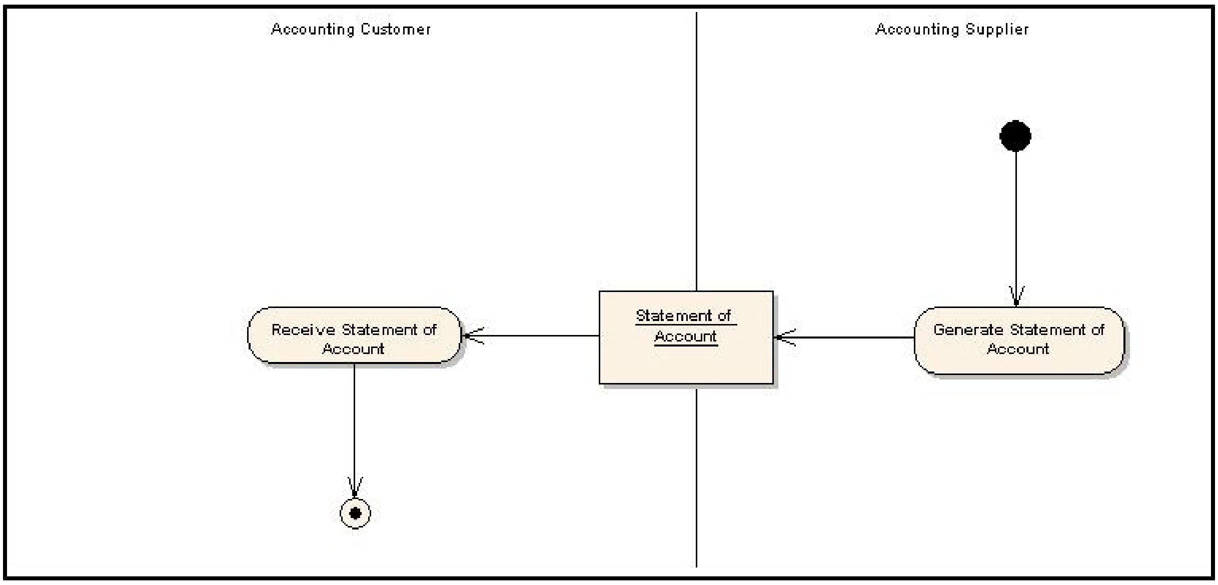
Figure 27. Payment Process



2.11.1. Report State of Accounts

A Statement of Account may be used to notify the Accounting Customer of the status of the billing.

Figure 28. Statement Process

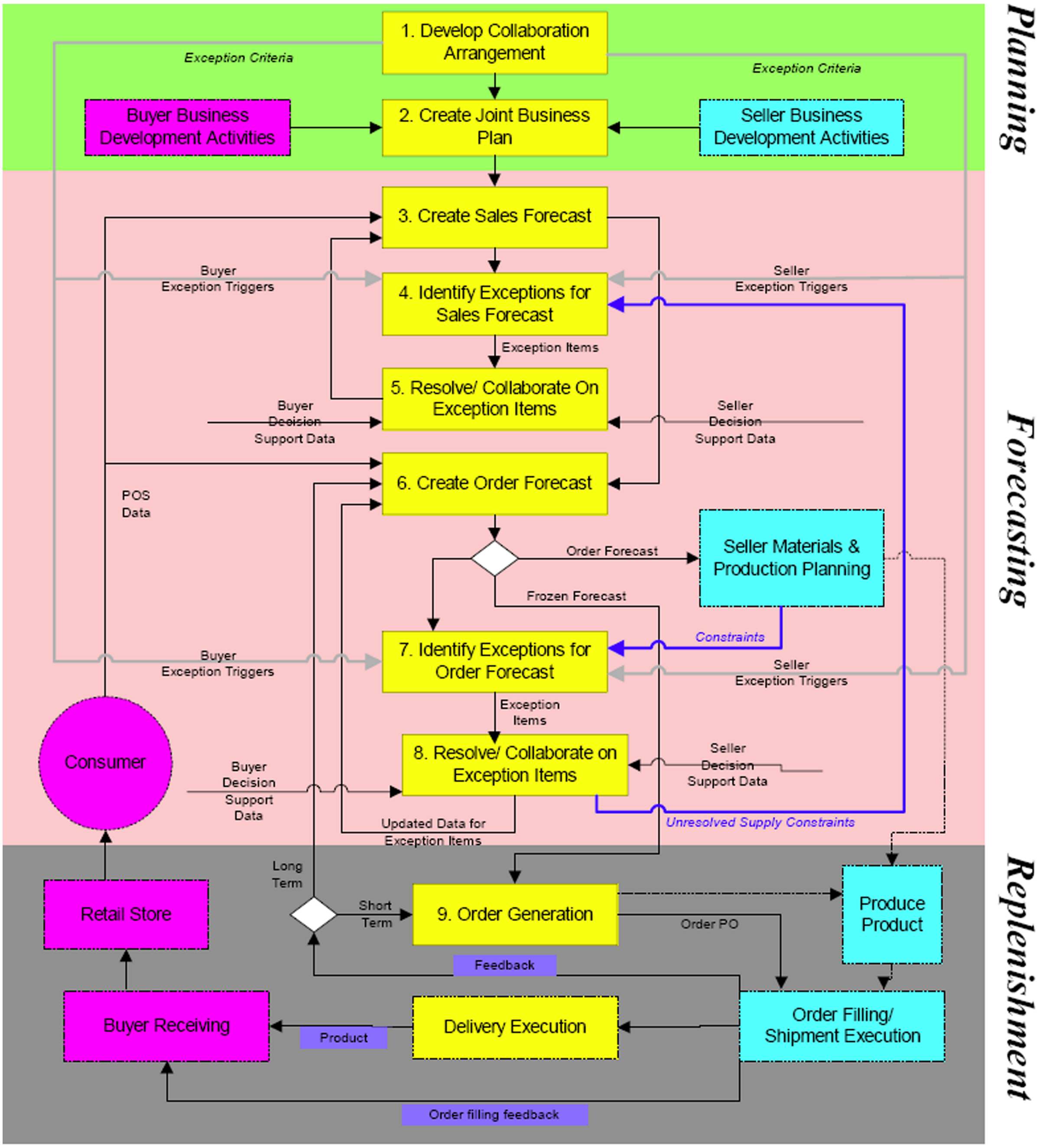


2.12. Collaborative Planning, Forecasting, and Replenishment

The VICS Collaborative Planning, Forecasting, and Replenishment (CPFR®) guidelines [CPFR] formalize the processes by which two trading partners agree upon a joint plan to forecast and monitor sales through replenishment and to recognize and respond to any exceptions.

In the UBL 2.1 context of use, these CPFR processes between the retailer and the manufacturer have been extended to cover the planning process between other parties such as the manufacturer and the supplier. These binary collaboration definitions are the template guidelines for implementers to build their own collaboration process based on their supply chain topology and requirements.

Figure 29. CPFR


Planning
Forecasting
Replenishment
1. Develop Collaboration Arrangement
Exception Criteria
Exception Criteria
Buyer Business Development Activities
2. Create Joint Business Plan
Seller Business Development Activities
3. Create Sales Forecast
Buyer Exception Triggers
Seller Exception Triggers
4. Identify Exceptions for Sales Forecast
Exception Items
5. Resolve/ Collaborate On Exception Items
Buyer Decision Support Data
Seller Decision Support Data
POS Data
6. Create Order Forecast
Order Forecast
Seller Materials & Production Planning
Frozen Forecast
Constraints
Buyer Exception Triggers
7. Identify Exceptions for Order Forecast
Seller Exception Triggers
Exception Items
Consumer
Buyer Decision Support Data
8. Resolve/ Collaborate on Exception Items
Seller Decision Support Data
Unresolved Supply Constraints
Updated Data for Exception Items
Long Term
Short Term
Retail Store
9. Order Generation
Order PO
Produce Product
Feedback
Buyer Receiving
Product
Delivery Execution
Order Filling/ Shipment Execution
Order filling feedback

As shown above, the seller and the buyer employ four main activities in order to improve the overall performance of the supply chain:

1. **Strategy and Planning** establish the ground rules for the collaborative relationship. Trading partners exchange information about their corporate strategies and business plans in order to collaborate on developing a joint business plan. The Joint Business Plan identifies the significant events that affect supply and demand in the planning period, such as promotions, inventory policy changes, store openings/closings, and product introductions.

2. The **Demand and Supply Management** phase involves the development of a shared plan on the consumer demand. The consumer demand at the point of sale is categorized as *sales forecasting* and the future product ordering based on the sales forecast is referred as *order forecast.*

3. **Execution** involves *order generation,* which transitions forecasts to firm demand, and *order fulfilment,* the process of producing, shipping, delivering, and stocking products for consumer purchase. Note: This phase may be implemented using other UBL processes.

4. The **Analysis** phase involves monitoring the execution of activities for exceptions that are identified during the strategy and planning phase. Calculation of key performance metrics and plan adjustments for improving results also take place in this phase.

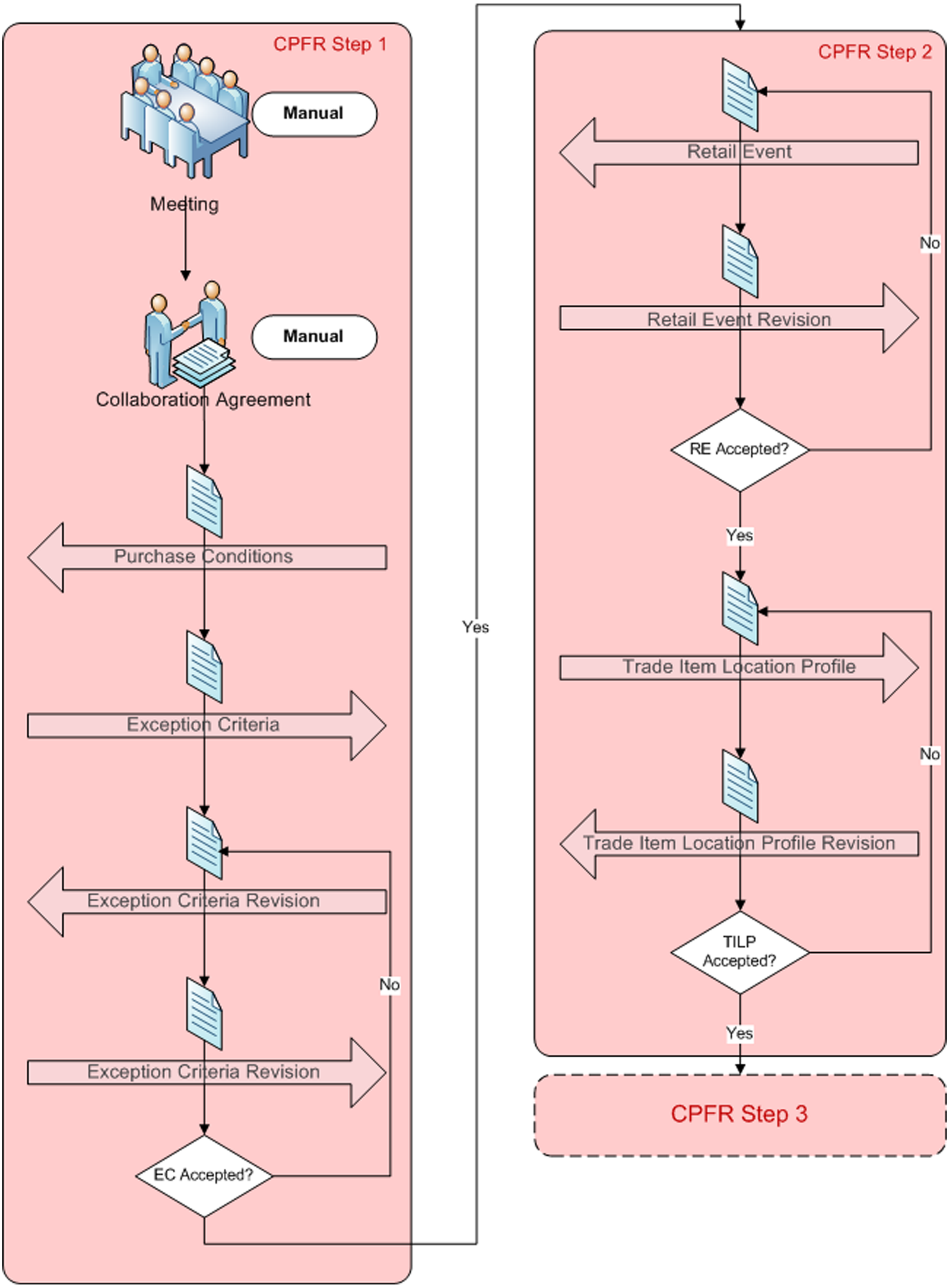
While these collaboration activities are presented in logical order, most companies are involved in all of them at any moment in time. There is no predefined sequence of steps. Execution issues can impact strategy, and analysis can lead to adjustments in forecasts.

2.12.1. Collaboration Agreement and Joint Business Planning

The Collaboration Arrangement is the preparatory step that defines the scope of the project, assigns roles, establishes procedures for data interchange, and issues identification and resolution. The following actions are performed through meetings and agreements:

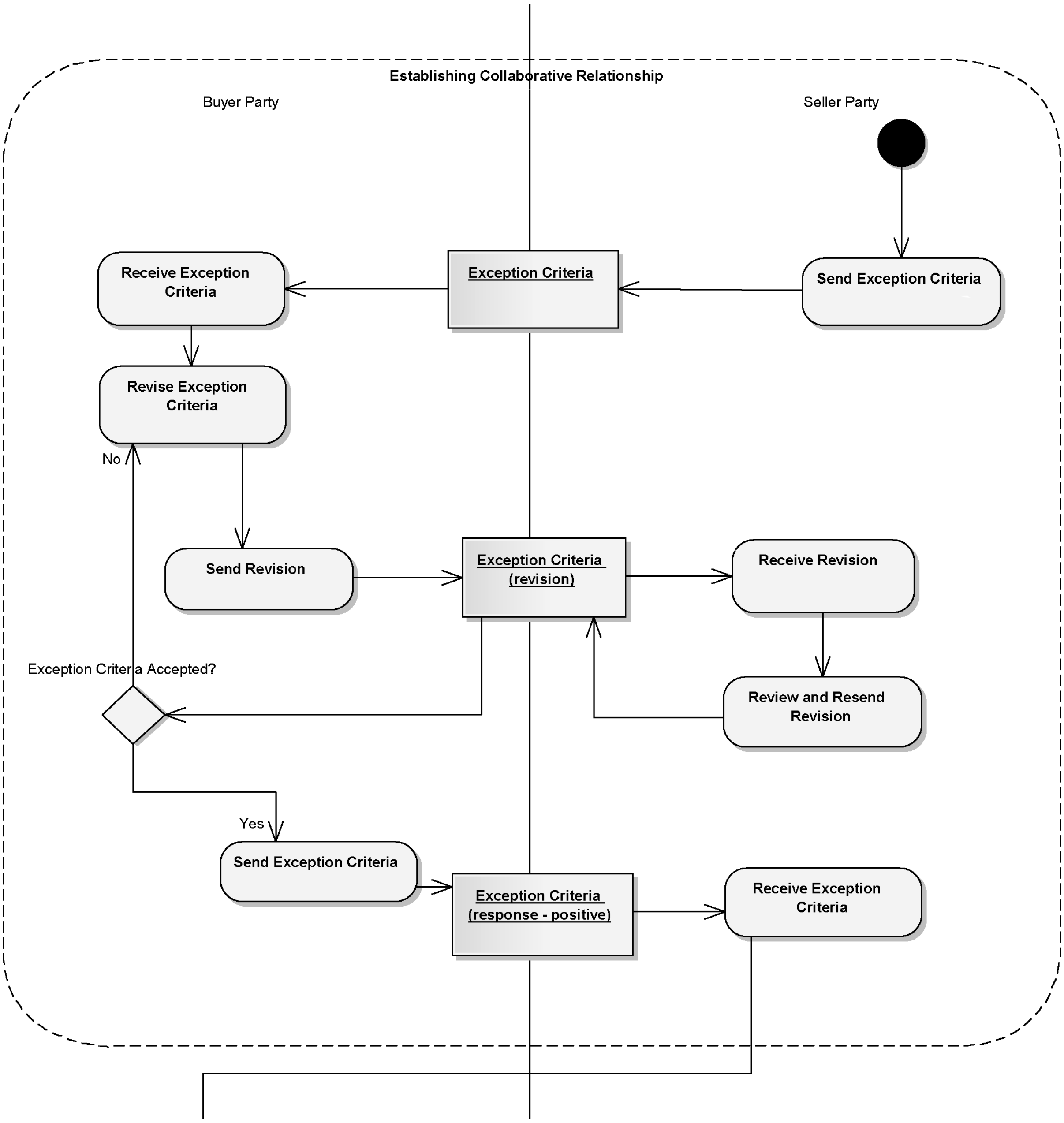
- Receive and review background information from the sales organization or buyers
- Identify the product categories that should be included in the initial scope
- Define Collaboration Objectives
- Define specific metrics that reflect the objectives
- Determine the Event collaboration cycle
- Determine the times of the review meetings to discuss the results
- Document the data sources that are essential for a successful event collaboration process, and
- Document additional information that can be used in the event analysis.

Figure 30. CPFR Steps 1 and 2



The first step of the CPFR Process continues with the exchange of messages containing purchase conditions. (In this initial OASIS public review build, these are assumed to be generic purchase condition messages. A UBL document type for Purchase Conditions is planned for addition in the second public review cycle.) Afterwards, for determining the exception criteria that should be monitored and handled during the execution, Exception Criteria messages are exchanged. Exchange of Exception Criteria Revision messages continues until the criteria are accepted by both sides.

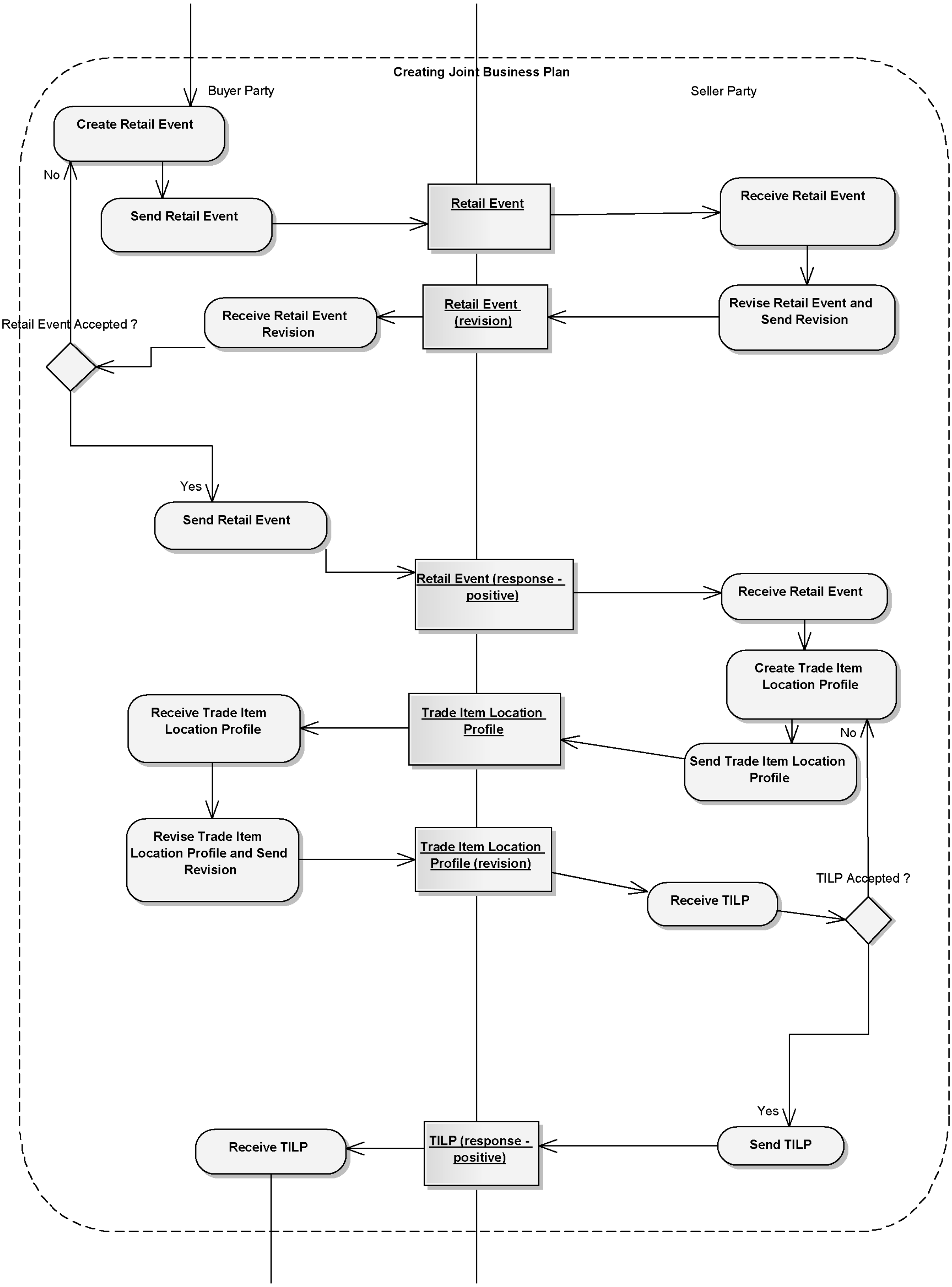
Figure 31. Establish Collaborative Relationships



Establishing Collaborative Relationship
Buyer Party
Seller Party
Receive Exception Criteria
Exception Criteria
Send Exception Criteria
Revise Exception Criteria
No
Send Revision
Exception Criteria (revision)
Receive Revision
Exception Criteria Accepted?
Review and Resend Revision
Yes
Send Exception Criteria
Exception Criteria (response - positive)
Receive Exception Criteria

In CPFR Step 2 (the Joint Business Planning phase) there are two messages that should be exchanged and agreed upon: Retail Event and Trade Item Location Profile. The revisions are exchanged until an agreement is achieved.

Figure 32. Create Joint Business Plan

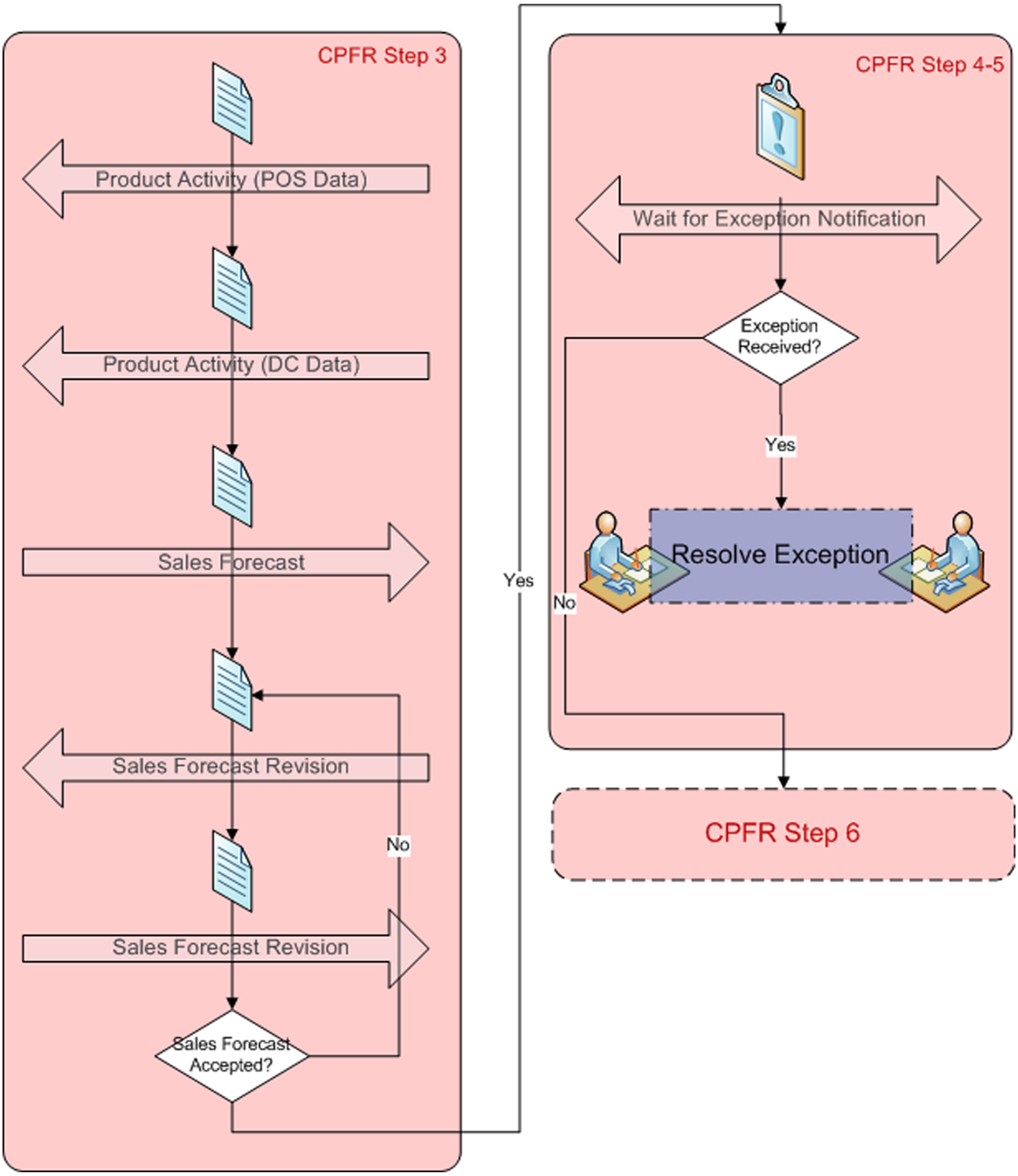

Creating Joint Business Plan
Buyer Party
Seller Party
Create Retail Event
No
Send Retail Event
Retail Event
Receive Retail Event
Retail Event (revision)
Revise Retail Event and Send Revision
Receive Retail Event Revision
Retail Event Accepted ?
Yes
Send Retail Event
Retail Event (response - positive)
Receive Retail Event
Create Trade Item Location Profile
Receive Trade Item Location Profile
Trade Item Location Profile
No
Send Trade Item Location Profile
Revise Trade Item Location Profile and Send Revision
Trade Item Location Profile (revision)
TILP Accepted ?
Receive TILP
Yes
Receive TILP
TILP (response - positive)
Send TILP

2.12.2. Sales Forecast Generation and Exception Handling

CPFR Step 2 helps the buyer and seller agree to the event details and calendar that meet their joint business and collaboration objectives. The objective of the event calendar is to ensure that events are planned to achieve the optimal results and to enable both parties to plan the execution of the event more accurately, from the preparation of advertising and displays to the production and delivery of the promotional stock.

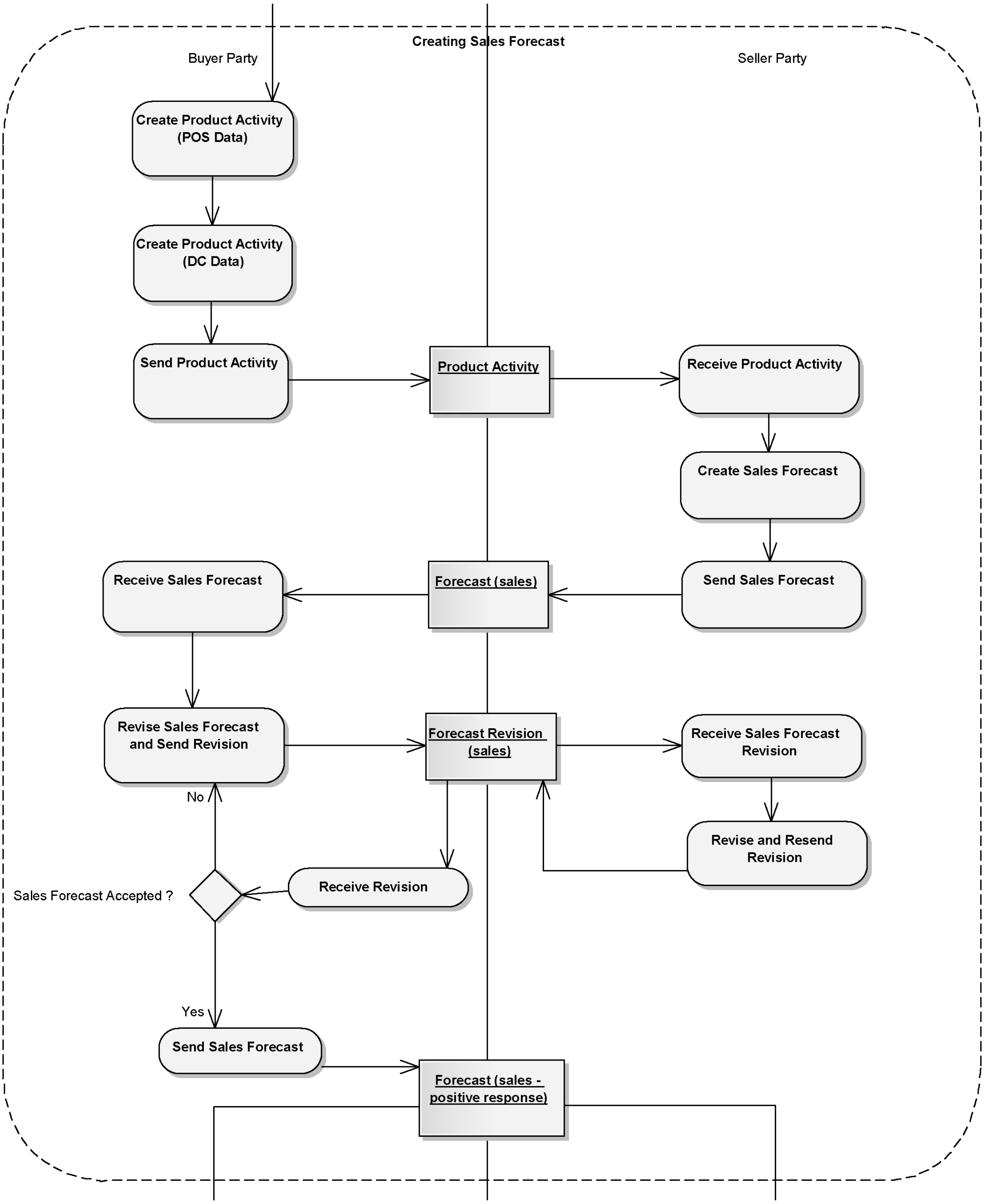
In CPFR Step 3, the Sales Forecast is generated. Following Option A, Conventional Order Management, from the CPFR implementation scenarios (see [CPFRoverview], Table 3), the responsible partner for the generation of Sales Forecast is the Seller. Having Event Calendar information and the Delivery Plan already in their system, there are two more kinds of information that the Seller needs for an effective Sales Forecast: POS Data and DC Data. As shown in Figure 25, both of these pieces of information are sent within a Product Activity Message. This time there is no revision of the messages because these messages contain statistical and historical information collected previously by the Buyer.

Figure 33. CPFR Steps 3, 4, and 5



Based on the event details (dates, products, tactics, etc.) and using the available data source(s), a volume estimate/forecast is created for each product/store combination included in the scope of the event by the Seller. During the calculation, sales forecasting algorithms make use of the coefficients for causal factors based on the event history. Once the Sales Forecast suggestion is generated and sent to the Buyer, the Buyer revises it and might recommend some changes on the Forecast. The Forecast Revision message exchange continues until the forecast is agreed by both sides.

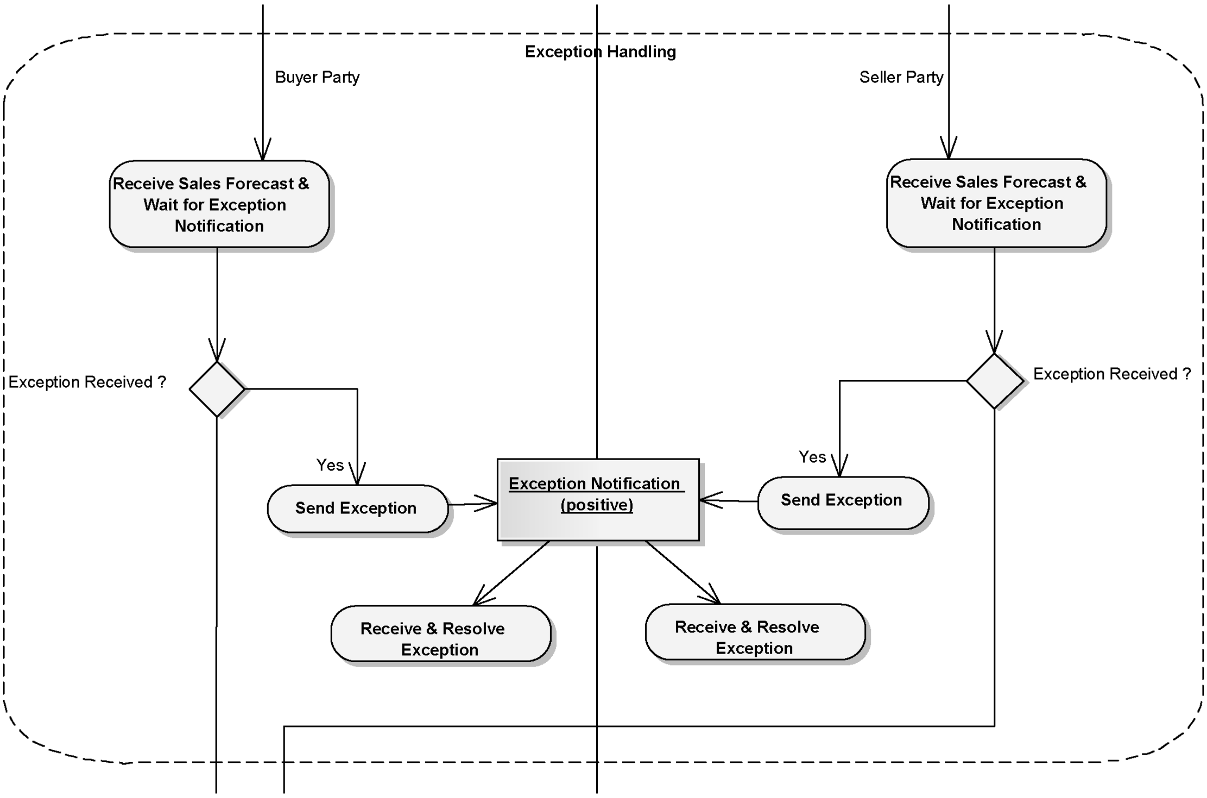
Figure 34. Create Sales Forecast


Creating Sales Forecast
Buyer Party
Seller Party
Create Product Activity (POS Data)
Create Product Activity (DC Data)
Send Product Activity
Product Activity
Receive Product Activity
Create Sales Forecast
Send Sales Forecast
Forecast (sales)
Receive Sales Forecast
Revise Sales Forecast and Send Revision
Forecast Revision (sales)
Receive Sales Forecast Revision
Revise and Resend Revision
No
Sales Forecast Accepted ?
Receive Revision
Yes
Send Sales Forecast
Forecast (sales - positive response)

On average, six weeks elapse between Sales Forecast Generation and Order Generation. During this period, both sides observe changes to the conditions. If one of the partners detects an exception invalidating the exception criteria defined in CPFR Step 1, it sends an Exception Notification message to the other party. Exceptional circumstances that may be communicated between trading partners include deviations between planned impacts (either between buyer and seller, or between subsequent generations of planned impacts from the same trading partner), as well as deviations between planned and actual impacts. It should be noted that both sides might detect an exception, and therefore both sides should be capable of sending and receiving exceptions. Of course, for specific implementations if the collaborating parties want to change this behaviour, they can customize the process so that one partner will be responsible for the generation of the exception notifications.

CPFR Step 4 is solely composed of the exception generation and receiving activity. CPFR Step 5, on the other hand, is the resolution of the Exceptions.

Figure 35. Exception Handling

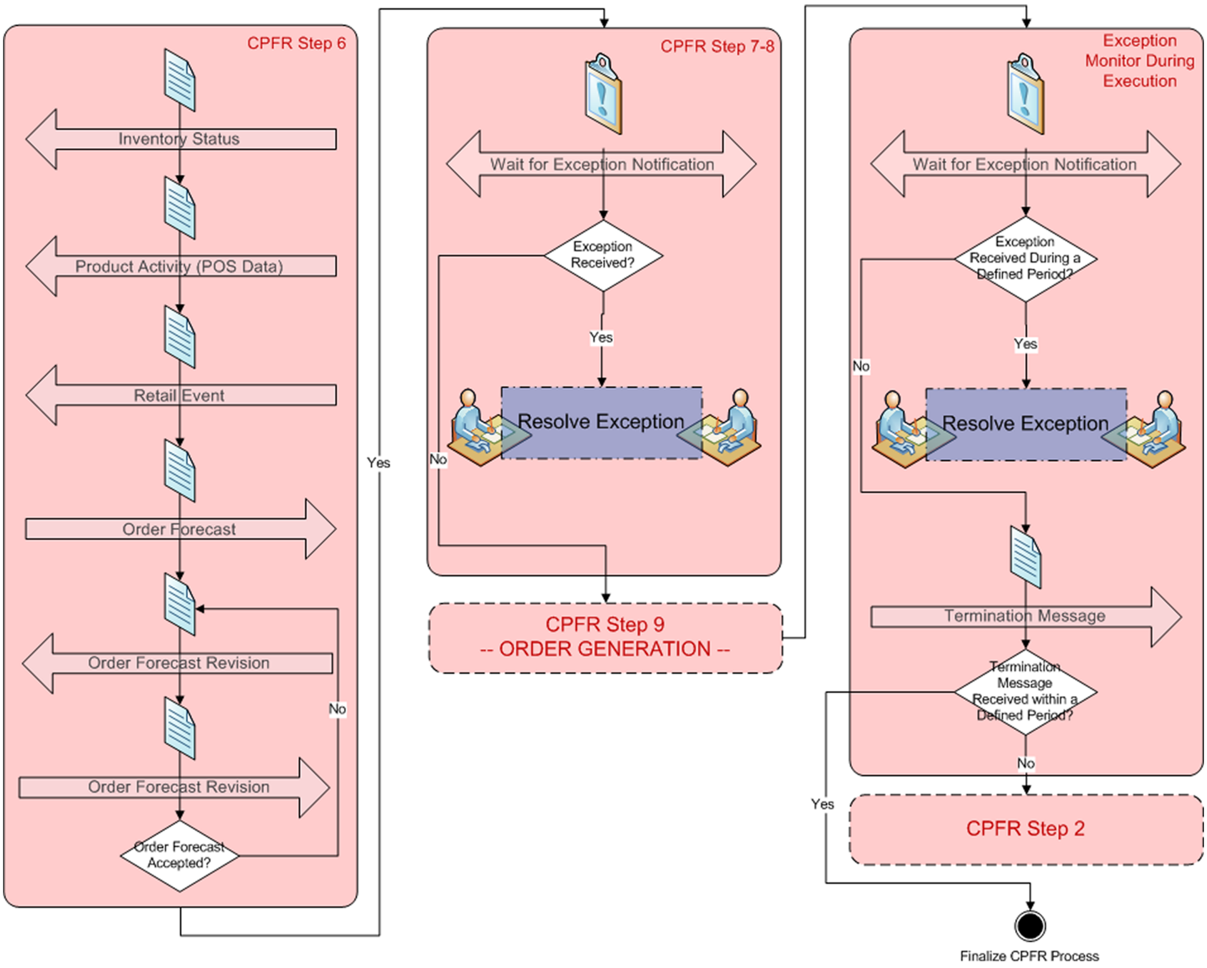

Exception Handling
Buyer Party
Seller Party
Receive Sales Forecast & Wait for Exception Notification
Receive Sales Forecast & Wait for Exception Notification
Exception Received ?
Exception Received ?
Yes
Yes
Send Exception
Exception Notification (positive)
Send Exception
Receive & Resolve Exception
Receive & Resolve Exception

If there is no Exception Notification Message within the defined period, the process continues with Order Forecast Generation (CPFR Step 6).

2.12.3. Order Forecast Generation and Exception Handling

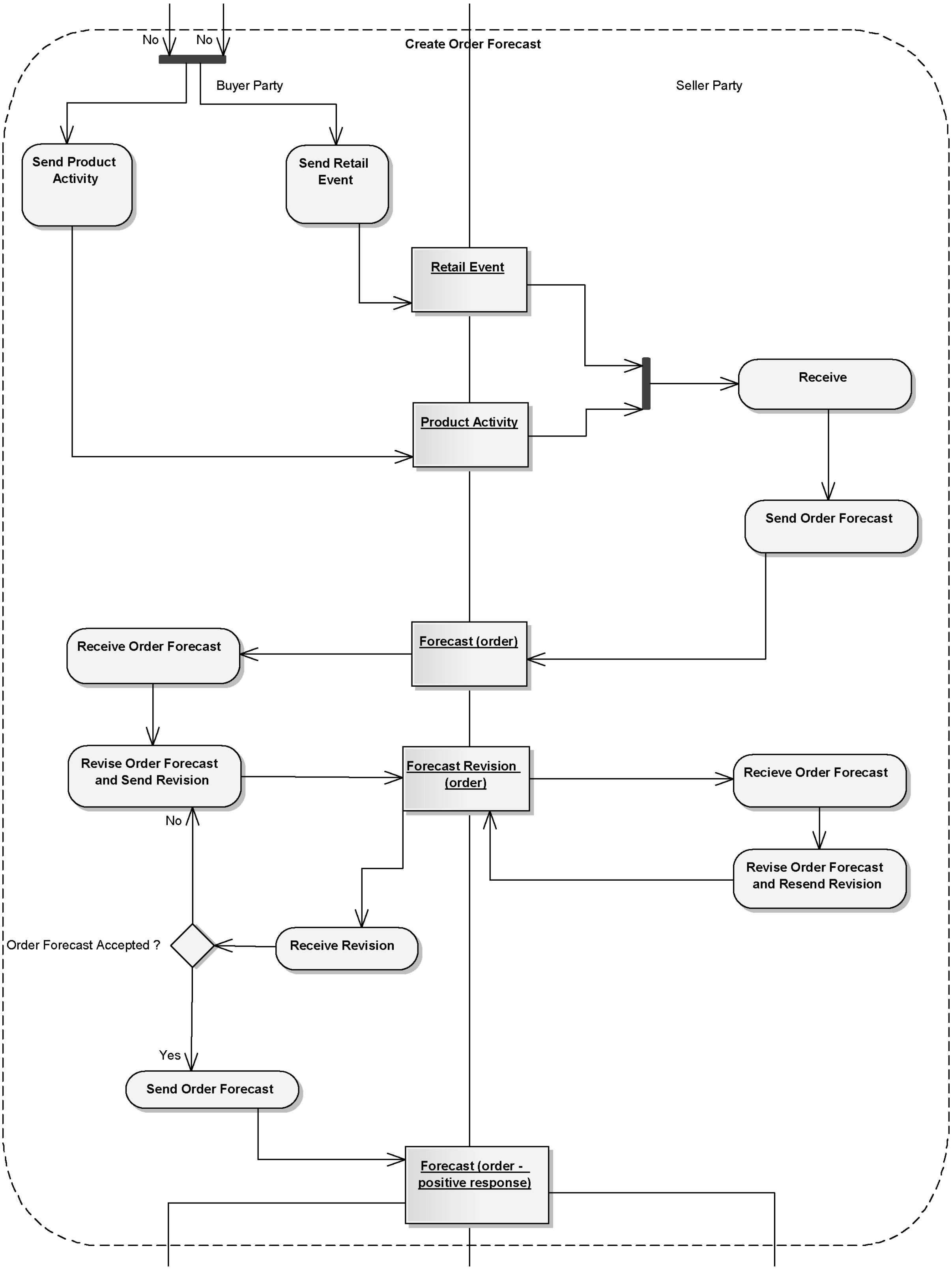
In the supply chain process, it is important for sales forecasts that are created to be converted into the shipment (order) forecasts that can then be used in the production planning processes at the manufacturing locations and be incorporated into the ordering processes at the retailer. As shown in Figure 28, the responsibility for creating Order Forecast belongs to the Seller per Option A of the CPFR implementation scenarios (see [CPFRoverview], Table 3). Sales forecasts can be transformed into order forecasts by incorporating inventory status information, possible retail event plans, and current point of sale data. Therefore, Buyer sends the updated versions of the Retail Event, Inventory Status, and POS Data to the Seller.

Figure 36. CPFR Steps 6, 7, 8 and 9



After the Seller creates the Order Forecast using the obtained data, it sends the forecast to the Buyer. The Buyer checks the order forecast and sends back a revision document which includes update requests if necessary. The exchange of Order Forecast Revisions continues until there are no further update requests and the Order Forecast is agreed by both sides.

Figure 37. Create Order Forecast



After the Order Forecast is frozen, the process continues with the exception detection activity (CPFR Step 7). The exception detection process that follows Order Forecast is similar to process described earlier for exeption detection following Sales Forecast (see Section 2.12.2, “Sales Forecast Generation and Exception Handling”). The only difference between the Order Forecast and Sales Forecast exceptions is the content of the exceptions.

CPFR Step 8, Order Forecast Exception Resolution activity, is handled similarly to Sales Forecast Exception Resolution.

Figure 38. Identifying and Resolving Exceptions for Order Forecast

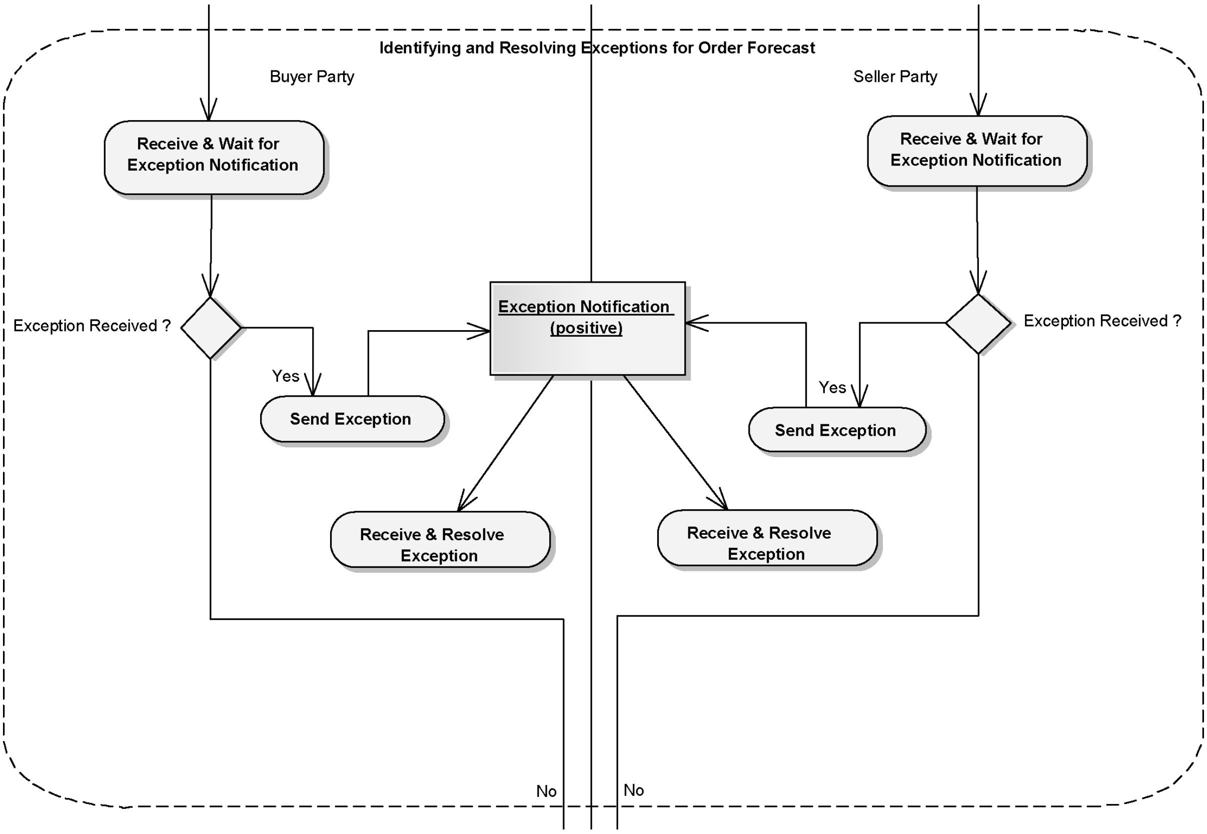
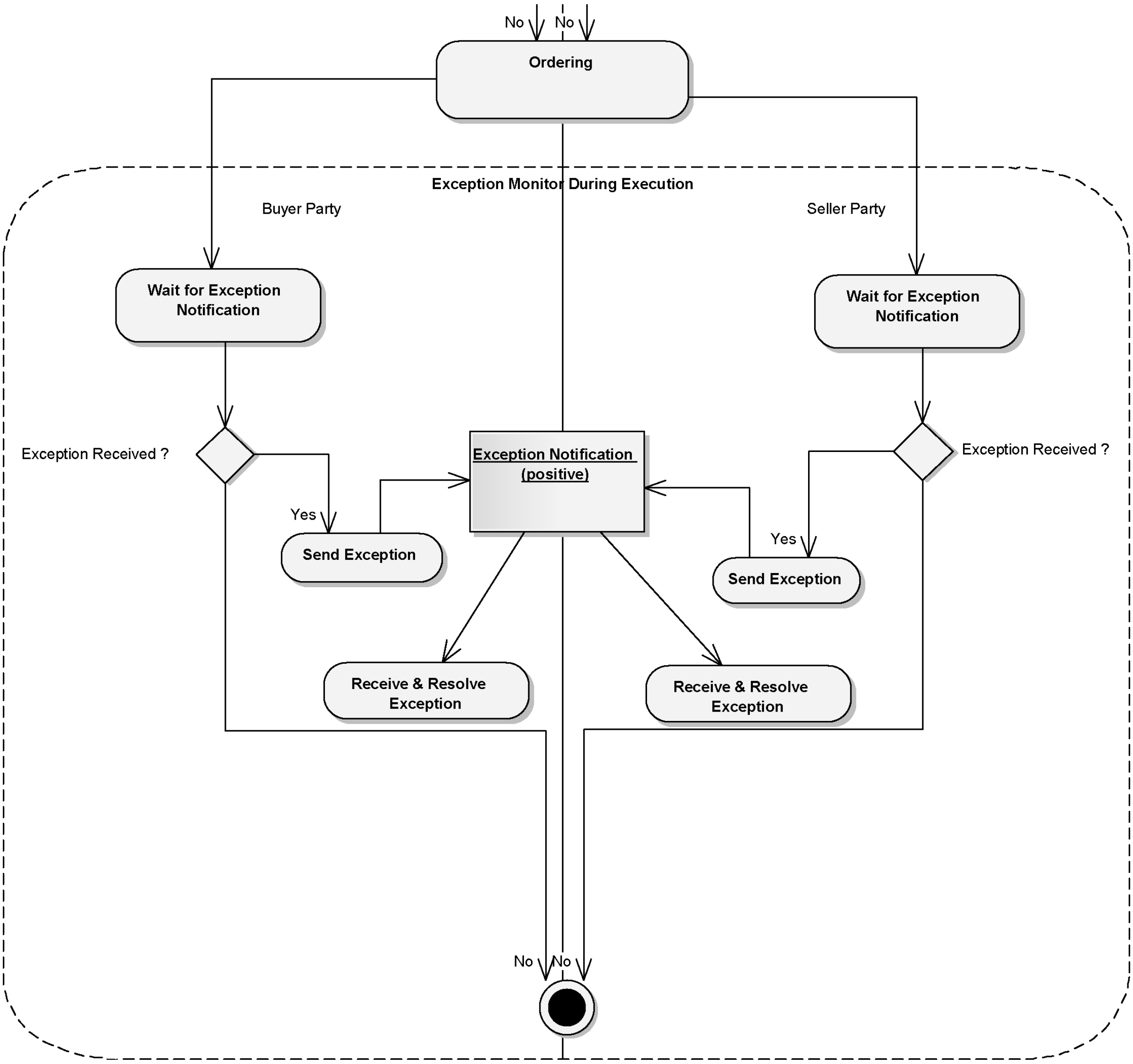

Identifying and Resolving Exceptions for Order Forecast
Buyer Party
Seller Party
Receive & Wait for Exception Notification
Receive & Wait for Exception Notification
Exception Received ?
Exception Received ?
Exception Notification (positive)
Yes
Yes
Send Exception
Send Exception
Receive & Resolve Exception
Receive & Resolve Exception
No
No

Figure 39. Exception Monitor During Execution


No
No
Ordering
Exception Monitor During Execution
Buyer Party
Seller Party
Wait for Exception Notification
Wait for Exception Notification
Exception Received ?
Exception Received ?
Exception Notification (positive)
Yes
Yes
Send Exception
Send Exception
Receive & Resolve Exception
Receive & Resolve Exception
No
No

If there is no exception during a period of time, the process continues with the Order Generation Step.

From the technical point of view, the exception monitoring and its resolution are exactly same as in the case of Order Forecast Exception Handling and Sales Forecast Exception Handling. The difference is in the content of the exceptions. The actual events and orders are compared to the Forecasted Sales and Forecasted Orders. When there is a situation violating the normal exception criteria, one of the sides might generate an exception notification. Besides comparison of forecasts, other information gathered during the execution is observed (e.g., event dates, POS data, etc.). The resolution of the exceptions is the same as the process carried out for Sales Forecast Exception resolution.

During this exception monitoring time, Buyer also waits for a possible Termination message indicating the end of the collaboration. The Termination message is not mentioned by the CPFR guidelines but is included in UBL to indicate the successful termination of the CPFR collaboration. Since the CPFR Planning Process is supposed to be an iterative process, most of the time there will not be a termination message and the process will continue from CPFR Step 2 with a new cycle of forecast generation and order execution.

2.13. Vendor Managed Inventory

Vendor Managed Inventory (VMI) is a family of business processes in which the Retailer Customer Party for an item provides certain information to the Seller Supplier Party, and the Seller Supplier Party takes full responsibility for maintaining an agreed-upon inventory of the item, usually at the Retailer Customer Party's point of sale. A third party logistics provider can also be involved to make sure that the Retailer Customer Party has the required level of inventory by adjusting the demand and supply gaps.

UBL supports three common models of VMI:

- Basic VMI
- Cyclic Replenishment Program (CRP)
- Replenishment on Customer Demand

These processes are described in more detail below. It should be noted that the particular semantics used here come from a large-scale UBL application developed for the Italian textile and clothing industry by ENEA, the Italian National Agency for New Technologies, Energy, and Sustainable Economic Development (see [eBiz-TCF]). These models are applicable to the implementation of vendor-managed relationships in a broad range of retail sectors, but for the sake of simplicity, and in keeping with the model application, the two principal parties in the VMI relationship (the Seller Supplier Party and the Retailer Customer Party) are referred to as "producer" and "retailer" in the descriptions that follow; more generically, they are vendor and customer.

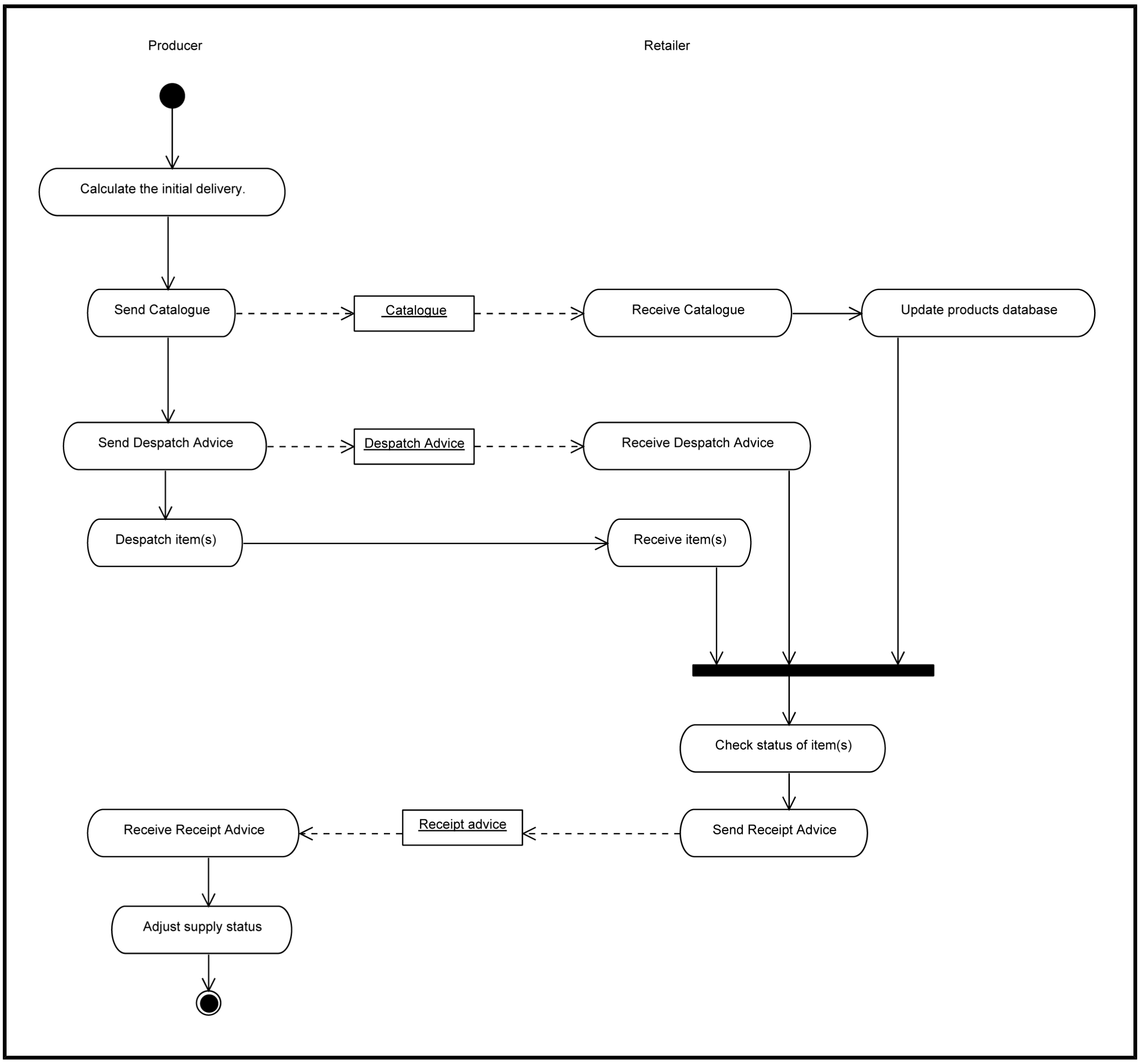
2.13.1. Basic Vendor Managed Inventory

In the classic VMI scenario, a shop-within-a-shop area or an entire store is managed completely by the producer. The logistic concept of VMI can be combined with consignment/concession as well as with charge-on-delivery as the financial model. Mostly it is combined with consignment.

2.13.1.1. Initial Stocking of the Area by Producer

At the beginning of the cooperation, the area is stocked by the producer. The retailer receives item and delivery information and reports back the goods actually received.

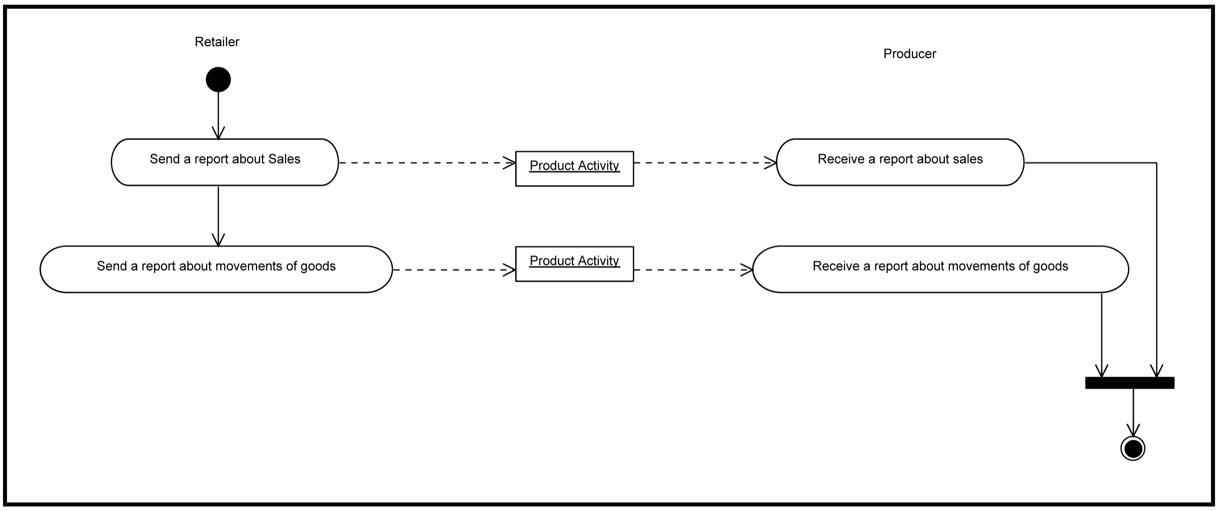
Figure 40. Initial Stocking of the Area by Producer



2.13.1.2. Report of Sales and Inventory Movement

The sales and inventory movement information is transferred from the retailer to the producer.

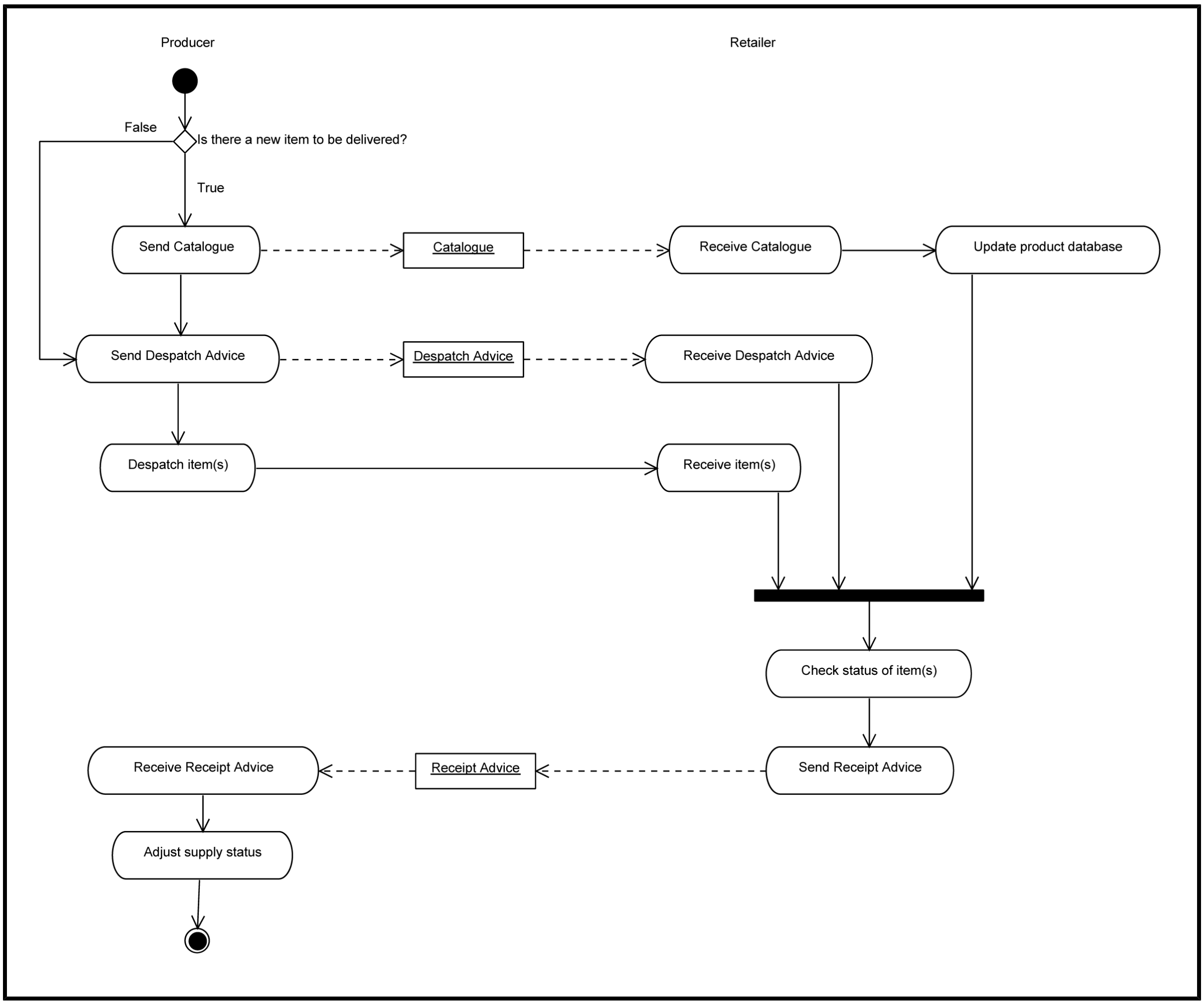
Figure 41. Report of Sales and Inventory Movement


Retailer
Producer
Send a report about Sales
Product Activity
Receive a report about sales
Send a report about movements of goods
Product Activity
Receive a report about movements of goods

2.13.1.3. Permanent Replenishment

Based on sales and inventory movement, the producer periodically makes a new delivery of goods accompanied by a Despatch Advice. If the delivery contains an item not previously stocked, an updated catalogue is also sent so that the retailer can add the item to its product database. Upon delivery of the goods, the retailer reports back the items received using a Receipt Advice.

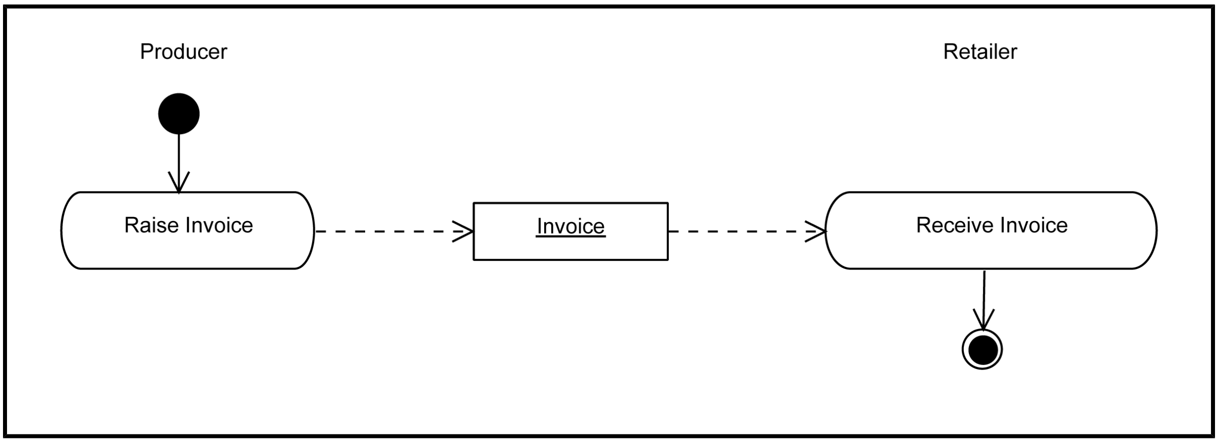
Figure 42. Permanent Replenishment



2.13.1.4. Invoicing for Vendor Managed Inventory

An invoice is sent either on a delivery or a sales basis. In a charge-on-delivery model, the data for the invoice is prepared from the delivery, and in a consignment/concession model from the sales reports.

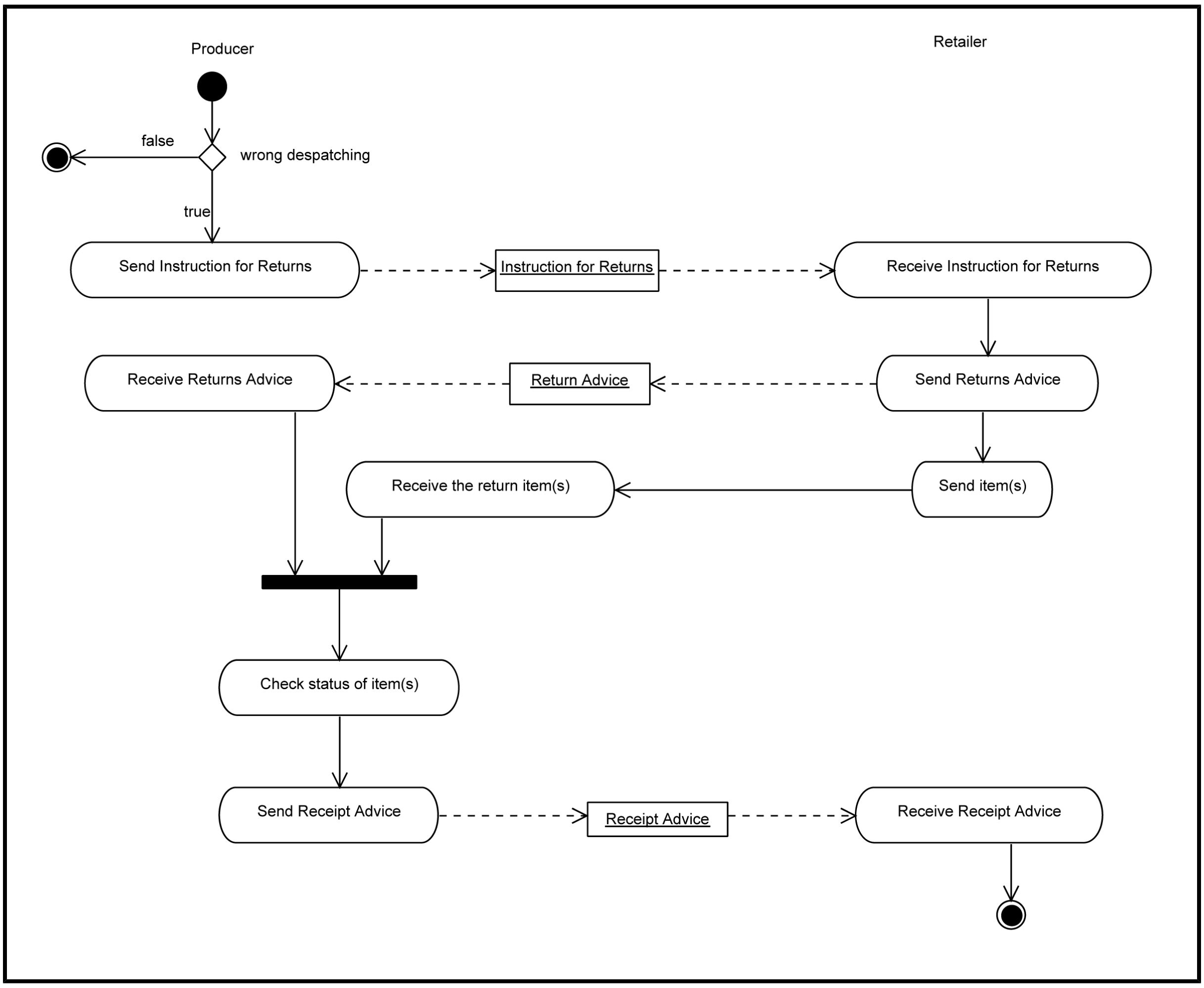
Figure 43. Invoicing for Vendor Managed Inventory



2.13.1.5. Returns Initiated by the Producer

If sales do not meet expectations, items are reallocated by the producer. Because the producer cannot request a retailer to send the products to a competitor, the producer requests a return and handles the goods afterwards by itself.

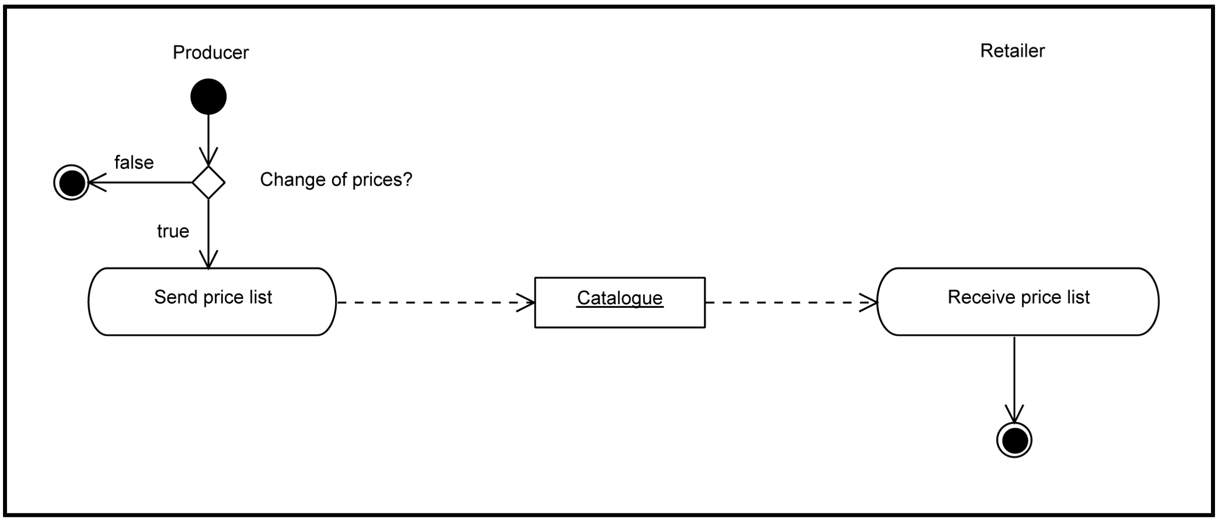
Figure 44. Returns Initiated by the Producer



2.13.1.6. Price Adjustments

In the event of a price change, an updated price list (in the form of a new catalogue containing the change) is sent from producer to retailer.

Figure 45. Price Adjustments



2.13.2. Cyclic Replenishment Program (CRP)

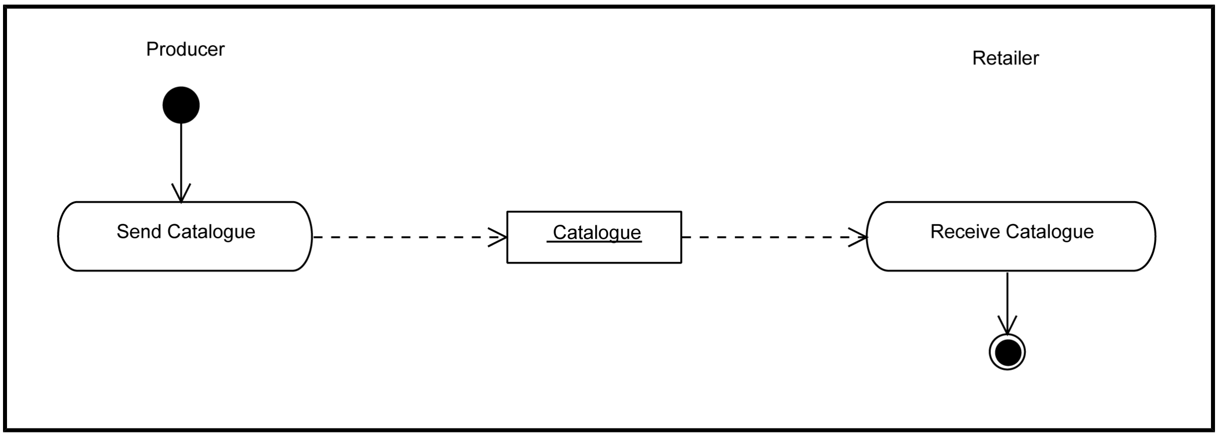
A variant of VMI is the Cyclic Replenishment Program (CRP). In this process, the producer establishes a catalogue of NOS (Never Out of Stock) or seasonal NOS items, and the retailer chooses items for cyclic (weekly) replenishment. The logistic scenario can be combined with the charge-on-delivery as well as with a consignment/concession model. At the end of every sales period, a report of sales and inventory movement at all retail locations is sent to the producer.

CRP differs from the third VMI variant, Replenishment on Customer Demand (below), in that the producer cannot change the terms of the order.

2.13.2.1. Transfer of Base Item Catalogue

The producer publishes the catalogue of its NOS and seasonal NOS items to the retailer.

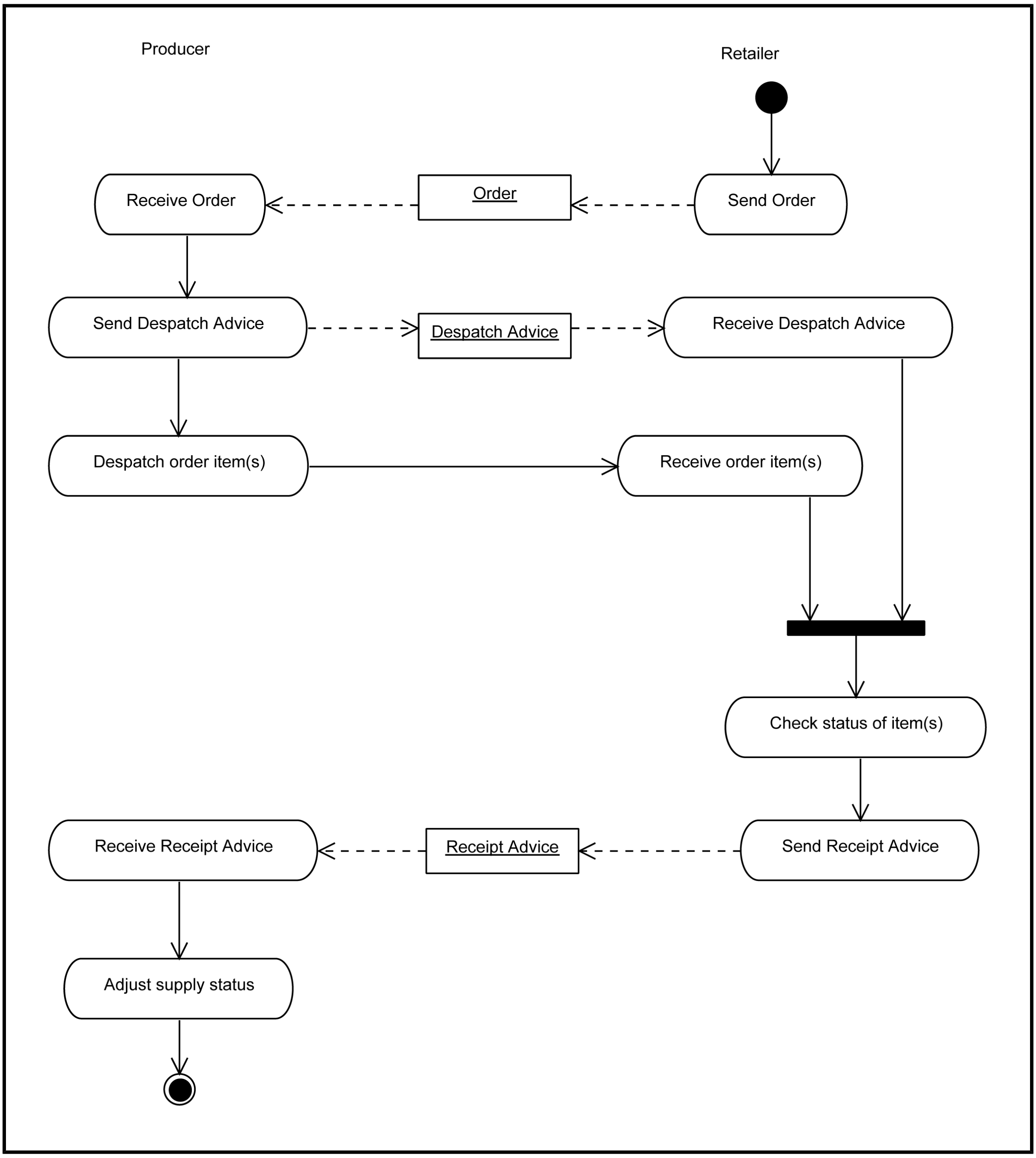
Figure 46. Transfer of Base Item Catalogue



2.13.2.2. Initial Stocking of the Area by Retailer

At the beginning of the cooperative relationship—or the beginning of a season, if seasonal NOS products are the focus—the retailer orders its base stock, and the products are delivered.

Figure 47. Initial Stocking of the Area by Retailer



2.13.2.3. Periodic (Weekly) Replenishment

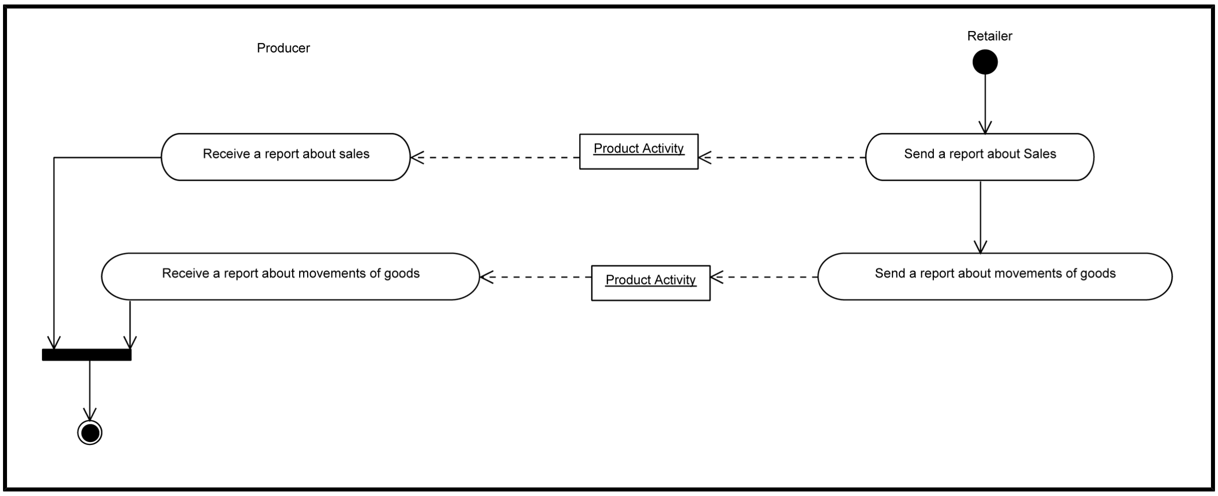
Each period (every week), the retailer's system calculates the quantities needed for replenishment of the product area. From the result, an order is sent, and the producer responds with a direct delivery within 48 hours.

The replenishment process uses the same documents in the same order as the Initial Stocking process, so the duplicate diagram is omitted here; see Figure 47, “Initial Stocking of the Area by Retailer”. It must be remembered, however, that the two processes are taking place at different points in time, so their pre and post conditions will be different.

2.13.2.4. Report of Sales and Inventory Movements

At the end of each sales day, a report of all sales and inventory movement at all retail locations is sent from the retailer to the producer.

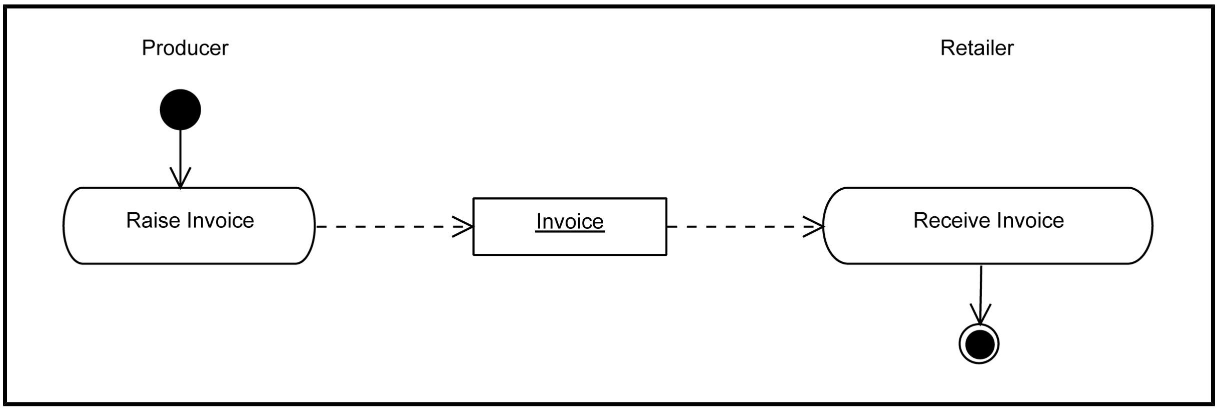
Figure 48. Report of Sales and Inventory Movements


Producer
Retailer
Receive a report about sales
Product Activity
Send a report about Sales
Receive a report about movements of goods
Product Activity
Send a report about movements of goods

2.13.2.5. Cyclic Replenishment Program Invoicing

An invoice is sent either on a delivery or a sales basis.

Figure 49. Invoicing for Cyclic Replenishment Program

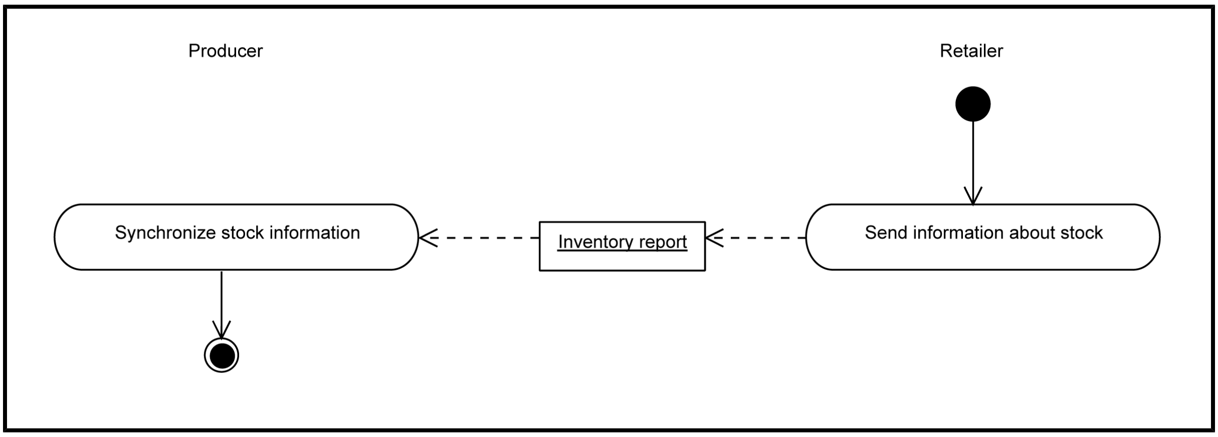


2.13.2.6. Synchronizing of Stock Information

Information about the actual stock is synchronised periodically (for example, every one to three months). This is combined at least once a year with a physical inventory.

The retailer sends an inventory report containing the information about the quantities currently in stock.

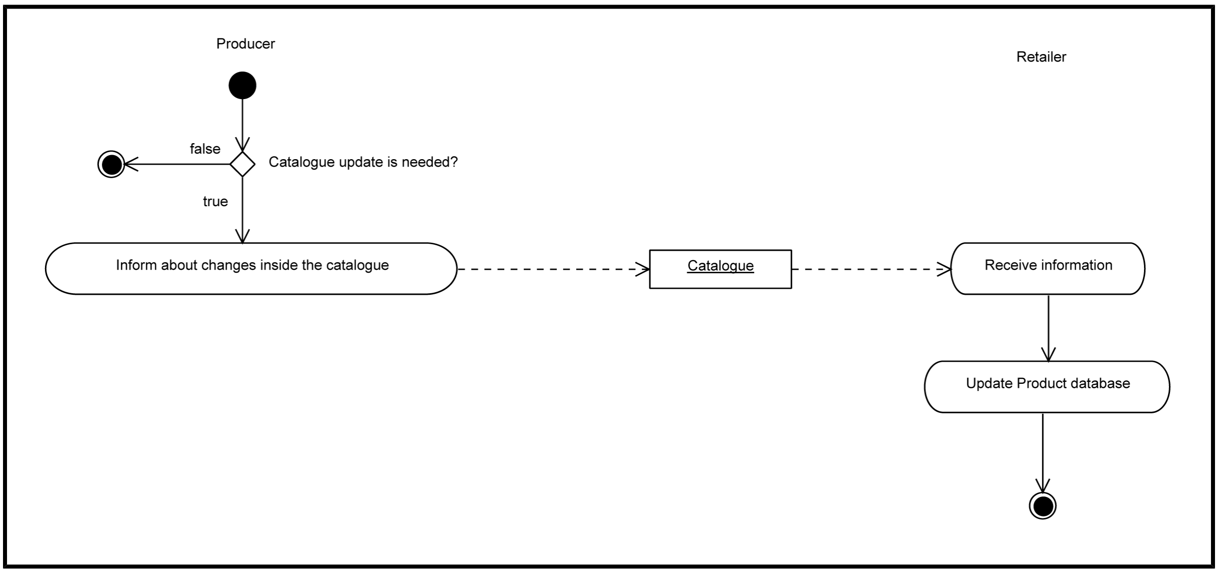
Figure 50. Synchronizing Stock Information



2.13.2.7. Changes to the Item Catalogue

In the event of a change, either inside an item belonging to the CRP catalogue or the relationship of an item to the CRP catalogue, information about the change is sent to the retailer.

Figure 51. Changes to the Item Catalogue



2.13.3. Replenishment On Customer Demand

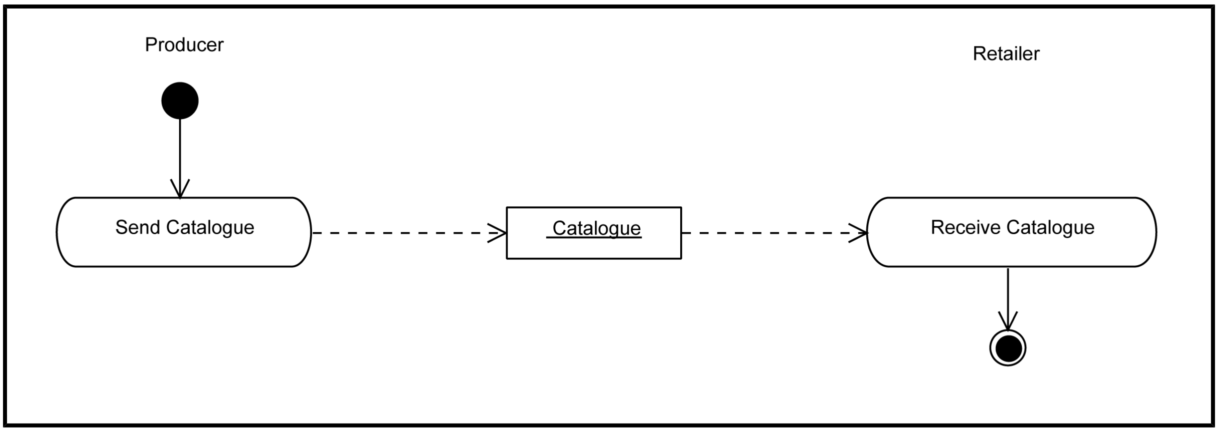
Another variant of VMI is Replenishment On Customer Demand. In this process, the producer selects a subset of its products for a specific retailer and sends out the related article catalogue. Then the producer periodically sends information about the availability of items so that the retailer can form the best ordering plan. The replenishment periodically happens on retailer (customer) demand, and unlike the case with CRP (above), the producer is allowed to propose changes to the orders. Also, because of the requirement to update item availability information, an additional document type (Stock Availability Report) is added to the process.

The processes of sales and inventory reporting, invoicing, stock synchronization, and changing the catalogue are identical to the same processes in CRP. As with CRP, a report of sales and inventory movement at all retail locations is sent to the producer at the end of every sales period. Invoicing and logistics are normally charge-on-delivery but can also be based on a consignment/concession model.

2.13.3.1. Transfer of Base Article Catalogue

The producer publishes a catalogue of its products to the retailer. The catalogue can include basic articles, never-out-of-stock (NOS) articles, seasonal articles, short-season-collection articles, or seasonal NOS articles.

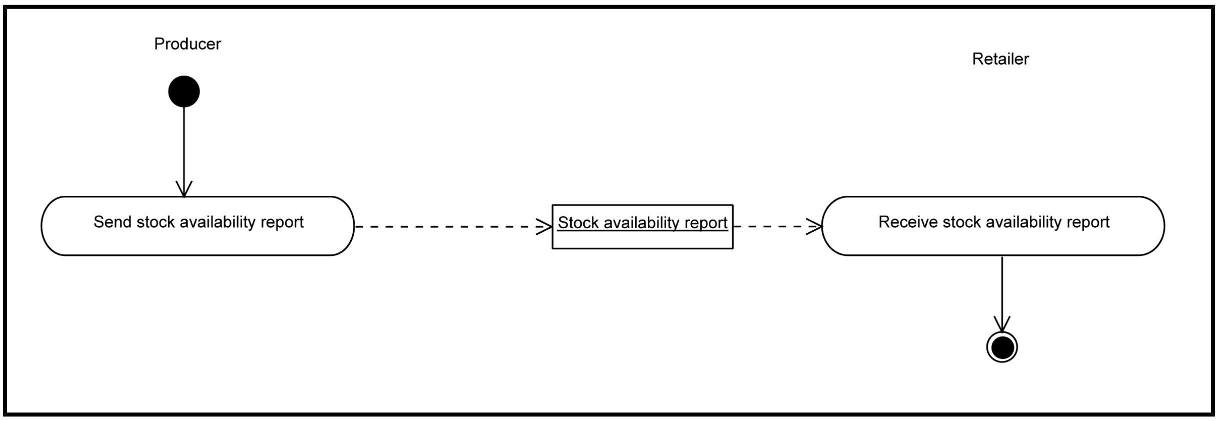
Figure 52. Transfer of Base Article Catalogue



2.13.3.2. Periodic Transfer of Article Availability Information

The producer sends out information about availability of goods (quantities on hand, quantities incoming, articles out of stock).

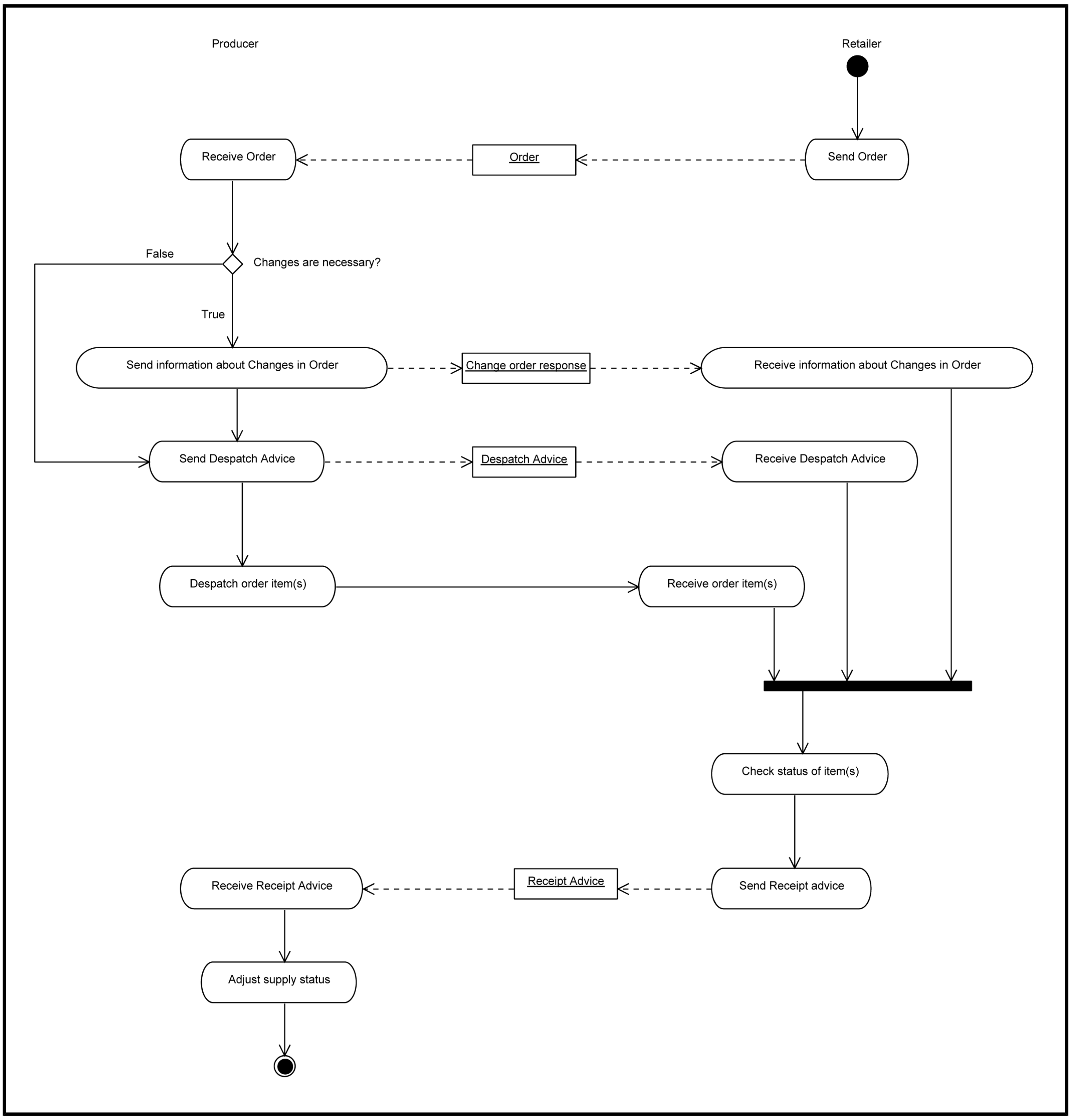
Figure 53. Periodic Transfer of Article Availability Information


Producer
Retailer
Send stock availability report
Stock availability report
Receive stock availability report

2.13.3.3. Initial Stocking of the Area by Producer and Retailer

At the beginning of the business cooperation —or perhaps at the beginning of a season, if seasonal NOS (never out of stock) products are the focus—the retailer orders its base stock and the products are delivered. Note that the producer is allowed to propose changes to the order (compare this figure with Figure 47, “Initial Stocking of the Area by Retailer”).

Figure 54. Initial Stocking of the Area by Producer and Retailer



2.13.3.4. Periodic Replenishment

Periodically, the retailer's system calculates the quantities needed for replenishment of the area. From the result, an order is sent, and the producer responds with a direct delivery within 48 hours.

The replenishment process uses the same documents in the same order as the Initial Stocking process, so the duplicate diagram is omitted here; see Figure 54, “Initial Stocking of the Area by Producer and Retailer”. It must be remembered, however, that the two processes are taking place at different points in time, so their pre and post conditions will be different.

2.13.3.5. Report of Sales and Inventory Movement

Sales and inventory movement information is transferred daily from the retailer to the producer.

The process for sales and inventory reporting is the same as in CRP (see Figure 48, “Report of Sales and Inventory Movements”).

2.13.3.6. Invoicing for Replenishment On Customer Demand

An invoice is sent either on a delivery or a sales basis.

The invoice process for Replenishment On Customer Demand is the same as for CRP (see Figure 49, “Invoicing for Cyclic Replenishment Program”).

2.13.3.7. Synchronizing Stock Information

Information about the actual stock is synchronised periodically (for example, every one to three months). Synchronization occurs at least once a year together with a physical inventory.

The stock synchronization process for Replenishment On Customer Demand is the same as in CRP (see Figure 50, “Synchronizing Stock Information”).

2.13.3.8. Changes to the Article Catalogue

In the event of a change, either inside an item belonging to the catalogue or the relationship of an item to the catalogue, information about the change is sent to the retailer.

The process for changing the catalogue in Replenishment On Customer Demand is the same as in CRP (see Figure 51, “Changes to the Item Catalogue”).

2.14. International Freight Management

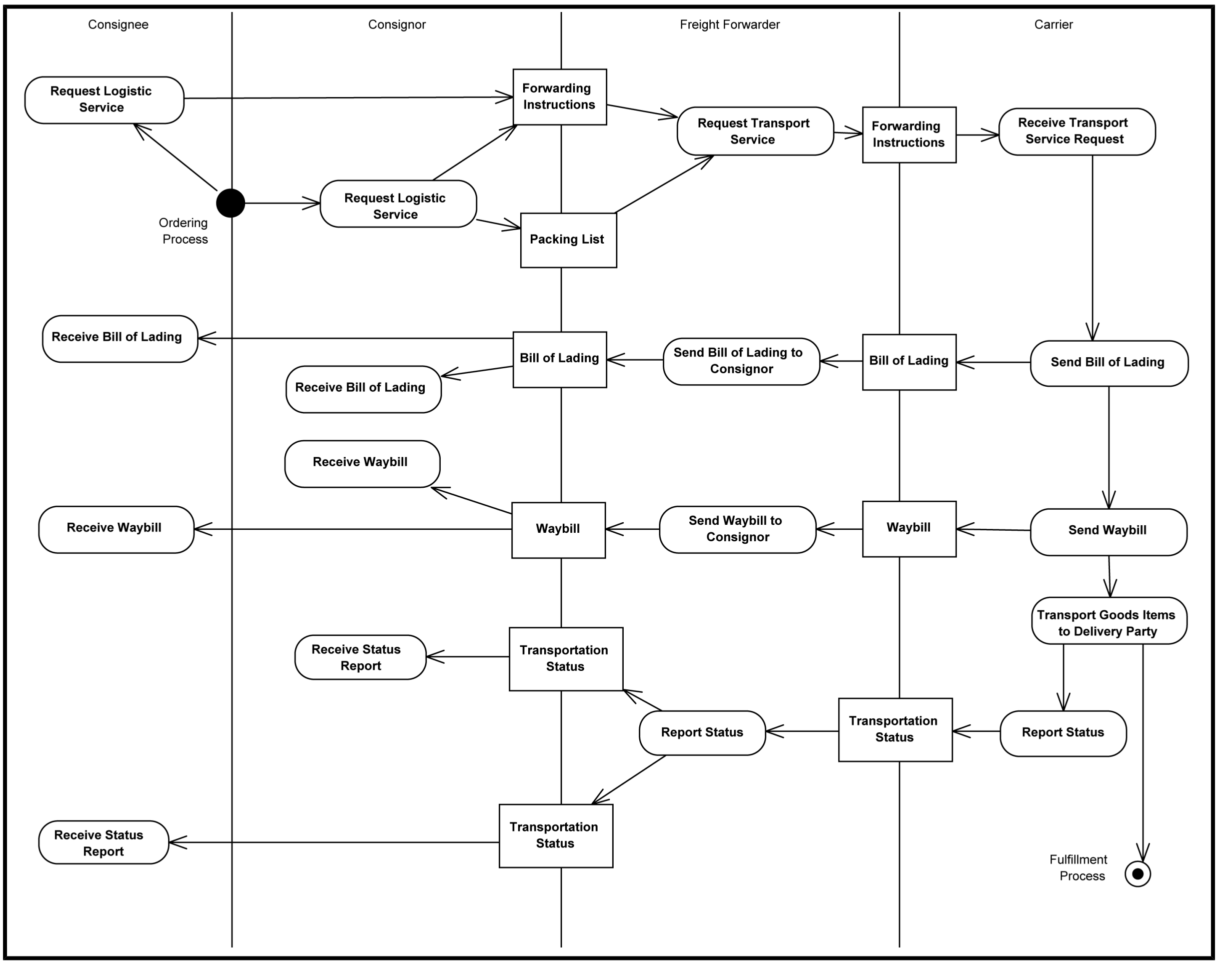
Freight management for domestic trade is typically accomplished using DespatchAdvice and ReceiptAdvice. The additional processes shown below are engineered to support the ordering and management of logistical services for international trade.

With receipt of an order and acknowledgement by the Supplier Party that the goods are available and ready to be shipped, the Consignor or Consignee initiates the transportation arrangements. This includes booking the consignment with a Transport Service Provider such as the Freight Forwarder or Carrier and advising the Delivery Party of the arrangements as needed.

Document types in these processes are Forwarding Instructions, Packing List, Waybill, and Bill of Lading.

It should be noted that these processes involve the Consignee and Consignor and do not cover all the logistical processes required to physically move the goods or regulatory notifications such as Customs declarations.

Figure 55. Initiate Freight Management Process



2.14.1. Forwarding Instructions

Forwarding Instructions are normally used by any party who gives instructions for the transportation services required for a consignment of goods (the Transport Service Buyer) to any party who is contracted to provide the transportation services (called the Transport Service Provider). Forwarding Instructions may also be used by any party who requests a booking of shipment space to be made for the transportation services required for a consignment of goods to any party who will provide the underlying transportation services. The parties who issue this document are commonly referred to as the shipper, consignee, or consignor, while the parties who receive this document are forwarders, carriers, shipping agents, etc.

Forwarding Instructions may also be issued by a freight forwarder or shipping agent in their capacity as a Transport Service Buyer. This document may be used to arrange for the transportation:

- Of different types of goods or cargoes
- Whether containerized or non-containerized
- Through different modes of transport, and
- From any origin to any destination.

2.14.2. Bill of Lading

A Bill of Lading is issued by the party who provides the physical transportation services (e.g., carrier) to the party who gives instructions for the transportation services (shipper, consignor, etc.) stating the details of the transportation, charges, and terms and conditions under which the transportation service is provided.

A Bill of Lading may also be issued by the party who acts as an agent for the carrier or other agents to the party who gives instructions for the transportation services (shipper, consignor, etc.) stating the details of the transportation, charges, and terms and conditions under which the transportation service is provided, but who does not provide the physical transportation service.

A Bill of Lading corresponds to the information on the Forwarding Instructions. It is used for ocean or river modes of transport.

A Bill of Lading may serve as a contractual document between the parties for the transportation service. The document evidences a contract of carriage by sea and the acceptance of responsibility for the goods by the carrier, by which the carrier undertakes to deliver the goods against surrender of the document. A provision in the document that the goods are to be delivered to the order of a named person, or to order, or to bearer, constitutes such an undertaking.

2.14.3. Waybill

A Waybill is issued by the party who provides the physical transportation services to the party who gives instructions for the transportation services (shipper, consignor, etc.). It states the details of the transportation, charges, and terms and conditions under which the transportation service is provided.

Unlike a Bill of Lading, a Waybill is not negotiable and cannot be assigned to a third party. It is issued as a cargo receipt and is not required to be surrendered at the destination in order to pick up the cargo. This simplifies the documentation procedures between a transport service buyer and a transport service seller.

2.14.4. Packing List

A Packing List is normally issued by the Consignor. It states the distribution of goods in individual packages.

Based on this detail, the party who provides the logistic services will make arrangement for the transportation of the goods.

2.15. Intermodal Freight Management

Intermodal transport implies the use of a combination of transport modes. Any support for the management of such chains has to support the modal shift of cargo flows from road to intermodal transport using road in combination with short sea shipping, inland waterways, and rail.

The Intermodal Freight Management process differs from conventional freight management in that it is generic and independent of transport mode. The focus is the multimodal transport chain as seen from the transport user's point of view. The transport user needs information about all the possible transport services that can be used to build a complete transport chain. If the choices to be made by the transport user are based upon the qualities of the transport services themselves, and not by which transport mode is used, the description of the transport services, and the exchanges of information about the roles and services must be simple and common.

The roles of the various Parties are defined as follows:

- The *Transport User* is the role representing anyone that needs to have cargo transported. The Transport User also provides the Transport Service Provider with instructions and detailed information about the transport items to be transported.

- The *Transport Service Provider* is the role that ensures the transport of the cargo from the origin to the destination. This includes the management of the transport services and the operation of the transport means and handling equipment. A Transport Service Provider may also provide administrative services required for moving the cargo, such as cargo inspection.

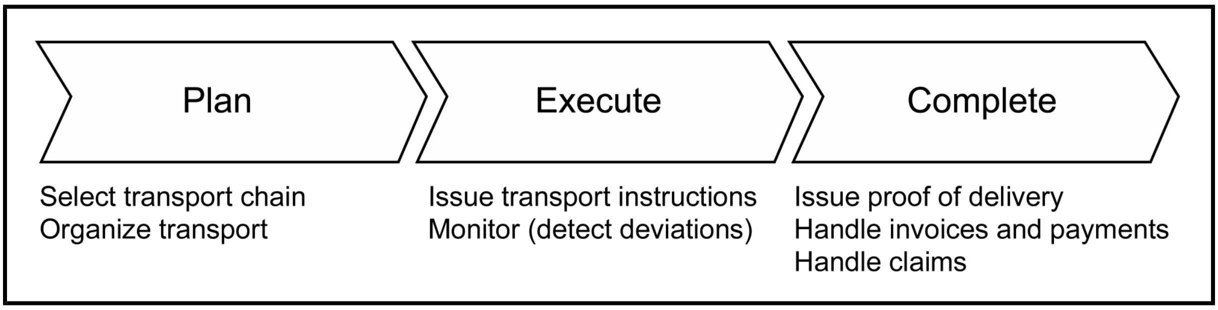
- The *Transportation Network Manager* is the role that extracts all information available regarding the infrastructure (static or dynamic) related to planning and executing transport and makes this information available to the Transport User and the Transport Service Provider.
- The *Transport Regulator* is the role that receives all mandatory reporting (and checks if reporting has been carried out) in order to ensure that all transport services are completed according to existing rules and regulations.

It should be noted that a person or organization may take on different roles. For example, a freight forwarder is, on the one hand, a Transport Service Provider when communicating with clients (Transport Users). On the other hand, the freight forwarder is a Transport User when acquiring services from subcontractors to ensure that a transport service is carried out between an origin and a destination.

The Intermodal Freight Management process takes place in three stages:

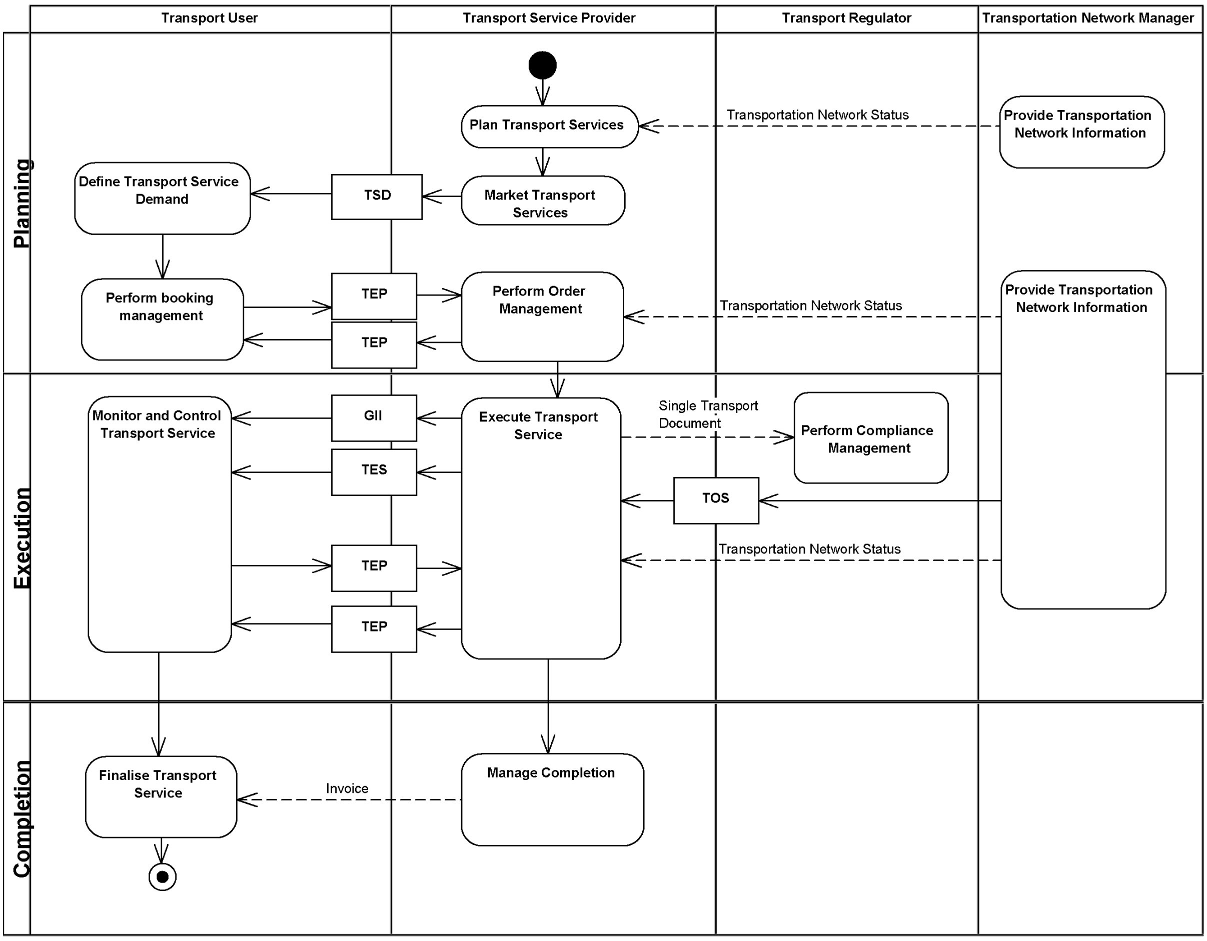
- *Planning*: Allows Transport Service Providers to advertise their services to Transport Users in a common format, in terms of a schedule and a freight rate published in a standard format. It allows Transport Users to provide a short list of potential services from which they can make a final choice by negotiating with potential Transport Service Providers. This is the “Plan” generic business process.
- *Execution*: Enables Transport Service Providers to manage the physical transport of the goods, exchanging information on the status of the shipment with the Transport Users and the status of the transport infrastructure with Transportation Network Managers; it also allows Transport Users and Transport Service Providers to exchange regulatory information with Transport Regulators (the “Execute” generic business process).
- *Completion*: Facilitates the issuing of proof of delivery and invoices between the Transport Service Provider and the Transport User (the “Complete” generic business process).

Figure 56. The Generic Intermodal Freight Management Process



These three stages are detailed in the following diagram.

Figure 57. The Intermodal Freight Management Process


Transport User
Transport Service Provider
Transport Regulator
Transportation Network Manager
Planning
Execution
Completion
Plan Transport Services
Transportation Network Status
Provide Transportation Network Information
Define Transport Service Demand
TSD
Market Transport Services
Perform booking management
TEP
TEP
Perform Order Management
Transportation Network Status
Provide Transportation Network Information
Monitor and Control Transport Service
GII
Execute Transport Service
Single Transport Document
Perform Compliance Management
TES
TOS
Transportation Network Status
TEP
TEP
Finalise Transport Service
Invoice
Manage Completion

2.15.1. Transport Execution Plan

The Transport Execution Plan (TEP) is a plan established between a Transport User and a Transport Service Provider. The process of establishing a TEP can be carried out after many interactions between the two roles, from the quotation stage up to the final agreement of the TEP.

The following diagrams show the actual transactions involved in both a basic scenario where the TEP is used to book and confirm a transport service and to send notification of a completed transport service (Figure 58, “Basic Transport Planning”) and a scenario involving an intermediary (Figure 59, “Transport Planning with Intermediary Role”).

Figure 58. Basic Transport Planning

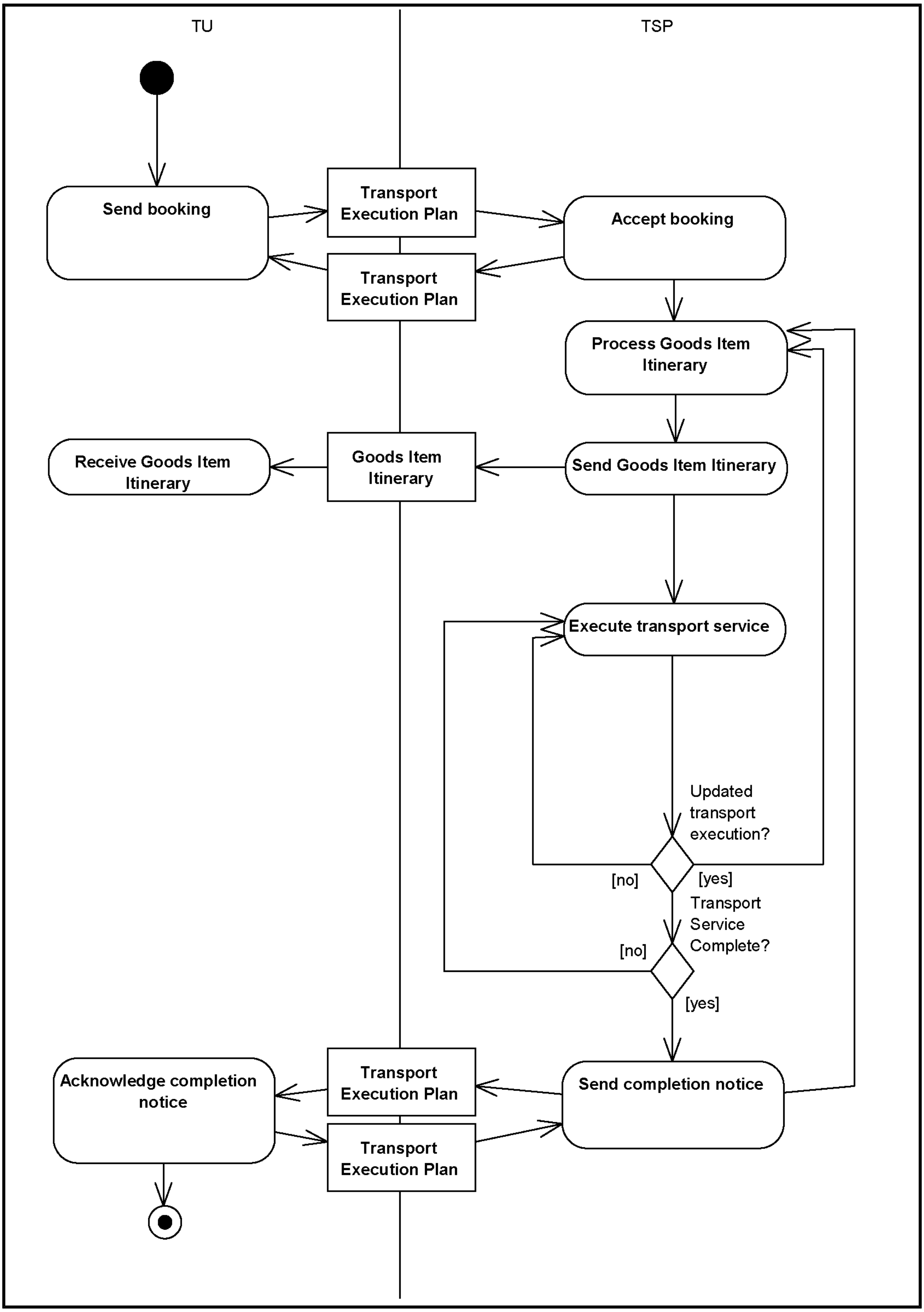
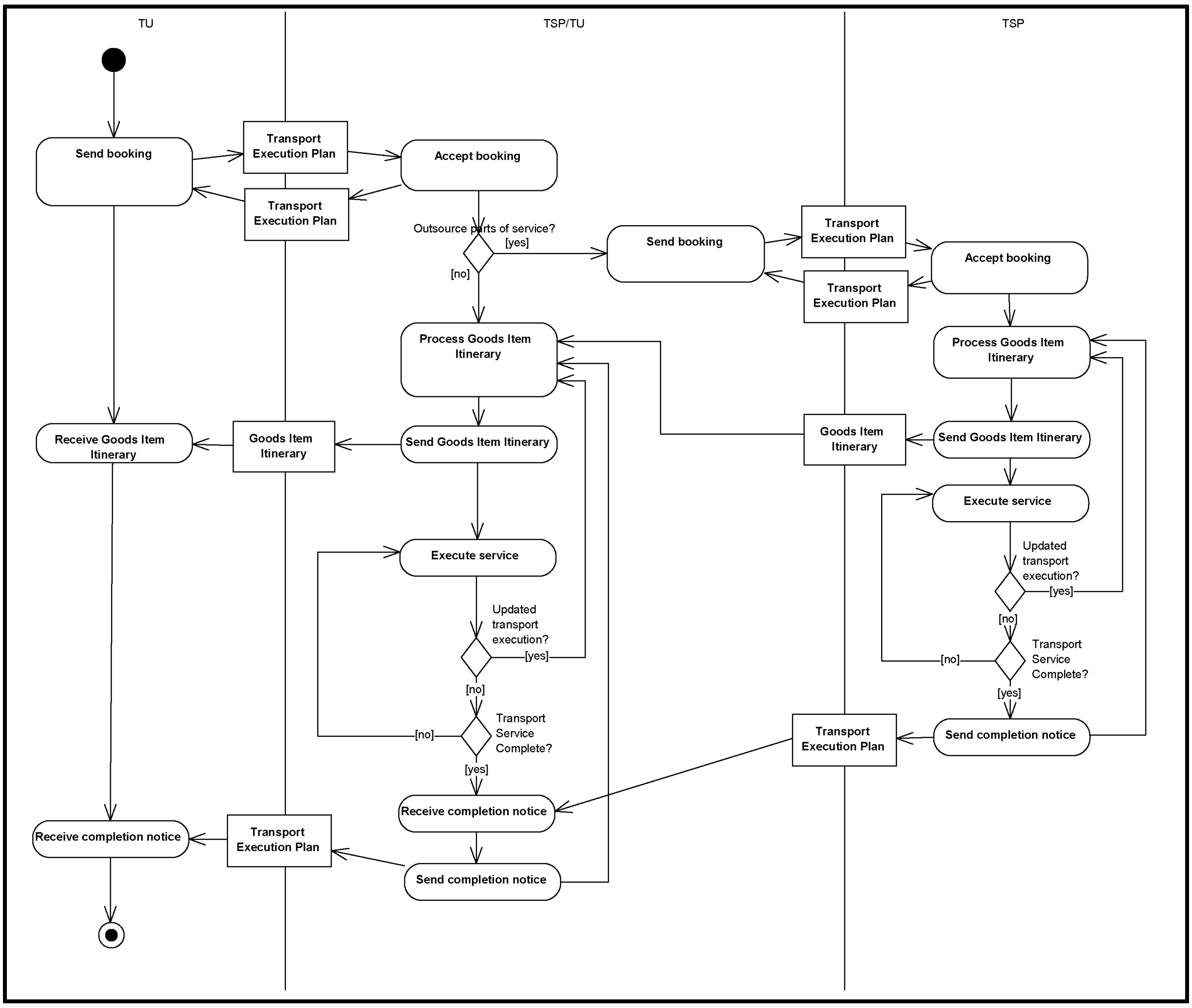

TU
TSP
Send booking
Transport Execution Plan
Accept booking
Transport Execution Plan
Process Goods Item Itinerary
Receive Goods Item Itinerary
Goods Item Itinerary
Send Goods Item Itinerary
Execute transport service
Updated transport execution?
[no]
[yes]
Transport Service Complete?
[no]
[yes]
Acknowledge completion notice
Transport Execution Plan
Send completion notice
Transport Execution Plan

Figure 59. Transport Planning with Intermediary Role

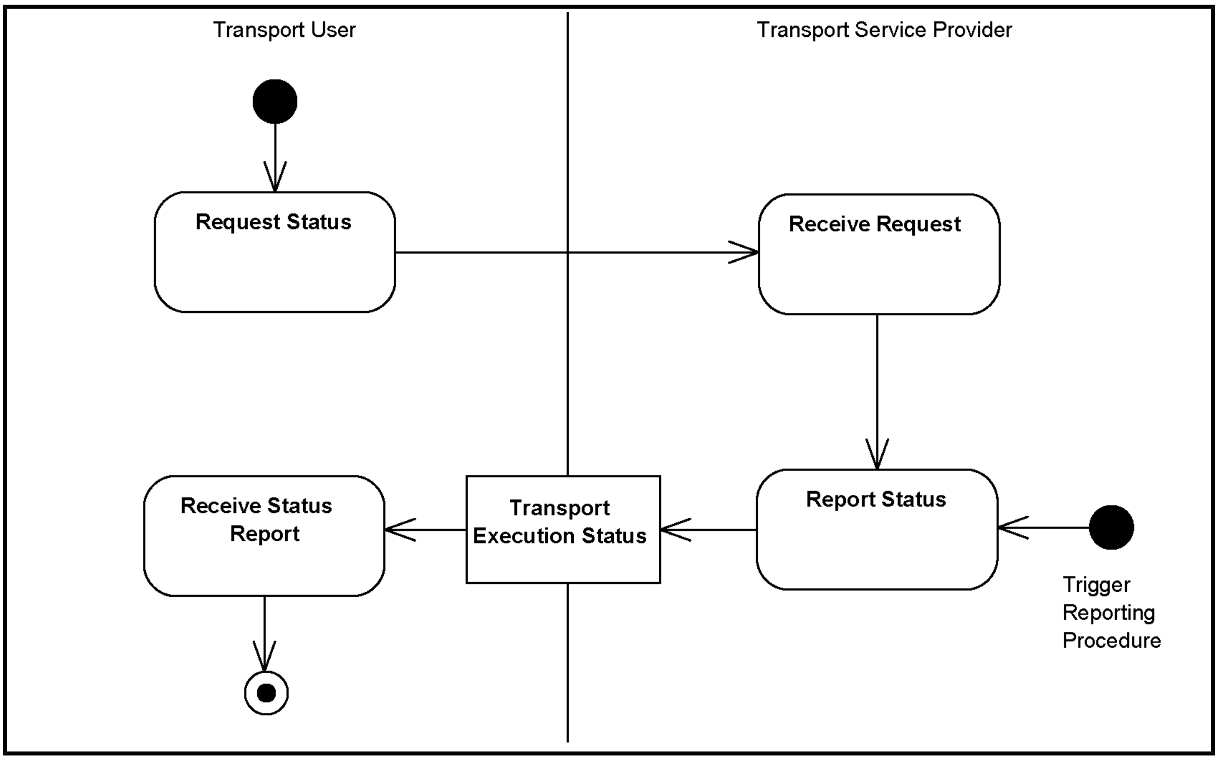


2.15.2. Transport Execution Status

The Transport Execution Status (TES) provides the status of the execution of a Transport Execution Plan. The TES must always refer to the Transport Execution Plan identifier and will return both the progress status of the transport execution and the condition status of the goods items being transported.

The Transport Execution Status may be sent from the Transport Service Provider as a response to a request, or it may be sent as a push operation based on trigger events.

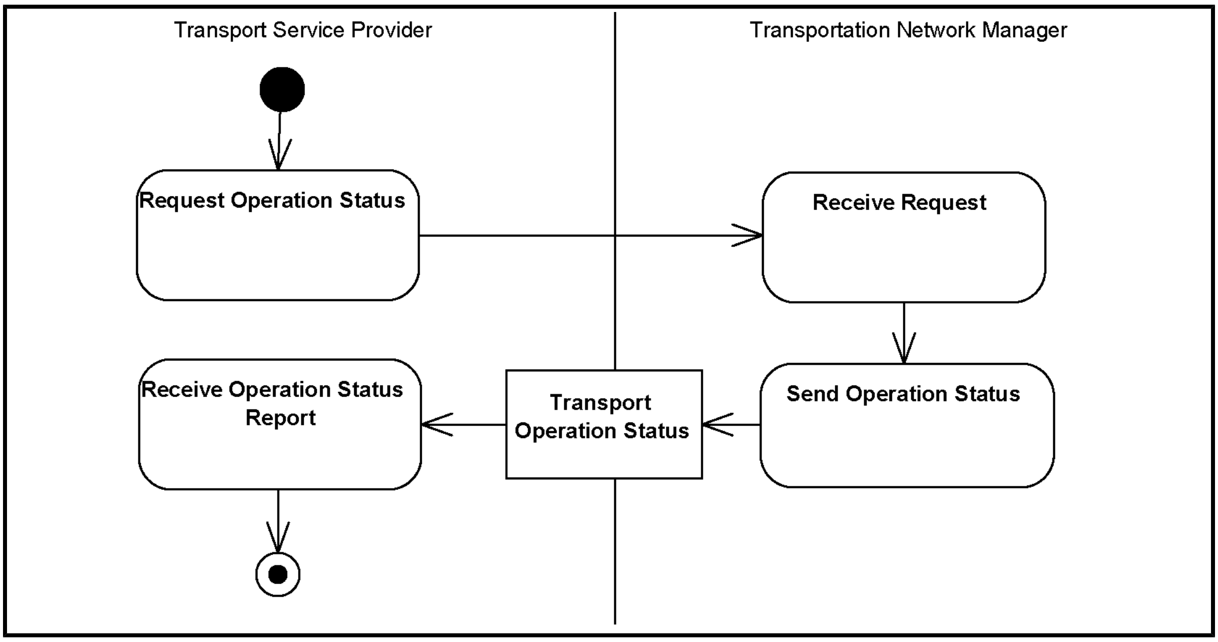
Figure 60. Transport Execution Status Process


Transport User
Transport Service Provider
Request Status
Receive Request
Receive Status Report
Transport Execution Status
Report Status
Trigger Reporting Procedure

2.15.3. Transport Operation Status

The Transport Operation Status (TOS) collects and reports information about the status of the transport means. The Transport Service Provider requests the Transportation Network Manager to provide status information related to a specific transport vehicle, using the vehicle identification number. The Transportation Network Manager then provides information about the location and time schedule status to the Transport Service Provider. The most typical use of TOS is to ask assistance from the Transportation Network Manager when estimated times of arrival are established. Reporting on the status of the goods themselves is covered by the Transport Execution Status and Freight Status Reporting process (Section 2.16, “Freight Status Reporting”).

Figure 61. Transport Operation Status Process

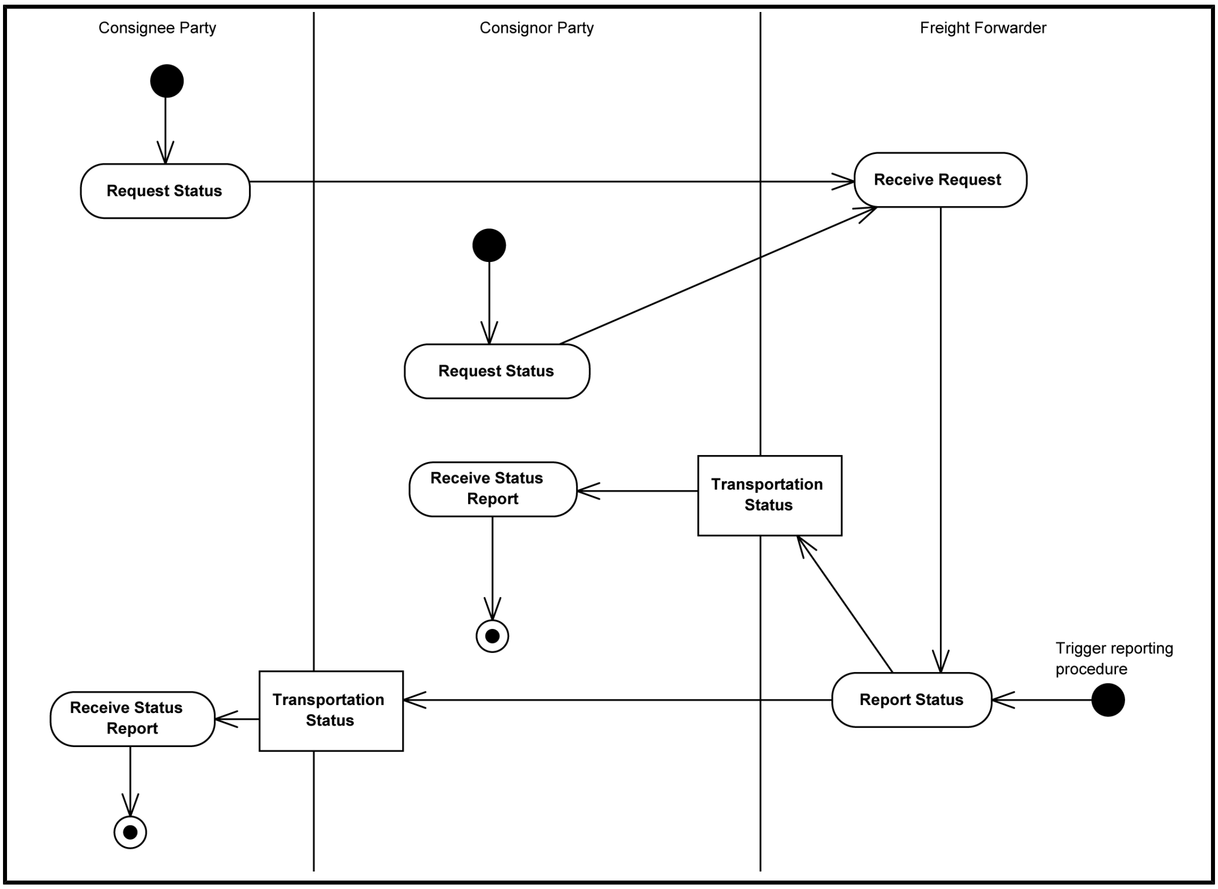

Transport Service Provider
Transportation Network Manager
Request Operation Status
Receive Request
Receive Operation Status Report
Transport Operation Status
Send Operation Status

2.16. Freight Status Reporting

Freight Status Reporting is the process by which a Freight Forwarder (also known as the Transport Service Provider) communicates the status of shipments currently under their management to the Consignee and/or Consignor (also known as the Transport Users).

A Transportation Status document is provided by the Freight Forwarder, either through an individual specific request or through an agreed status reporting procedure.

Figure 62. Freight Status Reporting Process

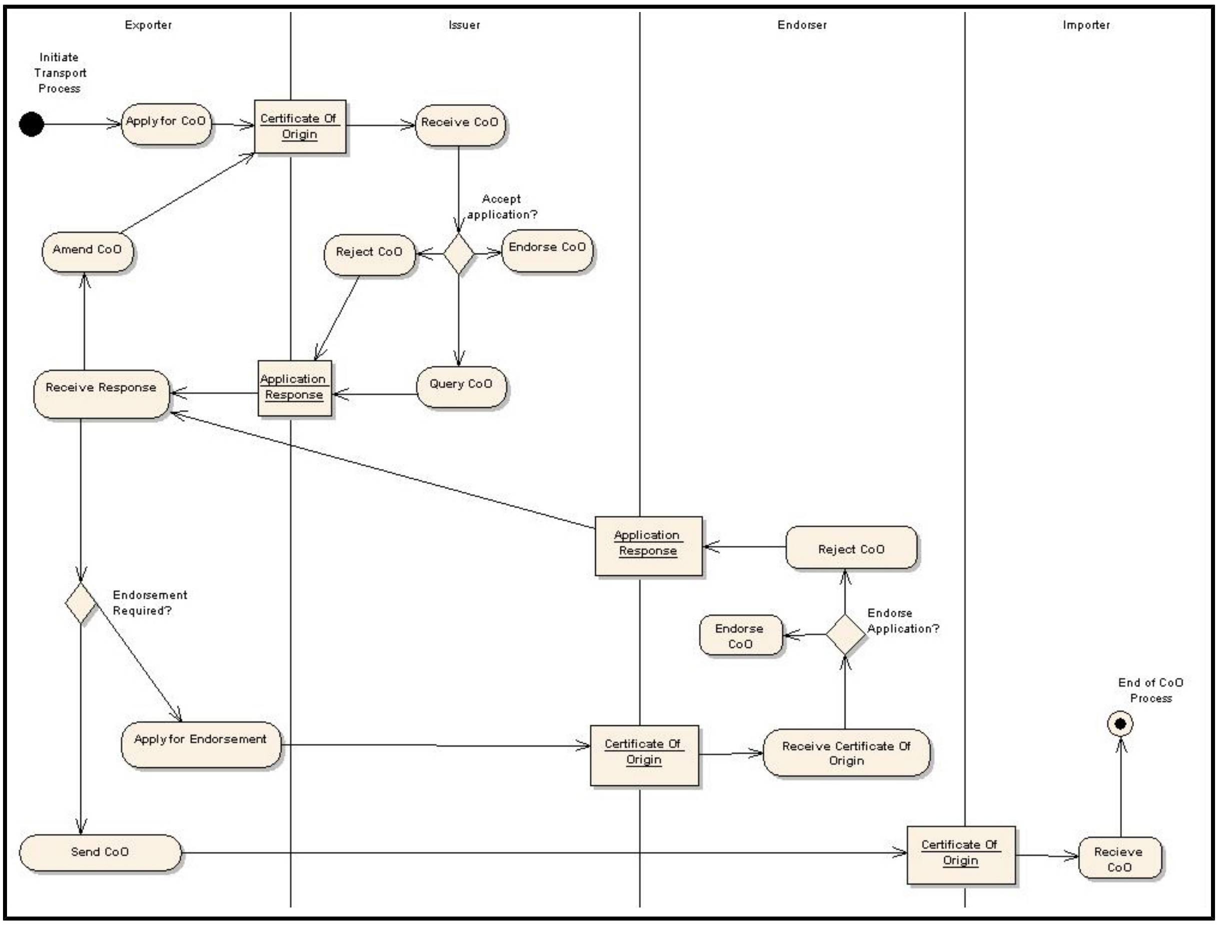


2.17. Certification of Origin of Goods

When an Exporter ships certain goods they may be required to attest to the origin of the goods. A Certificate of Origin is a document required by governments declaring that goods in a particular international shipment are of a certain origin.

It is the responsibility of the Exporter to sign the Certificate of Origin document and submit it to a local chamber of commerce or designated government agency or board. These parties are the endorser and issuer of the Certificate of Origin. The Endorser must have access to other documents, such as the commercial invoice and Bill of Lading, in order to verify the Exporter’s claims that the goods originated in that country. Finally, the issued Certificate of Origin is sent to the Importer.

Figure 63. Certification of Origin of Goods Process



2.18. Document Types

To support these business processes UBL 2.1 defines various document types. The following table lists all the UBL 2.1 document types together with their target business processes and roles for parties who would typically submit and receive them.

Table 1. Summary of UBL 2.1 Document Types

Document Name	Description	Processes Involved	Submitter Role	Receiver Role
Application Response	A document to indicate the application s response to a transaction. This may be a business response and/or a technical response, sent automatically by an application or initiated by a user.	Any	Sender	Receiver
Attached Document	A UBL wrapper that allows a document of any kind to be packaged with the UBL document that references it.	Any	Sender	Receiver
Awarded Notification	The document used to communicate the contract award to the winner	Tendering		
Bill Of Lading	The Bill of Lading is issued by the party who acts as an agent for the carrier or other agents to the party who gives instructions for the transportation services (shipper, consignor, etc.) stating the details of the transportation, charges, and terms and conditions under which the transportation service is provided. The party issuing this document does not necessarily provide the physical transportation service. It corresponds to the information on the Forwarding Instruction. It is used for any mode of transport. A Bill of Lading can serve as a contractual document between the parties for the transportation service. The document evidences a contract of carriage by sea and the acceptance of responsibility for the goods by the carrier, and by which the carrier undertakes to deliver the goods against surrender of the document. A provision in the document that the goods are to be delivered to the order of a named person, or to order, or to bearer, constitutes such an undertaking.	Freight Management	Freight Forwarder, Carrier	Consignor (or Consignee), Freight Forwarder
Call For Tenders	The document used for a Contracting Party to define the procurement project to buy goods, services or works during an specified period.	Tendering	Contracting Authority	Tenderer
Catalogue	The document that describes items, prices, and price validity.	Catalogue	Seller	Contracting Party
Catalogue Deletion	The document used to cancel an entire Catalogue.	Catalogue	Seller	Contracting Party
Catalogue Item Specification Update	The document used to update information about Items (e.g., technical descriptions and properties) on an existing Catalogue.	Catalogue	Seller	Contracting Party
Catalogue Pricing Update	The document used to update information about prices on an existing Catalogue.	Catalogue	Seller	Contracting Party
Catalogue Request	The document used to request a Catalogue.	Catalogue	Contracting Party	Seller
Certificate Of Origin	A document that describes the Certificate of Origin.	Certification of Origin of Goods	Exporter, Issuer	Issuer, Importer
Contract Award Notice	The document published by a Contracting Party to announce the awarding of a procurement project.	Tendering	Contracting Authority	Tenderer
Contract Notice	The document used for a Contracting Party to announce the project to buy goods, services or works.	Tendering	Contracting Authority	Tenderer
Credit Note	The document used to specify credits due to the Debtor from the Creditor.	Billing	Supplier Accounting Party	Customer Accounting Party
Debit Note	The document used to specify debits made by the Debtor.	Billing	Customer Accounting Party	Supplier Accounting Party
Despatch Advice	The document used to describe the despatch or delivery of goods and services.	Fulfilment	Despatch	Delivery

Document Name	**Description**	**Processes Involved**	**Submitter Role**	**Receiver Role**
Document Status	A document used to provide information about document status.	Any collaboration	Party currently controlling Status of the collaboration	Party requesting Status on collaboration
Document Status Request	A document used to request the status of another document.	Any collaboration	Party requesting Status on collaboration	Party currently controlling Status of the collaboration
Exception Criteria	Used to specify basic information about the content of the message including version number, creation date and time.	Collaborative Planning, Forecasting and Replenishment		
Exception Notification	The document used to notify an exception	Collaborative Planning, Forecasting and Replenishment		
Forecast	The document used to specify a forecast.	Collaborative Planning, Forecasting and Replenishment		
Forecast Revision	The document used to revise a Forecast.	Collaborative Planning, Forecasting and Replenishment		
Forwarding Instructions	The document issued to a forwarder, giving instructions regarding the action to be taken for the forwarding of goods described therein. Forwarding Instructions is used by any party who gives instructions for the transportation services required for a consignment of goods to any party who is contracted to provide the transportation services. The parties who issue this document are commonly referred to as the shipper or consignor, while the parties who receive this document are forwarders, carriers, shipping agents, etc. Note that this document may also be issued by a forwarder or shipping agent in their capacity as a shipper . This document can be used to arrange for the transportation (1) of different types of goods or cargoes; (2) whether containerized or non-containerized; (3) through different modes of transport including multi-modal; and (4) from any origin to any destination.	Freight Management	Consignor (or Consignee), Freight Forwarder	Freight Forwarder, Carrier
Freight Invoice	A document stating the charges incurred for the logistics service.	Freight Billing	Freight Forwarder	Consignor or Consignee
Guarantee Certificate	A document to notify the deposit of a guarantee.	Tendering	Tenderer	Contracting Authority
Instruction For Returns	This document is used to initiate a return of goods. The producer is requesting products which are badly sold either for use in other places or just to free the area from it.	Cyclic Replenishment Program		
Inventory Report	Report about the quantities of each item which are or will be on stock.	Cyclic Replenishment Program		
Invoice	The document used to request payment.	Billing	Supplier Accounting Party	Customer Accounting Party

Document Name	Description	Processes Involved	Submitter Role	Receiver Role
Item Information Request	The document used to request product activity, forecast, or performance data.	Collaborative Planning, Forecasting and Replenishment		
Order	The document used to order goods and services.	Ordering	Buyer	Seller
Order Cancellation	The document used to cancel an entire Order.	Ordering, Fulfilment	Buyer	Seller
Order Change	The document used to specify changes to an existing Order.	Ordering, Fulfilment	Buyer	Seller
Order Response	The document used to indicate detailed acceptance or rejection of an Order or to make a counter-offer.	Ordering	Seller	Buyer
Order Response Simple	The document used to indicate simple acceptance or rejection of an entire Order.	Ordering	Seller	Buyer
Packing List	A document stating the detail of how goods are packed.	Freight Management	Consignor	Freight Forwarder
Performance History	Performance History represents a collection of values gathered for key performance metrics in the trading partner relationship.	Cyclic Replenishment Program		
Prior Information Notice	The document used for a Contracting Party to declare the intention to buy goods, services or works during an specified period.	Tendering		
Product Activity	Product activity represents movement of a product through a location in terms of the base unit of measure for the item.	Cyclic Replenishment Program		
Quotation	The document used to quote for the provision of goods and services.	Quotation	Seller	Originator
Receipt Advice	The document used to describe the receipt of goods and services.	Fulfilment	Delivery	Despatch
Reminder	The document used to remind the customer of payments overdue.	Billing	Supplier Accounting Party and/or Payee	Customer Accounting Party and/or Payee
Remittance Advice	The document used to specify details of an actual payment.	Payment	Supplier Accounting Party and/or Payee	Customer Accounting Party and/or Payee
Request For Quotation	The document used to request a Quotation for goods and services from a Seller.	Quotation	Originator	Seller
Retail Event	The document used to specify basic information about the content of the Retail Event Meassage message including version number, creation date and time.	Cyclic Replenishment Program		
Self Billed Credit Note	The Credit Note created by the Debtor in a Self Billing arrangement with a Creditor; Self Billed Credit Note replaces Debit Note in such arrangements.	Billing	Customer Accounting Party	Supplier Accounting Party
Self Billed Invoice	The Invoice document created by the Customer (rather than the Supplier) in a Self Billing relationship.	Billing	Customer Accounting Party	Supplier Accounting Party
Statement	The document used to specify the status of Orders, Billing, and Payment. This document is a Statement of Account and not intended as a summary Invoice	Billing	Supplier Accounting Party	Customer Accounting Party
Stock Availability Report	Report about the quantities of each item which are or will be on stock.	Cyclic Replenishment Program		

Document Name	Description	Processes Involved	Submitter Role	Receiver Role
Tender	A message which a tenderer offers a tender to the tendering organization for bid.	Tendering	Tenderer	Contracting Authority
Tenderer Qualification	A document used for the Tenderer to declare things about his own condition.	Tendering	Tenderer	Contracting Authority
Tenderer Qualification Response	A message which the procurement organization sends to an economic operator in order to notify its admision or exclusion to/from the tendering process	Tendering	Contracting Authority	Tenderer
Tender Receipt	A message sent by the Contracting Party to an Economic Operator in order to notify the reception of the tendering offer	Tendering	Contracting Authority	Tenderer
Trade Item Location Profile	This document is used to send trade item attributes which are focused on replenishment policies.	Collaborative Planning, Forecasting and Replenishment		
Transportation Status	A message to report the transport status and/or change in the transport status (i.e. event) between agreed parties.	Freight Status Reporting	Freight Forwarder	Consignee, Consignor
Transport Execution Plan	A document which is used in the negotiation of a transport service between a transport user and a transport service provider	Intermodal Freight Management		
Transport Execution Status	The Transport Execution Status is used to provide the transport user with timing and condition status information about the transport operation	Intermodal Freight Management		
Transport Operation Status	A document being sent from Transportation Network Manager to Transport Service Provider giving a status on the transport operation	Intermodal Freight Management		
Unawarded Notification	The document used to communicate the contract has been awarded to another tenderer	Tendering	Contracting Authority	Tenderer
Utility Statement	The Utility Statement contains information on the consumption of services provided by utility suppliers to private and public customers. These utilities include electricity, gas, water and telephony services. The Utility Statement is therefore a supplement to an Invoice or CreditNote.	Utility Billing	Supplier Accounting Party	Customer Accounting Party
Waybill	The Waybill is issued by the party who acts as an agent for the carrier or other agents, to the party who gives instructions for the transportation services (shipper, consignor, etc.) stating the details of the transportation, charges, and terms and conditions under which the transportation service is provided. The party issuing this document could be a party other than that providing the physical transportation. It corresponds to the information on the Forwarding Instruction. It is used for all modes of transport. It can serve as a contractual document between the parties for the transportation service. The document made out by the carrier or on behalf of the carrier evidencing the contract for the transport of cargo.	Freight Management	Consignor (or Consignee), Freight Forwarder	Freight Forwarder, Carrier

2.19. Party Roles

In the UBL supply chain processes, two main actors, Customer and Supplier, represent the key organizations or

people involved in the processes. Each of these actors may play various roles. Some processes may also involve supplementary roles that may be provided by different parties.

The actual role undertaken is dependent on the context of use. For example, the Despatch Party and Delivery Party as applied to the Procurement process may differ in the Transportation process. In other words, whether the Consignor in a Transportation process is actually equal to the Despatch Party or Seller in the Procurement process depends on the specific circumstances.

The following table contains a description of the typical roles for the actor known as Party. Note that some roles require an extension of the information entities required. In UBL 2.1, the following are roles that extend the Party structure: Customer Party, Supplier Party, Contracting Party, Endorser Party, and Qualifying Party.

Table 2. Party Roles

Actor	Role	Description	Example	Synonyms	Sends	Receives
Customer Party	Originator	The party that had the original demand for the goods and/or services and therefore initiated the procurement transaction. The Originator participates in pre-ordering activity either through RFQ and Quotation or by receiving a Quotation as a response to a punchout transaction on a marketplace or Seller's website. If the Originator subsequently places an Order, the Originator adopts the role of Buyer. The Originator is typically the contact point for queries regarding the original requirement and may be referred to in an Order Change, Order Cancellation, or Order Response.	If an employee requests a computer, the employing company may become the Buyer, but the employee is the Originator. They need to receive information about the order.		Request for Quotation	Quotation
Customer Party	Buyer	The party that purchases the goods or services on behalf of the Originator. The Buyer may be referred to in Order Response, Despatch Advice, Invoice, Self Billed Invoice, Credit Note, and Account Statement.	A company may delegate the task of purchasing to a specialized group to consolidate orders and gain greater discounts.	Order Point	Order, Order Change, and Order Cancellation	Order Response
Customer Party	Delivery	The party to whom goods should be delivered. The Delivery Party may be the same as the Originator. The Delivery Party must be referred to at line item level in RFQ, Quotation, Order, Order change, Order Cancellation, and Order Response. The Delivery Party may be referred to at line level in Invoice, Self Billed Invoice, Credit Note, and Debit Note. The Delivery Party may be stipulated in a transport contract.	If a municipality buys a wheelchair for a citizen, the wheelchair must be delivered to the citizen (the Delivery Party). In such cases the citizen may be notified before delivering the wheelchair.	Delivery Point, Destination Party, Receiver, Recipient	Receipt Advice	Despatch Advice
Customer Party	Accounting Customer	The party responsible for making settlement relating to a purchase and resolving billing issues using a Debit Note. The Accounting Customer must be referred to in an Order and may be referred to in an Order Response. In a Self Billing scenario, the Accounting Customer is responsible for calculating and issuing tax invoices.	If a kindergarten buys some toys they may be the Originator, Buyer, and Delivery Party, but the municipality may play the role of Accounting Customer—they are going to pay for it.	Invoicee, Accounts Payable, Debtor	In a traditional Billing scenario: Debit Note, Account Response, and Remittance Advice In a Self Billing scenario: Self Billed Invoice, Self Billed Credit Note, and Remittance Advice	In a traditional Billing scenario: Invoice, Credit Note, and Statement of Account In a Self Billing scenario: Credit Note, Account Response, and Statement of Account

Actor	Role	Description	Example	Synonyms	Sends	Receives
Supplier Party	Seller	The party responsible for handling Originator and Buyer services. The Seller party is legally responsible for providing the goods to the Buyer. The Seller party receives and quotes against RFQs and may provide information to the Buyer's requisitioning process through Catalogues and Quotations.	The organization that sells wheelchairs to municipalities.	Sales Point, Provider, Customer Manager	Quotation, Order Response, Order Response Simple, Catalogue, Catalogue Deletion, Catalogue Item Specification Update, Catalogue Pricing Update	RFQ, Order, Order Change, Order Cancellation, Request for Catalogue
Supplier Party	Despatch	The party where goods are to be collected from. The Despatch Party may be stipulated in a transport contract.	The wheelchair Supplier may store chairs at a local warehouse. The warehouse will actually despatch the chair to the Delivery Party. The local warehouse is then the Despatch Party.	Despatch Point, Shipper, Sender	Despatch Advice	Receipt Advice
Supplier Party	Accounting Supplier	The party who claims the payment and is responsible for resolving billing issues and arranging settlement.	There are cases where the Accounting Supplier is not the Seller party. For example, factoring, where the invoicing is outsourced to another company.	Accounts Receivable, Invoice Issuer, Creditor	In a traditional Billing scenario: Invoice, Credit Note, and Statement of Account In a Self Billing scenario: Credit Note, Account Response and Statement of Account	In a traditional Billing scenario: Debit Note, Account Response, and Remittance Advice In a Self Billing scenario: Self Billed Invoice, Self Billing Credit Note, and Remittance Advice
Supplier Party	Payee	The party to whom the Invoice is paid.	The Accounting Supplier may not be the party to be paid due to changes in the organization, e.g., a company merger.	Accounts Receivable, Creditor		Remittance Advice
Customer Party	Contractor	The party responsible for the contract to which the Catalogue relates.	An organization has a central office for maintaining catalogues of approved items for purchase.	Central Catalogue Party, Purchasing Manager	Request for Catalogue	Catalogue, Catalogue Deletion, Catalogue Item Specification Update, Catalogue Pricing Update
Party	Provider	The party responsible for the integrity of the information provided about an item.	The manufacturer may publish and maintain the data sheets about a product.		Catalogue, Catalogue Deletion, Catalogue Item Specification Update, Catalogue Pricing Update	

Actor	Role	Description	Example	Synonyms	Sends	Receives
Party	Receiver	The party receiving a document. The party receiving a Catalogue. Catalogue items may never be ordered, so the recipient of the catalogue is not an Originator or a Buyer.	A marketplace may receive an Application Response.			Catalogue, Catalogue Deletion, Catalogue Item Specification Update, Catalogue Pricing Update, Application Response
Party	Sender	The party sending a document.	A marketplace may send an Application Response.		Application Response	
Party	Consignor	The party consigning the goods as stipulated in the transport contract. A Buyer, Delivery, Seller, or Despatcher Party may also play the role of Consignor. Also known as the Transport User. The Consignor may be stipulated in a transport contract.	The wheelchair Supplier may source from a local warehouse. The Freight Forwarder will collect the chair from the local warehouse, which is thus the Consignor. In this case, the warehouse also plays the role of Despatch Party to the Freight Forwarder.	Despatch Point, Shipper, Sender, Transport User	Forwarding Instructions, Packing List	Bill of Lading, Waybill, Freight Invoice, Transportation Status
Party	Consignee	The party receiving a consignment of goods as stipulated in the transport contract.	The party taking responsibility for the receipt of the consignment covering the wheelchair.	Delivery Point, Transport Service Buyer	Forwarding Instructions, Freight Invoice	Bill of Lading, Waybill, Freight Invoice, Transportation Status
Party	Freight Forwarder	The party arranging the carriage of goods, including connected services and/or associated formalities, on behalf of a Consignor or Consignee. Also known as the Transport Service Provider. The Freight Forwarder may also be the Carrier. The Freight Forwarder may create an invoice and bill to the Transport Service Buyer for the transportation service provided.	The Consignor may have a contract with this Freight Forwarder, which is a Transport Services Provider, to arrange all their transport needs.	Shipping Agent, Broker, Courier, Transport Service Provider	Forwarding Instructions, Freight Invoice, Transportation Status	Bill of Lading, Waybill, Packing List
Party	Carrier	The party providing physical transport services.	The Freight Forwarder may engage an airline company to deliver the wheelchair. The airline is then the Carrier and delivers the chair to the Delivery Party.	Freight Haulier, Shipper, Ships Agent, Shipping Company, Airline, Rail Operator, Road Haulier	Bill of Lading, Waybill	Forwarding Instructions

Actor	Role	Description	Example	Synonyms	Sends	Receives
Party	Exporter	The party who makes regulatory export declarations, or on whose behalf regulatory export declarations are made, and who is the owner of the goods or has similar right of disposal over them at the time when the declaration is accepted.	The wheelchair Supplier has to apply for a Certificate of Origin in order to sell the chairs overseas.	Seller, Consignor	Certificate of Origin	Application Response
Party	Endorser	The party appointed by the Government of a country who has the right to certify a Certificate of Origin. This endorsement restricts goods imported from certain countries for political or other reasons.	The Government agency validates all the information provided by Exporter for Certificate of Origin approval.	Authorized Organization, Embassy	Certificate of Origin, Application Response	Certificate of Origin
Party	Importer	The party who makes, or on whose behalf an agent or other authorized person makes, an import declaration. This may include a person who has possession of the goods or to whom the goods are consigned.	A specialized group in a company consolidates the purchase request and handles the receiving of goods.	Order Point, Delivery Party, Buyer, Customer, Consignee		Certificate of Origin

3. UBL 2.1 Schemas

The UBL 2.1 XSD schemas are the only normative representations of the UBL 2.1 document types and library components.

All of the UBL 2.1 XSD schemas are contained in the `xsd` subdirectory of the UBL 2.1 release package (see Appendix A, *Release Notes (Informative)* for more information regarding the structure of the 2.1 release package and Section 3.3, "Schema Dependencies" for information regarding dependencies among the schema modules). The `xsd` directory is further subdivided into `xsd/maindoc` and `xsd/common` subdirectories.

For convenience in implementing the schemas, a parallel (and technically non-normative) "runtime" set with the annotation elements stripped out is provided in the `xsdrt` directory.

3.1. UBL Document Schemas

XSD schemas defining the 60 UBL 2.1 document types are located in the `xsd/maindoc` directory, as listed below.

ApplicationResponse
xsd/maindoc/UBL-ApplicationResponse-2.1.xsd

AttachedDocument
xsd/maindoc/UBL-AttachedDocument-2.1.xsd

AwardedNotification
xsd/maindoc/UBL-AwardedNotification-2.1.xsd

BillOfLading
xsd/maindoc/UBL-BillOfLading-2.1.xsd

CallForTenders
xsd/maindoc/UBL-CallForTenders-2.1.xsd

Catalogue
xsd/maindoc/UBL-Catalogue-2.1.xsd

CatalogueDeletion
xsd/maindoc/UBL-CatalogueDeletion-2.1.xsd

CatalogueItemSpecificationUpdate
xsd/maindoc/UBL-CatalogueItemSpecificationUpdate-2.1.xsd

CataloguePricingUpdate
xsd/maindoc/UBL-CataloguePricingUpdate-2.1.xsd

CatalogueRequest
xsd/maindoc/UBL-CatalogueRequest-2.1.xsd

CertificateOfOrigin
xsd/maindoc/UBL-CertificateOfOrigin-2.1.xsd

ContractAwardNotice
xsd/maindoc/UBL-ContractAwardNotice-2.1.xsd

ContractNotice
xsd/maindoc/UBL-ContractNotice-2.1.xsd

CreditNote
xsd/maindoc/UBL-CreditNote-2.1.xsd

DebitNote
xsd/maindoc/UBL-DebitNote-2.1.xsd

DespatchAdvice
xsd/maindoc/UBL-DespatchAdvice-2.1.xsd

DocumentStatus
xsd/maindoc/UBL-DocumentStatus-2.1.xsd

DocumentStatusRequest
xsd/maindoc/UBL-DocumentStatusRequest-2.1.xsd

ExceptionCriteria
xsd/maindoc/UBL-ExceptionCriteria-2.1.xsd

ExceptionNotification
xsd/maindoc/UBL-ExceptionNotification-2.1.xsd

Forecast
xsd/maindoc/UBL-Forecast-2.1.xsd

ForecastRevision
xsd/maindoc/UBL-ForecastRevision-2.1.xsd

ForwardingInstructions
xsd/maindoc/UBL-ForwardingInstructions-2.1.xsd

FreightInvoice
xsd/maindoc/UBL-FreightInvoice-2.1.xsd

GuaranteeCertificate
xsd/maindoc/UBL-GuaranteeCertificate-2.1.xsd

InstructionForReturns
xsd/maindoc/UBL-InstructionForReturns-2.1.xsd

InventoryReport
xsd/maindoc/UBL-InventoryReport-2.1.xsd

Invoice
xsd/maindoc/UBL-Invoice-2.1.xsd

ItemInformationRequest
xsd/maindoc/UBL-ItemInformationRequest-2.1.xsd

Order
xsd/maindoc/UBL-Order-2.1.xsd

OrderCancellation
xsd/maindoc/UBL-OrderCancellation-2.1.xsd

OrderChange
xsd/maindoc/UBL-OrderChange-2.1.xsd

OrderResponse
xsd/maindoc/UBL-OrderResponse-2.1.xsd

OrderResponseSimple
xsd/maindoc/UBL-OrderResponseSimple-2.1.xsd

PackingList
xsd/maindoc/UBL-PackingList-2.1.xsd

PerformanceHistory
xsd/maindoc/UBL-PerformanceHistory-2.1.xsd

PriorInformationNotice
xsd/maindoc/UBL-PriorInformationNotice-2.1.xsd

ProductActivity
xsd/maindoc/UBL-ProductActivity-2.1.xsd

Quotation
xsd/maindoc/UBL-Quotation-2.1.xsd

ReceiptAdvice
xsd/maindoc/UBL-ReceiptAdvice-2.1.xsd

Reminder
xsd/maindoc/UBL-Reminder-2.1.xsd

RemittanceAdvice
xsd/maindoc/UBL-RemittanceAdvice-2.1.xsd

RequestForQuotation
xsd/maindoc/UBL-RequestForQuotation-2.1.xsd

RetailEvent
xsd/maindoc/UBL-RetailEvent-2.1.xsd

SelfBilledCreditNote
xsd/maindoc/UBL-SelfBilledCreditNote-2.1.xsd

SelfBilledInvoice
xsd/maindoc/UBL-SelfBilledInvoice-2.1.xsd

Statement
xsd/maindoc/UBL-Statement-2.1.xsd

StockAvailabilityReport
xsd/maindoc/UBL-StockAvailabilityReport-2.1.xsd

Tender
xsd/maindoc/UBL-Tender-2.1.xsd

TenderReceipt
xsd/maindoc/UBL-TenderReceipt-2.1.xsd

TendererQualification
xsd/maindoc/UBL-TendererQualification-2.1.xsd

TendererQualificationResponse
xsd/maindoc/UBL-TendererQualificationResponse-2.1.xsd

TradeItemLocationProfile
xsd/maindoc/UBL-TradeItemLocationProfile-2.1.xsd

TransportExecutionPlan
xsd/maindoc/UBL-TransportExecutionPlan-2.1.xsd

TransportExecutionStatus
xsd/maindoc/UBL-TransportExecutionStatus-2.1.xsd

TransportOperationStatus
xsd/maindoc/UBL-TransportOperationStatus-2.1.xsd

TransportationStatus
xsd/maindoc/UBL-TransportationStatus-2.1.xsd

UnawardedNotification
xsd/maindoc/UBL-UnawardedNotification-2.1.xsd

UtilityStatement
xsd/maindoc/UBL-UtilityStatement-2.1.xsd

Waybill
xsd/maindoc/UBL-Waybill-2.1.xsd

3.2. UBL Common Schemas

The `xsd/common` directory contains schemas referenced by the document schemas in `xsd/maindoc`. Elements defined in the common schemas constitute a library of reusable business data components from which the UBL document schemas are assembled.

The name of each schema file together with a brief description of its contents is given below.

3.2.1. Reusable BIE Schemas

CommonBasicComponents
xsd/common/UBL-CommonBasicComponents-2.1.xsd

The CommonBasicComponents schema defines the global Basic Business Information Entities (BBIEs) that are used throughout UBL, serving, in effect, as a “global BBIE type database” for constructing documents. BBIEs are the “leaf nodes” of UBL documents, corresponding to individual data fields in traditional printed business forms.

CommonAggregateComponents
xsd/common/UBL-CommonAggregateComponents-2.1.xsd

The CommonAggregateComponents schema defines the Aggregate Business Information Entities (ABIEs) that are used throughout UBL, serving, in effect, as an “ABIE type database” for constructing the main documents.

3.2.2. Reusable Data Type Schemas

CCTS_CCT_SchemaModule
xsd/common/CCTS_CCT_SchemaModule-2.1.xsd

This schema provides Core Component Types as defined by [CCTS]. These types are used to construct higher-level data types in a standardized and consistent manner. This schema is defined by UN/CEFACT and should not be modified. It is imported by the UBL Unqualified Data Type Schema, and its types are the basis upon which UBL's unqualified data types are defined.

UnqualifiedDataTypes

xsd/common/UBL-UnqualifiedDataTypes-2.1.xsd

This schema defines Unqualified Data Types for BBIE definition. These types are derived from the Core Component Types in `CCTS_CCT_SchemaModule`. Where an unqualified type is not based solely on an XSD data type, all CCTS supplementary components are made available in the UBL UDT from the CCTS CCT.

QualifiedDataTypes

xsd/common/UBL-QualifiedDataTypes-2.1.xsd

[CCTS] permits the definition of Qualified Datatypes as derivations from CCTS-specified Unqualified Datatypes. In UBL 2.1, all data type qualifications are expressed in the [CVA] file cva/UBL-DefaultDTQ-2.1.cva. The UBL-QualifiedDataTypes-2.1.xsd file in the UBL 2.1 release is included among the schema modules imported by the Common Library and all document-level schema fragments in order to be consistent with the relationship of types in a CCTS framework, though the schema module itself has no declarations.

See Appendix E, *Data Type Qualifications in UBL (Informative)* for information regarding UBL 2.1 data type derivation.

3.2.3. Documentation Metadata Schema

CoreComponentParameters

xsd/common/UBL-CoreComponentParameters-2.1.xsd

The CoreComponentParameters schema defines the structure of the annotation/documentation sections that appear in all the other schemas, providing a consistent format for metadata such as object class, representation terms, semantic descriptions, and other supplementary information.

While not required by UBL schemas, this module is provided to encourage consistency in the documentation of customized extensions.

3.2.4. Extension Content Schemas

UBL extensions enable the validation of user-defined additions to the standard schemas, which are sometimes needed to satisfy legal requirements and can perform other useful functions as well. UBL 2.1 schemas are supplied with a predefined standard extension that supports advanced digital signatures; see Section 5, “UBL XAdES Extension for Digital Signatures” and Section 3.3, “Schema Dependencies”. For further information regarding the UBL extension mechanism, see [Customization].

CommonExtensionComponents

xsd/common/UBL-CommonExtensionComponents-2.1.xsd

The CommonExtensionComponents schema defines the extension structures that are used in all UBL document types, providing metadata regarding the use of an extension embedded in a UBL document instance.

ExtensionContentDatatype

xsd/common/UBL-ExtensionContentDataType-2.1.xsd

The ExtensionContentDataType schema specifies the actual structure of the extension element containing the foreign non-UBL content. This is delivered as a functional illustration of the definition of an extension schema by declaring the content to be a single element either in or out of the UBL Signature Extension namespace. This content module is replaced when defining an extension schema of another namespace, similarly declaring the content to be a single element either in or out of that extension's namespace.

3.2.5. Signature Extension Schemas

See Section 5, “UBL XAdES Extension for Digital Signatures” for further information regarding the UBL XAdES Extension for Digital Signatures.

CommonSignatureComponents
xsd/common/UBL-CommonSignatureComponents-2.1.xsd

The CommonSignatureComponents schema defines the scaffolding structures containing the W3C Digital Signature information XML elements related to either the entire document or particular signature business objects found within the document.

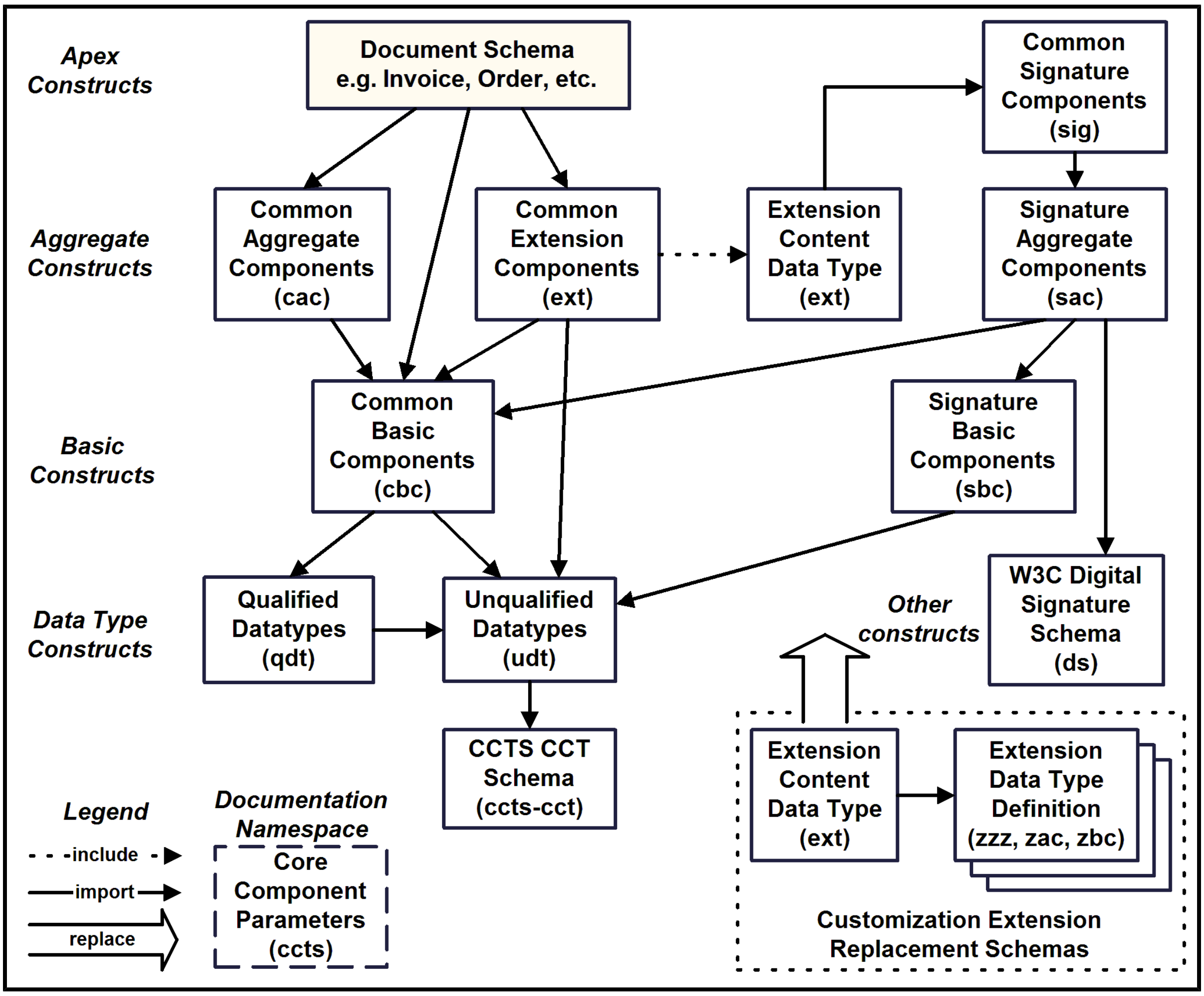
xmldsig-core-schema
xsd/common/UBL-xmldsig-core-schema-2.1.xsd

This is a copy of the W3C Digital Signature core schema file [http://www.w3.org/TR/xmldsig-core/xmldsig-core-schema.xsd], modified only to remove the unnecessary PUBLIC and SYSTEM identifiers from the DOCTYPE.

3.3. Schema Dependencies

The following diagram shows the dependencies among the schema modules making up a UBL 2.1 document schema.

Figure 64. UBL Schema Dependencies


Apex Constructs
Document Schema e.g. Invoice, Order, etc.
Common Signature Components (sig)
Aggregate Constructs
Common Aggregate Components (cac)
Common Extension Components (ext)
Extension Content Data Type (ext)
Signature Aggregate Components (sac)
Basic Constructs
Common Basic Components (cbc)
Signature Basic Components (sbc)
Data Type Constructs
Qualified Datatypes (qdt)
Unqualified Datatypes (udt)
Other constructs
W3C Digital Signature Schema (ds)
CCTS CCT Schema (ccts-cct)
Extension Content Data Type (ext)
Extension Data Type Definition (zzz, zac, zbc)
Customization Extension Replacement Schemas
Legend
include
import
replace
Documentation Namespace
Core Component Parameters (ccts)

The UBL schemas are delivered supporting the UBL standardized extension for digital signatures, defining the content of the extension to be a single element either in or out of the UBL signature extension namespace. As shown on the right in this diagram, a set of UBL schemas supporting a different user-customized extension is created by replacing the delivered ExtensionContentDataType schema fragment with one defining the content of the extension to be a singe element either in or out of the custom extension, importing the required custom schema fragments that define the custom content. For more regarding the signature extension, see Section 5, “UBL XAdES Extension for Digital Signatures”.

4. Additional Document Constraints

In addition to the UBL 2.1 document constraints formally expressed by the schemas in Section 3, "UBL 2.1 Schemas", UBL mandates several other rules governing conformant UBL 2.1 instances that cannot be expressed using W3C Schema. These additional UBL document rules, addressing instance validation, character encoding, and empty elements, are specified below.

These rules first appeared in the OASIS UBL 1.0 and UBL 1.0 NDR Standards. They are listed here because logically they belong with the great majority of UBL instance constraints specified in the schemas. To aid in coordinating references between these various publications, the rules below retain their original "IND" labels. The former IND4 was removed in the revision process leading to UBL 2.0.

4.1. Validation

The UBL library and document schemas are targeted at supporting business information exchanges. Business information exchanges require a high degree of precision to ensure that application processing and corresponding business cycle actions are reflective of the purpose, intent, and information content agreed to by both trading partners. Schemas provide the necessary mechanism for ensuring that instance documents do in fact support these requirements.

> **[IND1]** All UBL instance documents MUST validate to a corresponding schema.

4.2. Character Encoding

XML supports a wide variety of character encodings. Processors must understand which character encoding is employed in each XML document. XML 1.0 supports a default value of UTF-8 for character encoding, but best practice is to always identify the character encoding being employed.

> **[IND2]** All UBL instance documents MUST identify their character encoding within the XML declaration.

Example:

```
<?xml version="1.0" encoding="UTF-8"?>
```

UBL, as an OASIS TC, is obligated to conform to agreements OASIS has entered into. OASIS is a liaison member of the ISO IEC ITU UN/CEFACT eBusiness Memorandum of Understanding Management Group (MOUMG). Resolution 01/08 (MOU/MG01n83) requires the use of UTF-8.

> **[IND3]** In conformance with ISO IEC ITU UN/CEFACT eBusiness Memorandum of Understanding Management Group (MOUMG) Resolution 01/08 (MOU/MG01n83) as agreed to by OASIS, all UBL XML SHOULD be expressed using UTF-8.

Example:

```
<?xml version="1.0" encoding="UTF-8"?>
```

4.3. Empty Elements

Use of empty elements within XML instance documents is a source of controversy for a variety of reasons. An empty element does not simply represent data that is missing. It may express data that is not applicable for some reason, trigger the expression of an attribute, denote all possible values instead of just one, mark the end of a series of data, or appear as a result of an error in XML file generation. Conversely, missing data elements can also have meaning—data not provided by a trading partner. In information exchange environments, different trading partners may allow, require, or ban empty elements. UBL has determined that empty elements do not provide the level of assurance necessary for business information exchanges and therefore will not be used.

> **[IND5]** UBL conformant instance documents MUST NOT contain an element devoid of content or containing null values, except in the case of extension, where the UBL ExtensionContent element is used.

To ensure that no attempt is made to circumvent rule IND5, UBL also prohibits attempting to convey meaning by not conveying an element.

> **[IND6]** The absence of a construct or data in a UBL instance document MUST NOT carry meaning.

5. UBL XAdES Extension for Digital Signatures

NOTE: The mechanism for digital signatures described in this section has not been extensively tested and is included here only for the benefit of reviewers. The OASIS UBL TC invites all interested parties to exercise the method described here and report back via the UBL TC Comment page.

UBL extensions enable user-defined additions to the standard schemas. In this release (UBL 2.1 PRD1), the UBL 2.1 schemas are provided with a predefined standard extension that supports advanced digital signatures conforming to the ETSI XAdES specification [XAdES], thus satisfying EU legal requirements for electronically signed business documents.

This extension also serves as a case study for the creation of user-defined UBL extensions; see Section 5.8, “Notes for Extension Creators (Informative)”. Further information on the UBL extension mechanism can be found in [Customization].

UBL's implementation of XAdES puts all the signatures relating to a document in a single extension, which is engaged in validation by the `UBL-ExtensionContentDataType-2.1.xsd` schema module. A detailed analysis and description of the digital signature methodology is given in the UBL Security Subcommittee document titled "UBL Electronic Signature Profile Version 1.0".

5.1. XAdES Extension Namespaces

As is true for the document schemas and common library, the UBL XAdES extension is modeled with three namespaces: one for the apex element (a parallel to the document schema), one for new aggregate constructs (a parallel to the common aggregate schema) and one for new basic constructs (a parallel to the common basic schema). See Figure 64, “UBL Schema Dependencies”.

The `urn:oasis:names:specification:ubl:schema:xsd:CommonSignatureComponents-2` namespace is used for the apex element, the `urn:oasis:names:specification:ubl:schema:xsd:SignatureAggregateComponents-2` namespace is used for new aggregate elements, and the `urn:oasis:names:specification:ubl:schema:xsd:SignatureBasicComponents-2` namespace is used for new basic elements. The W3C digital signature [xmldsig] standard namespace `http://www.w3.org/2000/09/xmldsig#` is also used in this extension. These namespaces are bound to the `sig:`, `sac:`, `sbc:` and `ds:` prefixes respectively, but any prefix or even the default namespace can be used for any of these in an XML instance.

5.2. XAdES Extension Identification

This UBL Signature Extension is distinguished from other extensions and identified using the URI `urn:oasis:names:specification:ubl:profile:dsig:signature` in the `<ext:ExtensionURI>` element.

5.3. XAdES Extension Validation

The `UBL-ExtensionContentDataType-2.1.xsd` module links UBL extension validation to a particular extension configuration. The distribution version of this module supports XAdES digital signatures by declaring that the `<ext:ExtensionContent>` element contains a single `<sig:UBLDocumentSignatures>` element as the apex of all the document's electronic signatures.

The `<ext:ExtensionContent>` element alternatively allows any other non-signature namespace apex element in order to allow other extensions in the same document.

5.4. XAdES Extension Structure

The signature extension structure exists to contain one or more W3C standard digital schema constructs. The UBL scaffolding for this extension starts with a `<ext:UBLExtension>` element with two children: `<ext:ExtensionURI>` (for extension distinction and identification) and `<ext:ExtensionContent>` (for containing the extension information):

```
<ext:UBLExtension>
 <ext:ExtensionURI
>urn:oasis:names:specification:ubl:profile:dsig:signature</ext:ExtensionURI>
 <ext:ExtensionContent>
```

All signature information for the document is then contained under the `<sig:UBLDocumentSignatures>` apex element, a single mandatory child of `<ext:ExtensionContent>`:

```
<ext:ExtensionContent>
  <sig:UBLDocumentSignatures>
```

Every signature added to the extension is isolated under a separate `<sac:SignatureInformation>` aggregate element. As many of these aggregates can be in the extension as is needed, each one containing the information for a single digital extension.

Any of these aggregates can optionally be identified for referencing purposes using the common library `<cbc:ID>` element. Such an identifier may be useful in workflow scenarios where a particular signature needs to be identified external to the document, but its use is not obligatory. The identifier used can be any value, but for convenience the URI of a URN beginning with `urn:oasis:names:specification:ubl:signatures:` and ending with a number value is reserved for this purpose for UBL users. An example is `urn:oasis:names:specification:ubl:signatures:3`. As with all identifiers, each should be unique across all identifier values.

Any of these aggregates can optionally make reference to an existing `<cac:Signature>` business object in the same UBL document, but this is not obligatory. When needed, the `<sbc:ReferencedSignatureID>` basic element is used to point to the `<cbc:ID>` identifier value of the referenced `<cac:Signature>`. The identifier used can be any value, but for convenience the URI of a URN beginning with `urn:oasis:names:specification:ubl:signatures:` and ending with the local name of the parent of the signature business object and optionally followed with a colon and number, as in `urn:oasis:names:specification:ubl:signatures:IssuerEndorsement`, is reserved for this purpose for UBL users. As with all identifier references, the referenced identifier should exist and should be unique across all such identifier values.

A skeleton example of a single signature is as follows:

```
<ext:ExtensionContent>
  <sig:UBLDocumentSignatures xmlns:sig=
    "urn:oasis:names:specification:ubl:schema:xsd:CommonSignatureComponents-2"
    xmlns:sac=
 "urn:oasis:names:specification:ubl:schema:xsd:SignatureAggregateComponents-2"
    xmlns:sbc=
     "urn:oasis:names:specification:ubl:schema:xsd:SignatureBasicComponents-2">
    <sac:SignatureInformation>
      <cbc:ID>urn:oasis:names:specification:ubl:signatures:1</cbc:ID>
      <sbc:ReferencedSignatureID>MyParty</sbc:ReferencedSignatureID>
      <Signature xmlns="http://www.w3.org/2000/09/xmldsig#">
        <SignedInfo>
        ...
        </SignedInfo>
        <SignatureValue>
        ...
        </SignatureValue>
        <KeyInfo>
        ...
        </KeyInfo>
       </Signature>
    </sac:SignatureInformation>
  </sig:UBLDocumentSignatures>
</ext:ExtensionContent>
```

The document can have any number of `<sac:SignatureInformation>` elements, each containing at most one `<ds:Signature>` element. An element with an absent `<ds:Signature>` can be used as a placeholder for workflow software to identify the placement for a signature to be added.

5.5. XAdES Transformation

A requirement when using XAdES is to express in XPath that address that qualifies all nodes in an XML document included in the calculation of the digital signature hash. For a signature added to a document to remain valid, none of the information can change, nor can any information be added or removed from that portion of the document included in the hash calculation.

There are two such transformation expressions that should be used in the UBL signature extension; choose the appropriate one to meet the objectives of the signature being added to the document. Adding non-signature information to the UBL document will invalidate all signatures already in the extension. The choice to make is in regard to the support of adding additional signatures after adding the signature with the transformation expression.

The following transformation element in a digital signature flexibly prevents the signature being invalidated by the subsequent addition of other signatures within the extension:

```
   <Transform
     Algorithm="http://www.w3.org/TR/1999/REC-xpath-19991116">
    <XPath xmlns:sig=
"urn:oasis:names:specification:ubl:schema:xsd:CommonSignatureComponents-2">
      count(ancestor-or-self::sig:UBLDocumentSignatures |
             here()/ancestor::sig:UBLDocumentSignatures[1]) >
      count(ancestor-or-self::sig:UBLDocumentSignatures)
    </XPath>
   </Transform>
```

The following transformation element in a digital signature is inflexible and thus would be considered a "final" signature to be added to the document. Such a signature will be invalidated by the subsequent addition of other signatures to the document:

```
   <Transform
     Algorithm="http://www.w3.org/TR/1999/REC-xpath-19991116">
    <XPath xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
      count(ancestor-or-self::ds:Signature |
             here()/ancestor::ds:Signature[1]) >
      count(ancestor-or-self::ds:Signature)
    </XPath>
   </Transform>
```

5.6. Example XAdES Extension Instance

The `xml/UBL-Invoice-2.0-Signed.xml` [`xml/UBL-Invoice-2.0-Signed.xml`] sample document illustrates the embedding of three extensions in a single document, one of which is the signature extension. The example signature includes but one of the many formats of XAdES and its available options.

5.7. Extension Validation Methodology (Informative)

The single extension built into the UBL 2.1 distribution validates transparently, and the UBL extension mechanism allows the addition of other extensions in the same instance. But users wishing to include multiple extensions in a single instance should be aware of a schema property that requires a multipass approach to validating such instances.

Due to limitations of W3C Schema validation semantics (this is not the case in RELAX NG, for example), the global naming strategy underlying UBL 2 means that when there are many extensions in an instance, a set of UBL schemas can only validate one of the extension namespaces at a time. In order to add validation for another extension to the XAdES extension included in this package, therefore, one must create a second set of parallel schemas in which the signature extension included in this release is replaced with the schema of a different extension. A complete validation then consists of checking the document instance against each set of schemas in turn, with as many validation runs as there are extensions. Programming this is not difficult as long as one is aware of the need for it. Figure 64, "UBL Schema Dependencies" shows the construction of a set of parallel schemas in which user-defined extension modules with namespaces `ext:`, `zzz:`, `zac:`, and `zbc:` replace the XAdES extension modules with namespaces `ext:`, `sig:`, `sac:`, `sbc:` and `ds:`.

Alternatively, one can modify the XAdES extension itself to include additional data items, but in doing so, one can no longer claim to be using the XAdES signatures standardized by this specification.

5.8. Notes for Extension Creators (Informative)

The UBL XAdES extension has been modeled as an example to follow when designing and writing other custom extensions. The following points should be noted:

- Extension designers should follow the example in providing separate namespaces for apex element, aggregate constructs, and basic constructs if they wish the new items to be considered for inclusion in future UBL releases. This stuctures the new items for inclusion in the UBL common library.

- Whenever possible, existing UBL common library aggregate and basic constructs should be used in extensions rather than inventing new items with the same semantics. However, a common library aggregate construct should only be used when the entire aggregate and all of its descendants are applicable in the extension context without any changes. If any items must be removed, then a new extension aggregate with a new local name should be used. If all the constructs are applicable but some items need to be added, then a new extension aggregate with the same local name as the common library should be used, and the common library structure should be copied with the new constructs inserted.

6. Conformance

Conformance (as applied to UBL documents and schemas) and the distinction between UBL conformance and UBL compatibility is described in detail in the *UBL 2 Guidelines for Customization* [Customization].

Appendix A. Release Notes (Informative)

A.1. Availability

Online and downloadable versions of this release are available from the locations specified at the top of this document.

A.2. Status of This Release

Release of this package to the public begins its first public review. The UBL Technical Committee actively solicits input from the user community regarding this release. See Status for procedures to be used in submitting comments to the Committee. Note that the UBL TC *cannot* accept input from anyone outside the UBL TC (including OASIS members) unless it is submitted via the comment list.

THIS INITIAL RELEASE IS FOR TESTING PURPOSES ONLY AND SHOULD NOT BE USED FOR PRODUCTION SYSTEMS.

A.3. Package Structure

The first public review draft of the UBL 2.1 specification is published as a zip archive named prd1-UBL-2.1.zip. Unzipping this archive creates a directory named prd1-UBL-2.1 containing a master DocBook XML file (UBL-2.1.xml), a generated hypertext version of this file (UBL-2.1.html), a generated PDF version of this file (UBL-2.1.pdf), and a number of subdirectories. The files in these subdirectories, linked to from UBL-2.1.xml, UBL-2.1.html, and UBL-2.1.pdf, contain the various normative and informational pieces of the 2.1 release. A description of each subdirectory is given below. Note that while the UBL-2.1.xml file is the “original” of this specification, it may not be viewable in all currently available web browsers.

`art`
Diagrams and illustrations used in this specification

`asn`
ASN.1 UBL 2.1 schema; see Section F.1, “ASN.1 UBL 2.1 Specification”

`cl`
Code list specification files; see Appendix D, *UBL 2.1 Code Lists and Two-phase Validation (Informative)*

`css`
CSS stylesheets for viewing UBL-2.1.html

`cva`
Artefacts expressing data type qualifications; see [CVA] in section 1.2, Normative References

`db`
DocBook stylesheets for viewing UBL-2.1.xml

`doc`
Documents included with this release

`etc`
Miscellaneous supporting information

`mod`
Spreadsheet data models; see Appendix C, *The UBL 2.1 Data Model (Informative)*

`rnc`
: Alternative versions of the UBL 2.1 schemas in RELAX NG (compact syntax); see Section F.2, “UBL 2.1 RELAX NG Schemas”

`val`
: Test harness for demonstrating UBL 2.1 two-phase validation; see Appendix D, *UBL 2.1 Code Lists and Two-phase Validation (Informative)*

`xml`
: Sample UBL 2.1 instances

`xsd`
: XSD schemas; see Section 3, “UBL 2.1 Schemas”

`xsdrt`
: “Runtime” XSD schemas; see Section 3, “UBL 2.1 Schemas”

A.4. Support

UBL is a volunteer project of the international business community. Inquiries regarding UBL may be posted to the public ubl-dev list, archives for which are located at

> http://lists.oasis-open.org/archives/ubl-dev/

Subscriptions to ubl-dev can be made through the OASIS list manager at

> http://www.oasis-open.org/mlmanage/index.php

OASIS provides an official community gathering place and information resource for UBL at

> http://ubl.xml.org/

A.5. Expected Additions in PRD2

Several additional document types and information items are expected to be added to UBL 2.1 following this initial public review (PRD1), that is, in the second public review cycle (PRD2). They include but are not limited to the following:

New document types expected for PRD2

> GoodsItemItinerary
> TransportServiceDescription
> CatalogueTemplate
> VirtualCompanyDossier

New information items expected for PRD2

> AllowanceCharge within ItemLocationQuantity
> VersionID in DocumentReference

Note that the names of these items may change before their inclusion in PRD2.

Other changes expected in PRD2

> Addition of document-wide signature elements to the five document types that currently lack them: (CallForTenders, CatalogueRequest, ContractAwardNotice, ContractNotice, PriorInformationNotice)

A.6. Taxation Rules

UBL does not provide documents for tax reporting purposes. Instead, it provides structures to support the information on which taxes are based. These aim to be generic and not based on any specific tax regime.

A.7. UBL Customization

UBL provides a vocabulary that, for many user communities, can be used "as is." However, it is recognized that some user communities must address use cases whose requirements are not met by the UBL off-the-shelf solution. A separate OASIS Committee Specification known as the *UBL 2 Guidelines for Customization* [Customization] has been published to aid such users in developing custom solutions based on UBL.

The goal of UBL customization is to maintain a common understanding of the meaning of information being exchanged between specific implementations. The determining factors governing when to customize may be business-driven, technically driven, or both. The decision should be based on real world needs balanced against perceived economic benefits.

A.8. Upgrading from UBL 2.0 to UBL 2.1

For current UBL implementers, the most important thing to know about UBL 2.1 is that it is completely backward-compatible with UBL 2.0. In other words, any document that validates against a UBL 2.0 schema will validate against the UBL 2.1 version of that schema. The remaining differences relate mainly to functionality that has been added to the 2.0 framework in the areas of eTendering, sales reporting, utility statements, transport handling, and collaborative planning, forecasting, and replenishment (CPFR®).

Nonetheless, it would be unwise to simply overlay this UBL 2.1 release onto an existing 2.0 installation, and the possible differences among existing 1.0 and 2.0 installations are too large to allow a specific set of instructions to be provided for making the transition.

The brief history of UBL document types in the next section puts the new capabilities into context and may help owners of existing UBL 1.0 and 2.0 installations decide whether to upgrade to 2.1. New 2.1 users, on the other hand, can simply install 2.1 and rest assured that their software will interoperate with UBL documents generated by existing conformant UBL 2.0 installations. For more on the concept of conformance, see [Customization].

A.9. DEN Corrections in UBL 2.1 PRD1

Dictionary Entry Names (DENs) uniquely identify every BIE in the UBL library using the methodology described in [CCTS]. Several errors in dictionary entry naming were discovered and corrected in the course of preparing UBL 2.1 PRD1. These corrections have no effect on processing, validation, or instance generation; they are listed in the table below merely for completeness in documentation. A number of other DEN corrections are expected for PRD2 and will be documented there.

Table A.1. Dictionary Entry Name (DEN) Corrections in UBL 2.1

UBL 2.0 Dictionary Entry Name	UBL 2.1 Dictionary Entry Name
Application Response. Version Identifier. Identifier	Application Response. Version. Identifier
Certificate Of Origin. Version Identifier. Identifier	Certificate Of Origin. Version. Identifier
Item Comparison. Price. Amount	Item Comparison. Price Amount. Amount
Monetary Total. Allowance Total Amount. Amount	Monetary Total. Allowance_ Total Amount. Amount
Monetary Total. Charge Total Amount. Amount	Monetary Total. Charge_ Total Amount. Amount
Order Change. Customer Reference. Text	Order Change. Customer_ Reference. Text
Order Change. Sales Order Identifier. Identifier	Order Change. Sales_ Order Identifier. Identifier
Order Change. Sequence_ Number. Identifier	Order Change. Sequence Number. Identifier
Order Reference. Sales Order Identifier. Identifier	Order Reference. Sales_ Order Identifier. Identifier
Order Response. Customer Reference. Text	Order Response. Customer_ Reference. Text
Order Response. Sales Order Identifier. Identifier	Order Response. Sales_ Order Identifier. Identifier
Order. Customer Reference. Text	Order. Customer_ Reference. Text
Order. Sales Order Identifier. Identifier	Order. Sales_ Order Identifier. Identifier
Packing List. Version Identifier. Identifier	Packing List. Version. Identifier
Receipt Line. Oversupply Quantity. Quantity	Receipt Line. Oversupply_ Quantity. Quantity
Signature. Validator Identifier. Identifier	Signature. Validator. Identifier
Status. Sequence. Identifier	Status. Sequence Identifier. Identifier

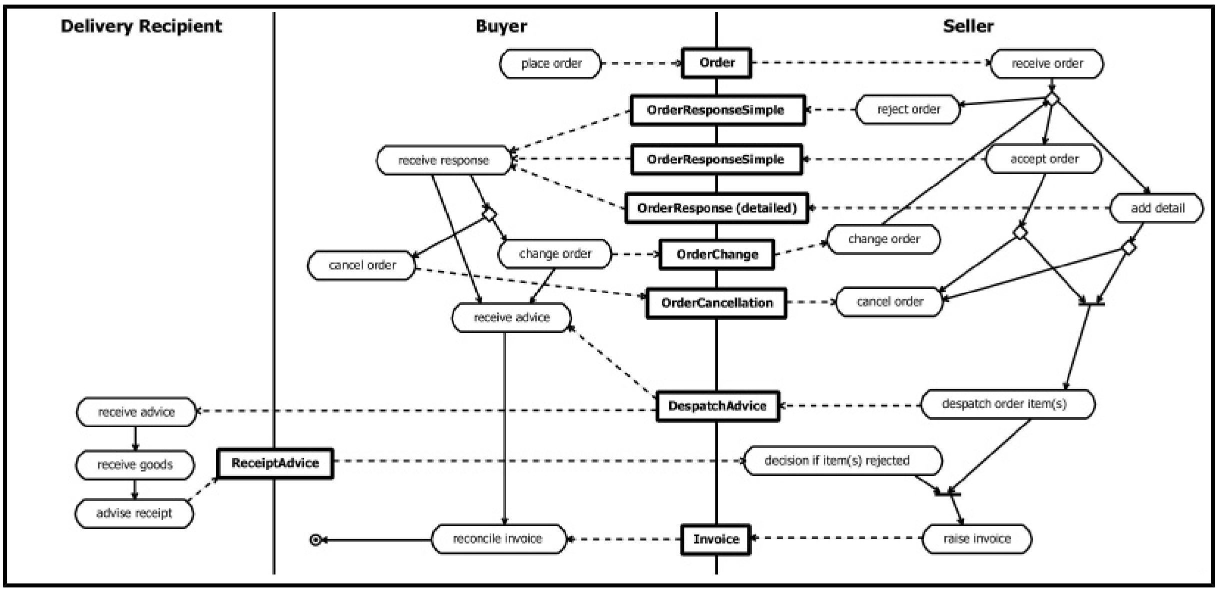
Appendix B. Revision History

Since its first release as an OASIS Standard in 2004, UBL has experienced one major and one minor version upgrade.

B.1. UBL 1.0

Though apparently limited in scope, the eight document types provided in UBL 1.0 (2004) are applicable to a very large number of real-world use cases and have been widely deployed. These original 1.0 document types, later updated in UBL 2.0 and continued here in 2.1, are Order, OrderResponse, OrderResponseSimple, OrderChange, OrderCancellation, DespatchAdvice, ReceiptAdvice, and Invoice. The figure below shows the original assumed process context for this most basic set of UBL document types. The scope of the process corresponds roughly to that of the UBL 2 Order, Fulfillment, and Traditional Billing processes described in the text (see Section 2.6, “Ordering”, Section 2.7, “Fulfilment”, and Section 2.8.2, “Traditional Billing”).

Figure B.1. UBL 1.0 Order-to-Invoice Business Process



Because later versions of UBL (versions 2.0 and following) do not maintain backward compatibility with UBL 1.0 document instances (that is, UBL 1.0 document instances will not validate against schemas from UBL 2.0 and later), use of UBL 1.0 in new installations is deprecated. All the business functionality of UBL 1.0 (including suitably revised versions of the original eight document types) is maintained in later versions.

B.2. Major Revision: UBL 2.0

Adoption of UBL 1.0 following ratification as an OASIS standard in November 2004 resulted in major inputs of new business content beyond the eight basic order-to-invoice business documents specified in the original release. In particular, contributions from representatives of government procurement, taxation, and transportation agencies in Europe, Asia, and North America resulted in greatly expanded pre-order and post-invoice capabilities together with the addition of several transport-related document types, bringing the total number of document types in UBL 2.0 to 31.

The new release also featured changes in UBL's use of XML schema methodology—most importantly, the adoption of global scoping for all element types — breaking backward compatibility with UBL 1.0 instances and therefore necessitating designation as a major revision, signified by incrementing the version number from 1.0 to 2.0 rather than 1.1. The original eight UBL 1.0 document types were revised to reflect these changes.

UBL 2.0 achieved OASIS Standardization in December 2006, and the package was updated and corrected in May 2008.

The 23 document types added in UBL 2.0 can be summarized as follows:

Added UBL 2.0 document types for sourcing: Catalogue, CatalogueDeletion, CatalogueItemSpecificationUpdate, CataloguePricingUpdate, CatalogueRequest, Quotation, RequestForQuotation

Added UBL 2.0 document types for fulfilment: BillOfLading, CertificateOfOrigin, ForwardingInstructions, PackingList, TransportationStatus, Waybill

Added UBL 2.0 document types for billing: CreditNote, DebitNote, FreightInvoice, Reminder, SelfBilledCreditNote, SelfBilledInvoice

Added UBL 2.0 document types for payment: RemittanceAdvice, Statement

Added UBL 2.0 supplementary document types: ApplicationResponse, AttachedDocument

B.3. Minor Revision: UBL 2.1

B.3.1. New Document Types in UBL 2.1

Because it preserves backward compatibility with UBL 2.0, UBL 2.1 is technically a minor release, not a major one. However, it does add 29 new document types in this first public review draft, bringing the total number of UBL business documents to 60.

Added UBL 2.1 document types for eTendering: AwardedNotification, CallForTenders, ContractAwardNotice, ContractNotice, GuaranteeCertificate, Tender, TenderReceipt, TendererQualification, TendererQualificationResponse, UnawardedNotification

Added UBL 2.1 document types for Collaborative planning, forecasting, and replenishment: ExceptionCriteria, ExceptionNotification, Forecast, ForecastRevision, ItemInformationRequest, PriorInformationNotice, TradeItemLocationProfile

Added UBL 2.1 document types for Vendor Managed Inventory: InstructionForReturns, InventoryReport, PerformanceHistory, ProductActivity, RetailEvent, StockAvailabilityReport

Added UBL 2.1 document types for Intermodal Freight Management: TransportExecutionPlan, TransportExecutionStatus, TransportOperationStatus

Added UBL 2.1 document type for Utility billing: UtilityStatement

Added UBL 2.1 supplementary document types: DocumentStatus, DocumentStatusRequest

B.3.2. Financial Information Enhancements in UBL 2.1

UBL 2.1 has been enhanced to support the financial information required for downstream processing of Invoices within financial services. Through standardization, business vocabularies such as UBL for eBusiness and ISO 20022 for eFinance enable Straight Through Processing (STP) and paperless trading along the entire Financial Supply Chain.

Based on analysis conducted during the current development cycle by the UBL Financial Information Requirements Task Group (FIRTG), the following enhancements have been included in UBL 2.1:

> **Financial account:** Aligned with today's needs and designed for truly global usage (AliasName, AccountTypeCode, ...). A financial account can be now associated to the Person information aggregate, not only to a Party.
>
> **Payment mandate** information can optionally be sent as part of the Order; this can be considered a simplification for small businesses.
>
> **Trade financing:** UBL 2.1 is designed to support basic trade financing practices (invoice financing, factoring, pre-shipment/order financing, Letter of Credit, ...)
>
> **Payments reconciliation:** UBL Invoice and Remittance Advice can be used together with financial messages to ensure end-to-end transport of reconciliation identifiers (invoicing party references). In particular, UBL provides a solution for advanced external remittance, where the UBL Remittance Advice is used to transmit the details of complex remittance information associated with the payment initiation process (see ISO20022 guides for details). Person is now enriched with a person identification, which is often required by the banking sector for legal reasons.
>
> **Currency Amounts:** UBL 2.1 features improved handling of alternative currency amounts.

UBL 2.1 also includes enhancements to legal information.

> **Party Legal Entity:** The Party's legal information has been considerably enriched with information required by advanced procurement and global usage.
>
> **Service Provider Party:** The electronic trade is increasingly supported and executed through Service Providers into several forms like the outsourcing and ASP modes.
>
> **UBL Party** is now improved to keep track of services handled by one or more service providers.
>
> **Power of Attorney** can now be associated to a Party.

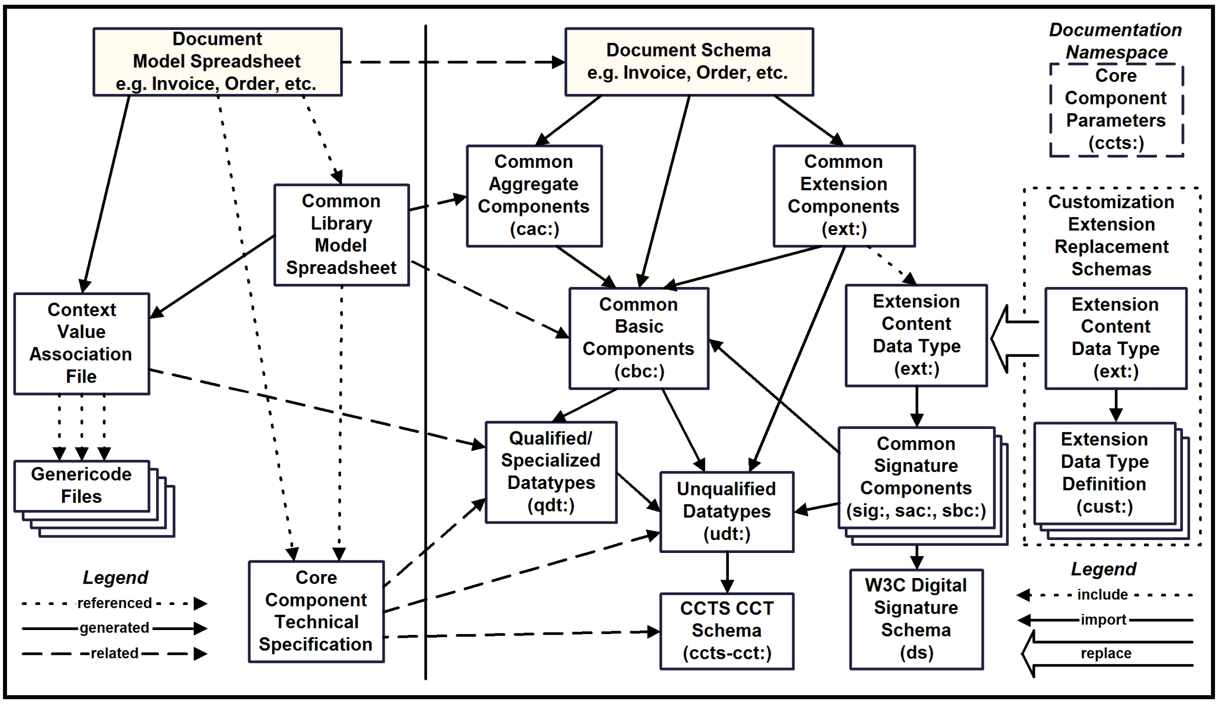
Appendix C. The UBL 2.1 Data Model (Informative)

Following the principles of the ebXML Core Components Technical Specification (CCTS), the UBL data model is based on a conceptual library of reusable information items known as Business Information Entities (BIEs). BIEs include BBIEs (“basic” individual pieces of information), ABIEs (aggregations of other BIEs), and ASBIEs (associations to other ABIEs). See [CCTS] for a further explanation of these terms. Each business document defined by UBL is an ABIE created by assembling items appropriate to that document type from the BIE library.

Historically, both the UBL common library and the assembly models for the individual UBL documents have been expressed as spreadsheets using a format specifically developed for UBL business information modeling. In former UBL releases, the XSD schemas that serve as the normative expression of UBL syntax were generated directly from the spreadsheets prepared by business experts according to the UBL NDR (see Appendix G, *UBL 2.1 Naming and Design Rules (Informative)*). In UBL 2.0, the entire data model was also entered (via the spreadsheets) into the internal format of a commercial data management system from GEFEG that was used to insure data integrity. In UBL 2.1, that process has been taken one step further; the data model is instantiated and maintained in iSurf eDoCreator, and both schemas and spreadsheets are generated from that internal model, with the schemas again produced according the UBL Naming and Design Rules and the spreadsheets serving to provide base-level human-readable documentation and to capture the supplementary metadata required by [CCTS]. By preserving a vendor-neutral representation from which schemas can be generated directly if necessary, the spreadsheets guarantee that the UBL model is not bound to a single production system.

The following diagram shows the conceptual relationships between spreadsheets and schemas in UBL 2.1, with spreadsheets on the left and schemas on the right. Compare Figure 64, “UBL Schema Dependencies”.

Figure C.1. UBL Spreadsheet Realization



Document Model Spreadsheet e.g. Invoice, Order, etc.
Document Schema e.g. Invoice, Order, etc.
Documentation Namespace
Core Component Parameters (ccts:)
Common Aggregate Components (cac:)
Common Extension Components (ext:)
Common Library Model Spreadsheet
Customization Extension Replacement Schemas
Context Value Association File
Common Basic Components (cbc:)
Extension Content Data Type (ext:)
Extension Content Data Type (ext:)
Genericode Files
Qualified/ Specialized Datatypes (qdt:)
Unqualified Datatypes (udt:)
Common Signature Components (sig:, sac:, sbc:)
Extension Data Type Definition (cust:)
Legend
referenced
generated
related
Core Component Technical Specification
CCTS CCT Schema (ccts-cct:)
W3C Digital Signature Schema (ds)
Legend
include
import
replace

C.1. The UBL Common Library

As noted above, UBL is based on a reusable library of Business Information Entities. In the current release, the Common Library contains more than two thousand of these individually defined data items. The entire UBL 2.1 library of Business Information Entities is contained in a single spreadsheet.

mod/common/UBL-CommonLibrary-2.1.ods
mod/common/UBL-CommonLibrary-2.1.xls

C.2. UBL Document Models

A UBL 2.1 document model defines a single “root” Aggregate BIE. This may contain several Basic BIEs and Association BIEs. Assembling the components of all the Association BIEs referenced from this root creates the hierarchical structure necessary to represent the document type. There are common patterns in the structure of many UBL 2.1 document models, but UBL does not enforce a "common header" for all business documents.

The document models are provided in this package as Excel and ODF spreadsheets. As noted above, these spreadsheets function as a basic form of documentation. A more accessible form of documentation is provided by HTML reports contributed by Crane Softwrights and included here by permission. Each document report summarizes business object definitions and selected columns of the corresponding spreadsheet in a hyperlinked form that omits unused elements to facilitate rapid review of each document model.

There is also a single master report incorporating every document type and the entire common library:

mod/summary/reports/UBL-AllDocuments-2.1.html

For notes on the use of these reports, see

mod/summary/readme-Reports.html

For other views of this information, see Section C.3, “Business Information Entity Documentation” and Section C.4, “UML-Based View of the UBL Data Model”.

ApplicationResponse

mod/maindoc/UBL-ApplicationResponse-2.1.ods
mod/maindoc/UBL-ApplicationResponse-2.1.xls
mod/summary/reports/UBL-ApplicationResponse-2.1.html

AttachedDocument

mod/maindoc/UBL-AttachedDocument-2.1.ods
mod/maindoc/UBL-AttachedDocument-2.1.xls
mod/summary/reports/UBL-AttachedDocument-2.1.html

AwardedNotification

mod/maindoc/UBL-AwardedNotification-2.1.ods
mod/maindoc/UBL-AwardedNotification-2.1.xls
mod/summary/reports/UBL-AwardedNotification-2.1.html

BillOfLading

mod/maindoc/UBL-BillOfLading-2.1.ods
mod/maindoc/UBL-BillOfLading-2.1.xls
mod/summary/reports/UBL-BillOfLading-2.1.html

CallForTenders

mod/maindoc/UBL-CallForTenders-2.1.ods
mod/maindoc/UBL-CallForTenders-2.1.xls
mod/summary/reports/UBL-CallForTenders-2.1.html

Catalogue

mod/maindoc/UBL-Catalogue-2.1.ods
mod/maindoc/UBL-Catalogue-2.1.xls
mod/summary/reports/UBL-Catalogue-2.1.html

CatalogueDeletion

mod/maindoc/UBL-CatalogueDeletion-2.1.ods
mod/maindoc/UBL-CatalogueDeletion-2.1.xls
mod/summary/reports/UBL-CatalogueDeletion-2.1.html

CatalogueItemSpecificationUpdate

mod/maindoc/UBL-CatalogueItemSpecificationUpdate-2.1.ods
mod/maindoc/UBL-CatalogueItemSpecificationUpdate-2.1.xls
mod/summary/reports/UBL-CatalogueItemSpecificationUpdate-2.1.html

CataloguePricingUpdate

mod/maindoc/UBL-CataloguePricingUpdate-2.1.ods
mod/maindoc/UBL-CataloguePricingUpdate-2.1.xls
mod/summary/reports/UBL-CataloguePricingUpdate-2.1.html

CatalogueRequest

mod/maindoc/UBL-CatalogueRequest-2.1.ods
mod/maindoc/UBL-CatalogueRequest-2.1.xls
mod/summary/reports/UBL-CatalogueRequest-2.1.html

CertificateOfOrigin

mod/maindoc/UBL-CertificateOfOrigin-2.1.ods
mod/maindoc/UBL-CertificateOfOrigin-2.1.xls
mod/summary/reports/UBL-CertificateOfOrigin-2.1.html

ContractAwardNotice

mod/maindoc/UBL-ContractAwardNotice-2.1.ods
mod/maindoc/UBL-ContractAwardNotice-2.1.xls
mod/summary/reports/UBL-ContractAwardNotice-2.1.html

ContractNotice

mod/maindoc/UBL-ContractNotice-2.1.ods
mod/maindoc/UBL-ContractNotice-2.1.xls
mod/summary/reports/UBL-ContractNotice-2.1.html

CreditNote

mod/maindoc/UBL-CreditNote-2.1.ods
mod/maindoc/UBL-CreditNote-2.1.xls
mod/summary/reports/UBL-CreditNote-2.1.html

DebitNote

mod/maindoc/UBL-DebitNote-2.1.ods
mod/maindoc/UBL-DebitNote-2.1.xls
mod/summary/reports/UBL-DebitNote-2.1.html

DespatchAdvice

mod/maindoc/UBL-DespatchAdvice-2.1.ods
mod/maindoc/UBL-DespatchAdvice-2.1.xls
mod/summary/reports/UBL-DespatchAdvice-2.1.html

DocumentStatus

mod/maindoc/UBL-DocumentStatus-2.1.ods
mod/maindoc/UBL-DocumentStatus-2.1.xls
mod/summary/reports/UBL-DocumentStatus-2.1.html

DocumentStatusRequest

mod/maindoc/UBL-DocumentStatusRequest-2.1.ods
mod/maindoc/UBL-DocumentStatusRequest-2.1.xls
mod/summary/reports/UBL-DocumentStatusRequest-2.1.html

ExceptionCriteria

mod/maindoc/UBL-ExceptionCriteria-2.1.ods
mod/maindoc/UBL-ExceptionCriteria-2.1.xls
mod/summary/reports/UBL-ExceptionCriteria-2.1.html

ExceptionNotification

mod/maindoc/UBL-ExceptionNotification-2.1.ods
mod/maindoc/UBL-ExceptionNotification-2.1.xls
mod/summary/reports/UBL-ExceptionNotification-2.1.html

Forecast

mod/maindoc/UBL-Forecast-2.1.ods
mod/maindoc/UBL-Forecast-2.1.xls
mod/summary/reports/UBL-Forecast-2.1.html

ForecastRevision

mod/maindoc/UBL-ForecastRevision-2.1.ods
mod/maindoc/UBL-ForecastRevision-2.1.xls
mod/summary/reports/UBL-ForecastRevision-2.1.html

ForwardingInstructions

mod/maindoc/UBL-ForwardingInstructions-2.1.ods
mod/maindoc/UBL-ForwardingInstructions-2.1.xls
mod/summary/reports/UBL-ForwardingInstructions-2.1.html

FreightInvoice

mod/maindoc/UBL-FreightInvoice-2.1.ods
mod/maindoc/UBL-FreightInvoice-2.1.xls
mod/summary/reports/UBL-FreightInvoice-2.1.html

GuaranteeCertificate

mod/maindoc/UBL-GuaranteeCertificate-2.1.ods
mod/maindoc/UBL-GuaranteeCertificate-2.1.xls
mod/summary/reports/UBL-GuaranteeCertificate-2.1.html

InstructionForReturns

mod/maindoc/UBL-InstructionForReturns-2.1.ods
mod/maindoc/UBL-InstructionForReturns-2.1.xls
mod/summary/reports/UBL-InstructionForReturns-2.1.html

InventoryReport

mod/maindoc/UBL-InventoryReport-2.1.ods
mod/maindoc/UBL-InventoryReport-2.1.xls
mod/summary/reports/UBL-InventoryReport-2.1.html

Invoice

mod/maindoc/UBL-Invoice-2.1.ods
mod/maindoc/UBL-Invoice-2.1.xls
mod/summary/reports/UBL-Invoice-2.1.html

ItemInformationRequest

mod/maindoc/UBL-ItemInformationRequest-2.1.ods
mod/maindoc/UBL-ItemInformationRequest-2.1.xls
mod/summary/reports/UBL-ItemInformationRequest-2.1.html

Order

mod/maindoc/UBL-Order-2.1.ods
mod/maindoc/UBL-Order-2.1.xls
mod/summary/reports/UBL-Order-2.1.html

OrderCancellation

mod/maindoc/UBL-OrderCancellation-2.1.ods
mod/maindoc/UBL-OrderCancellation-2.1.xls
mod/summary/reports/UBL-OrderCancellation-2.1.html

OrderChange

mod/maindoc/UBL-OrderChange-2.1.ods
mod/maindoc/UBL-OrderChange-2.1.xls
mod/summary/reports/UBL-OrderChange-2.1.html

OrderResponse

mod/maindoc/UBL-OrderResponse-2.1.ods
mod/maindoc/UBL-OrderResponse-2.1.xls
mod/summary/reports/UBL-OrderResponse-2.1.html

OrderResponseSimple

mod/maindoc/UBL-OrderResponseSimple-2.1.ods
mod/maindoc/UBL-OrderResponseSimple-2.1.xls
mod/summary/reports/UBL-OrderResponseSimple-2.1.html

PackingList

mod/maindoc/UBL-PackingList-2.1.ods
mod/maindoc/UBL-PackingList-2.1.xls
mod/summary/reports/UBL-PackingList-2.1.html

PerformanceHistory

mod/maindoc/UBL-PerformanceHistory-2.1.ods
mod/maindoc/UBL-PerformanceHistory-2.1.xls
mod/summary/reports/UBL-PerformanceHistory-2.1.html

PriorInformationNotice

mod/maindoc/UBL-PriorInformationNotice-2.1.ods
mod/maindoc/UBL-PriorInformationNotice-2.1.xls
mod/summary/reports/UBL-PriorInformationNotice-2.1.html

ProductActivity

mod/maindoc/UBL-ProductActivity-2.1.ods
mod/maindoc/UBL-ProductActivity-2.1.xls
mod/summary/reports/UBL-ProductActivity-2.1.html

Quotation

mod/maindoc/UBL-Quotation-2.1.ods
mod/maindoc/UBL-Quotation-2.1.xls
mod/summary/reports/UBL-Quotation-2.1.html

ReceiptAdvice

mod/maindoc/UBL-ReceiptAdvice-2.1.ods
mod/maindoc/UBL-ReceiptAdvice-2.1.xls
mod/summary/reports/UBL-ReceiptAdvice-2.1.html

Reminder

mod/maindoc/UBL-Reminder-2.1.ods
mod/maindoc/UBL-Reminder-2.1.xls
mod/summary/reports/UBL-Reminder-2.1.html

RemittanceAdvice

mod/maindoc/UBL-RemittanceAdvice-2.1.ods
mod/maindoc/UBL-RemittanceAdvice-2.1.xls
mod/summary/reports/UBL-RemittanceAdvice-2.1.html

RequestForQuotation

mod/maindoc/UBL-RequestForQuotation-2.1.ods
mod/maindoc/UBL-RequestForQuotation-2.1.xls
mod/summary/reports/UBL-RequestForQuotation-2.1.html

RetailEvent

mod/maindoc/UBL-RetailEvent-2.1.ods
mod/maindoc/UBL-RetailEvent-2.1.xls
mod/summary/reports/UBL-RetailEvent-2.1.html

SelfBilledCreditNote

mod/maindoc/UBL-SelfBilledCreditNote-2.1.ods
mod/maindoc/UBL-SelfBilledCreditNote-2.1.xls
mod/summary/reports/UBL-SelfBilledCreditNote-2.1.html

SelfBilledInvoice

mod/maindoc/UBL-SelfBilledInvoice-2.1.ods
mod/maindoc/UBL-SelfBilledInvoice-2.1.xls
mod/summary/reports/UBL-SelfBilledInvoice-2.1.html

Statement

mod/maindoc/UBL-Statement-2.1.ods
mod/maindoc/UBL-Statement-2.1.xls
mod/summary/reports/UBL-Statement-2.1.html

StockAvailabilityReport

mod/maindoc/UBL-StockAvailabilityReport-2.1.ods
mod/maindoc/UBL-StockAvailabilityReport-2.1.xls
mod/summary/reports/UBL-StockAvailabilityReport-2.1.html

Tender

mod/maindoc/UBL-Tender-2.1.ods
mod/maindoc/UBL-Tender-2.1.xls
mod/summary/reports/UBL-Tender-2.1.html

TenderReceipt

mod/maindoc/UBL-TenderReceipt-2.1.ods
mod/maindoc/UBL-TenderReceipt-2.1.xls
mod/summary/reports/UBL-TenderReceipt-2.1.html

TendererQualification

mod/maindoc/UBL-TendererQualification-2.1.ods
mod/maindoc/UBL-TendererQualification-2.1.xls
mod/summary/reports/UBL-TendererQualification-2.1.html

TendererQualificationResponse

mod/maindoc/UBL-TendererQualificationResponse-2.1.ods
mod/maindoc/UBL-TendererQualificationResponse-2.1.xls
mod/summary/reports/UBL-TendererQualificationResponse-2.1.html

TradeItemLocationProfile

mod/maindoc/UBL-TradeItemLocationProfile-2.1.ods
mod/maindoc/UBL-TradeItemLocationProfile-2.1.xls
mod/summary/reports/UBL-TradeItemLocationProfile-2.1.html

TransportExecutionPlan

mod/maindoc/UBL-TransportExecutionPlan-2.1.ods
mod/maindoc/UBL-TransportExecutionPlan-2.1.xls
mod/summary/reports/UBL-TransportExecutionPlan-2.1.html

TransportExecutionStatus

mod/maindoc/UBL-TransportExecutionStatus-2.1.ods
mod/maindoc/UBL-TransportExecutionStatus-2.1.xls
mod/summary/reports/UBL-TransportExecutionStatus-2.1.html

TransportOperationStatus

mod/maindoc/UBL-TransportOperationStatus-2.1.ods
mod/maindoc/UBL-TransportOperationStatus-2.1.xls
mod/summary/reports/UBL-TransportOperationStatus-2.1.html

TransportationStatus

mod/maindoc/UBL-TransportationStatus-2.1.ods
mod/maindoc/UBL-TransportationStatus-2.1.xls
mod/summary/reports/UBL-TransportationStatus-2.1.html

UnawardedNotification

mod/maindoc/UBL-UnawardedNotification-2.1.ods
mod/maindoc/UBL-UnawardedNotification-2.1.xls
mod/summary/reports/UBL-UnawardedNotification-2.1.html

UtilityStatement

mod/maindoc/UBL-UtilityStatement-2.1.ods
mod/maindoc/UBL-UtilityStatement-2.1.xls
mod/summary/reports/UBL-UtilityStatement-2.1.html

Waybill

mod/maindoc/UBL-Waybill-2.1.ods
mod/maindoc/UBL-Waybill-2.1.xls
mod/summary/reports/UBL-Waybill-2.1.html

C.3. Business Information Entity Documentation

The `mod` directory also contains a complete list of all the UBL 2.1 business information entities (BBIEs, ABIEs, and ASBIEs) in genericode format and an HTML file displaying information about ABIEs and ASBIEs in table form.

mod/UBL-Entities-2.1.gc
mod/UBL-ABIE-Reuse-Table-2.1.html

C.4. UML-Based View of the UBL Data Model

For users of Enterprise Architect, a view of the UBL 2.1 data model as a set of UML classes is provided in this package as an EA project.

uml/UBL-2.1-CommonLibraryUMLClasses.eap.zip

Appendix D. UBL 2.1 Code Lists and Two-phase Validation (Informative)

D.1. Introduction

Code lists—the sets of codes such as “FR” and “USD” that are used to specify countries, currencies, and so on—play an important role in UBL, just as they do in all electronic business messaging schemes. By default, UBL uses several lists of standard codes published by agencies such as ISO and UN/CEFACT, as well as various codes that are specific to UBL.

In UBL 1.0 (2004), standard and default code list values were enumerated directly in the UBL schemas. This allowed all UBL 1.0 instances to be validated in a single pass using generic XML XSD (W3C Schema) processors. However, the specification of the default values directly in the schemas also made it difficult to modify the code lists to suit individual trading partner relationships and impossible to extend the list of allowable code list values while still using the standard UBL schemas as published by OASIS.

To give users maximum flexibility in configuring and updating UBL code lists without changing the standard UBL schemas, UBL 2.0 introduced a two-phase validation model that has now been fully implemented in UBL 2.1. In the first phase, the UBL instance is checked for structure and vocabulary against a standard UBL schema using a generic schema validator (or custom-built software performing the same function). This is exactly the same procedure used for validation in UBL 1.0, except that the schemas do not contain hardwired code list values. Then in an added second validation (or verification) phase, code list values in the instance are checked against values obtained from external code list configuration files using an XSLT 1.0 processor driven by an XSLT 1.0 stylesheet. The default code list values assumed by the UBL 2.1 specification are expressed as data type qualifications in a file named `UBL-2.1-DefaultDTQ.xsl` located in the `val` directory, as described in more detail below. Publicly available tools were used to create the XSL file using the methodology described in the "Validation" section of [Customization], the *UBL Guidelines for Customization*.

Separating the checking of structure and vocabulary from the checking of code values allows trading partners to easily and precisely specify code list subsets and extensions and to apply them not just to individual UBL document types but also to particular elements and subtrees within UBL document instances. Another way to say this is that the the UBL code list methodology allows different versions of the same code list to be used in different document contexts. Thus, for example, a business in Canada might agree with a business in the United States to use a set of code list configuration files that allow the Buyer to be associated with either a U.S. state or a Canadian province but restrict the Seller to just U.S. states—that is, to apply a code list subset containing state and province codes in one place in a document instance and a different code list subset containing just state codes in another place in the instance.

D.2. Default Validation Setup

To facilitate the processing of UBL 2.0 instances using the two-phase method, an “out-of-the-box” collection of open-source software that can be used to demonstrate default validation of UBL 2.1 documents is included in the `val` directory of this release package. The validation harness assumes a Linux or Windows system with no currently installed XML or XSLT processing software.

The Java Runtime Environment (JRE) 1.5 or later is required to use the programs in the `val` directory; JRE versions below 1.5 will throw an error from the `xjparse.jar` module used to invoke the xerces schema parser. If necessary, download and install the latest JRE from the following location before continuing:

> http://www.java.com/en/download/manual.jsp

To demonstrate UBL 2.1 default validation:

1. Change to the `val` directory.

2. From within that directory, enter the test command

   `test.bat` (Windows)

   or

   `sh test.sh` (Linux)

   The output, which is explained in the next section, should resemble the output shown in Figure D.1 (the spacing has been manually adjusted to make the output easier to read).

   Figure D.1. Validation test output

```
############################################################
Validating order-test-good.xml
############################################################
============== Phase 1: XSD schema validation ==============
No schema validation errors.
============ Phase 2: XSLT code list validation ============
No code list validation errors.
############################################################
Validating order-test-bad1.xml
############################################################
============== Phase 1: XSD schema validation ==============
Attempting validating, namespace-aware parse
Error:file:///c:/d/ubl/2/val/order-test-bad1.xml:48:23:cvc-complex-type.2.4.a:
Invalid content was found starting with element 'cbc:ChannelCod'.
One of '{"urn:oasis:names:specification:ubl:schema:xsd:CommonBasicComponents-2":ChannelCode,
"urn:oasis:names:specification:ubl:schema:xsd:CommonBasicComponents-2":Channel,
"urn:oasis:names:specification:ubl:schema:xsd:CommonBasicComponents-2":Value}' is expected.
Parse succeeded (0.822) with 1 error and no warnings.
############################################################
Validating order-test-bad2.xml
############################################################
============== Phase 1: XSD schema validation ==============
No schema validation errors.
============ Phase 2: XSLT code list validation ============
Value supplied ' LA ' is unacceptable for codes identified by 'ChannelCodeType'
in the context: cbc:ChannelCode
Processing terminated by xsl:message at line 18
```

3. From within the `val` directory, you can now validate any UBL document against the UBL 2.1 schemas by executing commands of the form

   `validate <ubl-schema> <ubl-document>`

   where `<ubl-document>` is the path of a document to be validated and `<ubl-schema>` is the path of the UBL 2.1 schema for that document type (Order, Invoice, etc.). For example, the scripts `val/testsamples.bat` and `val/testsamples.sh` show this process being used to validate the sample XML instances in the `xml` directory.

D.3. Discussion of the Default Validation Test

The test output displayed above demonstrates the default validation process with three test files: a valid UBL Order (`order-test-good.xml`); a UBL Order containing a bad (misspelled) element (`order-test-bad1.xml`); and a UBL Order that is schema-valid but contains an illegal code list value (`order-test-bad2.xml`). The file `test.bat` (Windows) or `test.sh` (Linux) is used to run the script `validate.bat` or `validate.sh` against each of the test files.

The first run using `order-test-good.xml` demonstrates both phases of the default validation process running normally. In the first phase, a standard W3C Schema (XSD) validator, xerces, is invoked from `w3cschema.bat` (or `w3cschema.sh`) to validate the specified UBL document (`.xml`) against the specified UBL 2.1 runtime schema (`.xsd`). Since the input is a valid UBL Order, the output of the first phase simply indicates that the file is valid against the given Order schema.

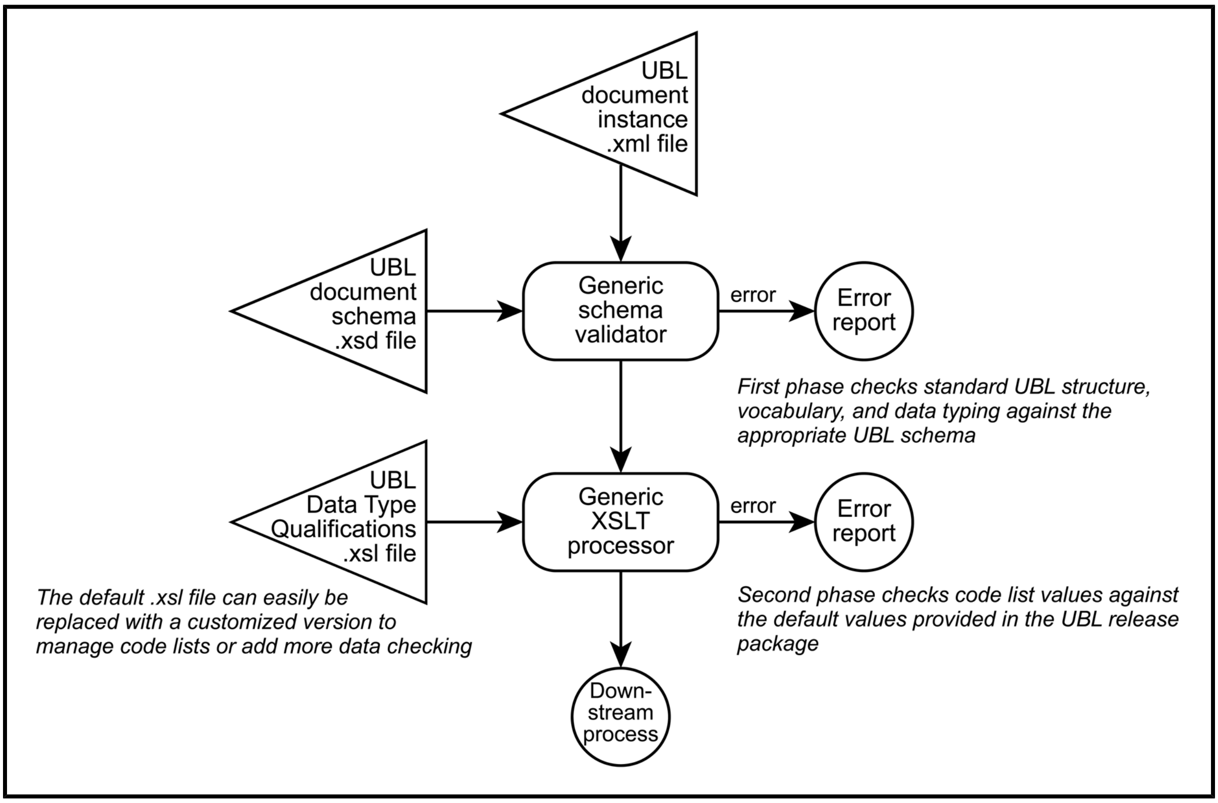
The second phase of validation uses a standard XSLT 1.0 engine, saxon, to verify that the values of various codes used in the UBL document to be tested (currency codes, packaging types, etc.) are valid in terms of the default UBL 2.1 code list values specified in `UBL-DefaultDTQ-2.1.xsl`. Here the output line "No code list validation errors" from the `validate` script indicates that the saxon run (invoked from `xslt.bat` or `xslt.sh`) finds no illegal code values in the document.

The second run shows what happens when the input document (`order-test-bad1.xml`) contains an actual structure or vocabulary error, in this case due to omission of the trailing "e" from the element named `cbc:ChannelCode`. When the xerces parser encounters the malformed element name, it emits the error message shown in the example, and the `validate` script reacts to a non-zero status code from `w3cschema.bat` (or `w3cschema.sh`) by terminating the validation process.

In the third run, the input document `order-test-bad2.xml` is structurally valid according to the Order schema, but it contains an illegal code list value (the ChannelCode "AL" for cell phone has been mistyped as "LA"). Thus it passes the first phase when tested against the schema but fails the second phase when tested against `UBL-DefaultDTQ-2.1.xsl`.

To summarize, input documents are checked in the first validation phase for correctness of structure and vocabulary, using the constraints expressed in the appropriate UBL schema, and then they are checked in the second phase for correctness of default code list values, using the default constraints expressed in the XSLT file `UBL-DefaultDTQ-2.1.xsl`. This process is illustrated in the following diagram.

Figure D.2. Two-phase Default UBL 2.1 Validation


UBL document instance .xml file
UBL document schema .xsd file
Generic schema validator
error
Error report
First phase checks standard UBL structure, vocabulary, and data typing against the appropriate UBL schema
UBL Data Type Qualifications .xsl file
Generic XSLT processor
error
Error report
The default .xsl file can easily be replaced with a customized version to manage code lists or add more data checking
Second phase checks code list values against the default values provided in the UBL release package
Down-stream process

It should be clear from the foregoing that the second phase of the default validation process can safely be omitted if it is considered unnecessary to check code list values. However, the reverse is not true; the second phase depends for correct operation on a prior check for structural validity, and therefore it will not give reliable results if run in the absence of the first (schema) validation phase.

D.4. Customizing the Default XSLT File

The validation framework provided in the val directory can be used to implement code list changes, define variant code lists to fit specific trading partner agreements, or associate different versions of the same code list with different parts of the same UBL document by substituting a custom process (be it XSLT or some other language or process) for the default UBL-DefaultDTQ-2.1.xsl provided in the UBL 2.1 distribution. This allows extensive code list management without the need to change the standard UBL 2.1 schemas. Schematron-based techniques for generating a custom XSLT file to take the place of UBL-DefaultDTQ-2.1.xsl are explained in [CVA] and [Customization]. See also Appendix E, *Data Type Qualifications in UBL (Informative)* for more about UBL data type qualifications.

Since XSLT is a very powerful general-purpose XML transformation tool, the same framework can be extended to perform fairly sophisticated business rule checking by manually coding additional logic into the XSLT file that drives the second validation phase. Such modification is beyond the scope of the customization methodologies associated specifically with UBL, but a business analyst willing to perform XSLT programming can use this mechanism to offload a large proportion of input filtering from the backend business application to a simpler input processing area. Additional XSLT scripts can be added to extract logical subtrees of incoming UBL documents for allocation to different downstream processes and to perform even more extensive front-end processing.

D.5. Sources for the Default Validation Framework

Components of several freely available software distributions were used to create the val directory. Sources are given below so that these components can be updated as later releases become available.

- The files resolver.jar and xercesImpl.jar are from the xerces-j 2.8.0 binary distribution at

  http://archive.apache.org/dist/xml/xerces-j/Xerces-J-bin.2.8.0.zip

- The file xjparse.jar (renamed from xjparse-1.0.jar) is from the xjparse 1.0 distribution at

  http://nwalsh.com/java/xjparse/

- The file saxon.jar is from the saxon 6.5.5 distribution at

  http://prdownloads.sourceforge.net/saxon/saxon6-5-5.zip

D.6. Code Lists Included in UBL 2.1

The code lists included in the UBL 2.1 distribution use an OASIS Standard XML format for code lists called [genericode]. Each code list in the distribution occupies its own genericode file. Documentation on the UBL code lists is contained in a generated report file:

cva/UBL-DefaultDTQ-2.1.html

The code list files in UBL 2.1 are divided into two subdirectories, cl/gc/default and cl/gc/special-purpose.

D.6.1. cl/gc/default

The code lists in the cl/gc/default directory contain the default code values represented in UBL-DefaultDTQ-2.1.xsl. A second-phase code list check using an unmodified version of the test setup from this distribution as described above will verify all occurrences of code values from these lists against the values specified in the cl/gc/default directory. These are the code lists expected to be used in most application contexts.

cl/gc/default/AllowanceChargeReasonCode-2.1.gc
cl/gc/default/BinaryObjectMimeCode-2.1.gc
cl/gc/default/ChannelCode-2.1.gc
cl/gc/default/CurrencyCode-2.1.gc
cl/gc/default/PackagingTypeCode-2.1.gc
cl/gc/default/PaymentMeansCode-2.1.gc
cl/gc/default/TransportEquipmentTypeCode-2.1.gc
cl/gc/default/TransportModeCode-2.1.gc
cl/gc/default/UnitOfMeasureCode-2.1.gc

D.6.2. cl/gc/special-purpose

This directory contains genericode versions of code lists that are used only in certain application contexts. They are not included in the `UBL-DefaultDTQ-2.1.xsl` file included in this distribution, but are provided here in the `cl/gc/special-purpose` directory to make them available for incorporation into custom XSLT scripts.

The files in this directory are as follows:

cl/gc/special-purpose/AdjustmentReasonCode-2.1.gc
cl/gc/special-purpose/CollaborationPriorityCode-2.1.gc
cl/gc/special-purpose/ComparisonDataSourceCode-2.1.gc
cl/gc/special-purpose/DataSourceCode-2.1.gc
cl/gc/special-purpose/DisplayTacticTypeCode-2.1.gc
cl/gc/special-purpose/ExceptionStatusCode-2.1.gc
cl/gc/special-purpose/ForecastPurposeCode-2.1.gc
cl/gc/special-purpose/ForecastTypeCode-2.1.gc
cl/gc/special-purpose/LanguageCode-2.1.gc
cl/gc/special-purpose/MiscellaneousEventTypeCode-2.1.gc
cl/gc/special-purpose/PerformanceMetricCriterionCode-2.1.gc
cl/gc/special-purpose/PromotionalEventTypeCode-2.1.gc
cl/gc/special-purpose/ResolutionCode-2.1.gc
cl/gc/special-purpose/RetailEventStatusCode-2.1.gc
cl/gc/special-purpose/RevisionStatusCode-2.1.gc
cl/gc/special-purpose/StatusCode-2.1.gc
cl/gc/special-purpose/SupplyChainActivityTypeCode-2.1.gc
cl/gc/special-purpose/ThresholdValueComparisonCode-2.1.gc
cl/gc/special-purpose/TimeFrequencyCode-2.1.gc
cl/gc/special-purpose/TransportationStatusCode-2.1.gc

Appendix E. Data Type Qualifications in UBL (Informative)

All UBL data types ultimately derive either from the UN/CEFACT Core Components Technical Specification [CCTS] Core Component Types (CCT) or from the W3C Schema specification [XSD2] itself; this derivation takes place in the UBL UDT module. The following table lists the CCTS 2.01 Core Component Types.

Table E.1. CCTS Unqualified Data Types

CCTS Data Type	Definition
Amount. Type	A number of monetary units specified in a currency where the unit of currency is explicit or implied.
Binary Object. Type	A set of finite-length sequences of binary octets.
Code. Type	A character string (letters, figures or symbols) that for brevity and/or language independence may be used to represent or replace a definitive value or text of an Attribute together with relevant supplementary information.
Date Time. Type	A particular point in the progression of time together with relevant supplementary information.
Identifier. Type	A character string to identify and distinguish uniquely, one instance of an object in an identification scheme from all other objects in the same scheme together with relevant supplementary information.
Indicator. Type	A list of two mutually exclusive Boolean values that express the only possible states of a Property.
Measure. Type	A numeric value determined by measuring an object along with the specified unit of measure.
Numeric. Type	Numeric information that is assigned or is determined by calculation, counting, or sequencing. It does not require a unit of quantity or unit of measure.
Quantity. Type	A counted number of non-monetary units possibly including fractions.
Text. Type	A character string (i.e. a finite set of characters) generally in the form of words of a language.

The UBL unqualified data types include the CCTS unqualified data types (named per UBL NDR) and a few others, as listed in the following table.

Table E.2. UBL Unqualified Data Types

UBL Unqualified Data Type	**Definition**
AmountType	A number of monetary units specified in a currency where the unit of the currency is explicit or implied.
BinaryObjectType	A set of finite-length sequences of binary octets.
GraphicType	A diagram, graph, mathematical curves, or similar representation.
PictureType	A diagram, graph, mathematical curves, or similar representation.
SoundType	A diagram, graph, mathematical curves, or similar representation.
VideoType	A diagram, graph, mathematical curves, or similar representation.
CodeType	A character string (letters, figures, or symbols) that for brevity and/or languange independence may be used to represent or replace a definitive value or text of an attribute together with relevant supplementary information.
DateTimeType	A particular point in the progression of time together with the relevant supplementary information.
DateType	One calendar day according the Gregorian calendar.
TimeType	The instance of time that occurs every day.
IdentifierType	A character string to identify and distinguish uniquely, one instance of an object in an identification scheme from all other objects in the same scheme together with relevant supplementary information.
IndicatorType	A list of two mutually exclusive Boolean values that express the only possible states of a property.
MeasureType	A numeric value determined by measuring an object along with the specified unit of measure.
NumericType	Numeric information that is assigned or is determined by calculation, counting, or sequencing. It does not require a unit of quantity or unit of measure.
ValueType	Numeric information that is assigned or is determined by calculation, counting, or sequencing. It does not require a unit of quantity or unit of measure.
PercentType	Numeric information that is assigned or is determined by calculation, counting, or sequencing. It does not require a unit of quantity or unit of measure.
RateType	Numeric information that is assigned or is determined by calculation, counting, or sequencing. It does not require a unit of quantity or unit of measuret.
QuantityType	A counted number of non-monetary units possibly including fractions.
TextType	A character string (i.e. a finite set of characters) generally in the form of words of a language.
NameType	A character string that consititues the distinctive designation of a person, place, thing or concept.

Some UBL BBIEs have data type qualifications based on the unqualified UBL types. These qualified types are all code types, and their definitions are the mechanism whereby a specific set of values is associated with each code.

UBL data type qualifications are expressed formally in an OASIS [CVA] (Context/Value Association) file contained in the cva directory of the 2.1 distribution. The specification of the CVA mechanism and format is maintained by the OASIS Code List Representation Technical Committee.

cva/UBL-DefaultDTQ-2.1.cva

A human-readable version is provided in an accompanying HTML file, which also serves as primary documentation on the UBL codes defined as qualified data types.

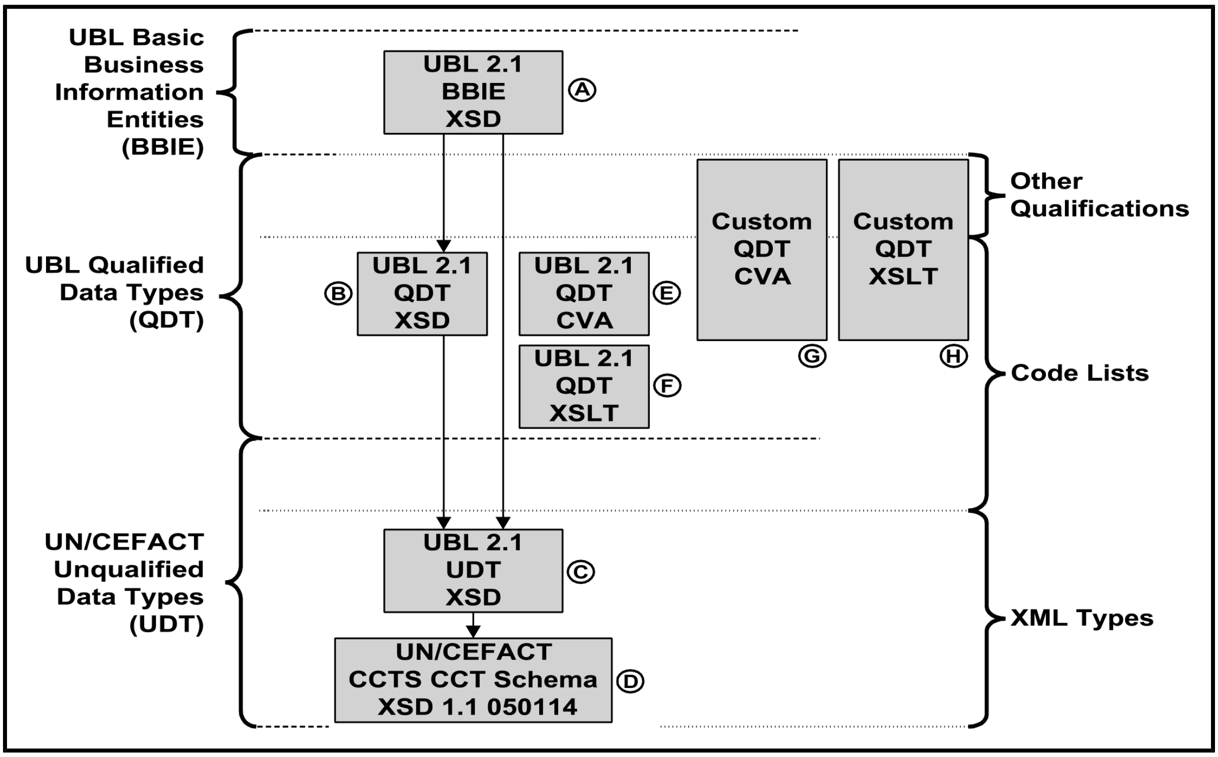
cva/UBL-DefaultDTQ-2.1.html

The `val` directory contains the predefined CVA associations compiled into an XSLT file, `UBL-DefaultDTQ-2.1.xsl`, which is used in the recommended two-phase validation process to perform a check of code list values. See Appendix D, *UBL 2.1 Code Lists and Two-phase Validation (Informative)* for a description of this process.

val/UBL-DefaultDTQ-2.1.xsl

The UBL 2.1 approach to data type qualification is illustrated in the following diagram.

Figure E.1. Data Type Qualification in UBL 2.1



UBL Basic Business Information Entities (BBIE)
UBL 2.1 BBIE XSD
A
Other Qualifications
Custom QDT CVA
Custom QDT XSLT
UBL Qualified Data Types (QDT)
B
UBL 2.1 QDT XSD
UBL 2.1 QDT CVA
E
G
H
Code Lists
UBL 2.1 QDT XSLT
F
UN/CEFACT Unqualified Data Types (UDT)
UBL 2.1 UDT XSD
C
XML Types
UN/CEFACT CCTS CCT Schema XSD 1.1 050114
D

In UBL 2.1, the schema library of common basic components (basic information entities or BBIEs, **(A)** in the diagram) is based on a combination of the data types defined in the file of UBL 2.1 qualified data types **(B)** and the data types defined in a file of UBL 2.1 unqualified data types **(C)**. The latter inherits the data type definitions in the UN/CEFACT CCTS CCT schema module Ver. 1.1 050114 **(D)**. The UBL 2.1 CVA file **(E)** controls the creation of the UBL 2.1 XSLT stylesheet **(F)** used in validation. While this XSLT file, `UBL-2.1-DefaultDTQ.xsl`, can, in theory, apply to qualified data type qualifications in general (such as field length restrictions and value range restrictions), the version of this file included in the UBL 2.1 release contains only code list values.

The two remaining boxes on the right in the diagram illustrate that users can add further data type qualifications if desired by preparing a custom CVA **(G)** and creating a custom XSLT file **(H)** to replace the default CVA and XSLT stylesheet provided in the UBL 2.1 distribution.

Users intending to prepare a custom CVA should note that `cva/UBL-DefaultDTQ-2.1.cva` contains relative URIs that expect the UBL 2.0 code lists from the UBL 2.0 Update Package in a sibling directory named `os-UBL-2.0`. This is irrelevant to users of the precompiled `val/UBL-DefaultDTQ-2.1.xsl` file contained in the UBL 2.1 package, but users wishing to create their own CVA file must first install the UBL 2.0 release and then the UBL 2.0 update. To properly install the update, first download and install the original UBL 2.0 release:

http://docs.oasis-open.org/ubl/os-UBL-2.0.zip

Then download and install the UBL 2.0 update:

http://docs.oasis-open.org/ubl/os-UBL-2.0-update-delta.zip

Complete installation instructions can be found in the update package. As indicated above, the `os-UBL-2.0` directory thus created must be a sibling to the directory created by installing the UBL 2.1 package.

Appendix F. Alternative Representations of the UBL 2.1 Schemas (Informative)

UBL 2.1 continues the practice, adopted at the beginning of the UBL effort, of creating its normative XML specifications using W3C Schema (XSD) syntax. Included in this release are two additional alternative specifications of the same content: a set of UBL 2.1 ASN.1 modules and a set of UBL 2.1 RELAX NG (compact syntax) schemas. These alternative representations are technically non-normative, but both are generated directly from the XSD and, with the exception of the UBL 2.1 digital signature extension (see Section 5, "UBL XAdES Extension for Digital Signatures"), both are intended to implement the same document instance constraints.

F.1. ASN.1 UBL 2.1 Specification

The UBL ASN.1 specification linked below provides an alternative schema definition for UBL documents in accordance with ITU-T X.680-X.693 [ASN.1]. The UBL ASN.1 specification defines the same UBL documents as the UBL XSD schemas that constitute the normative definitions of valid UBL documents. The UBL ASN.1 XML specification enables ASN.1 tools to be used for UBL transfers, and in conjunction with the ASN.1 Packed Encoding Rules, it provides a specification for an efficient binary encoding of UBL messages.

The zip archive below contains the ASN.1 modules corresponding to the UBL 2.1 document schemas as individual text files. The ASN.1 modules were created using a tool from OSS Nokalva [http://www.oss.com/] that conforms to ITU-T Recommendation X.694 | ISO/IEC 8825-5 for converting XSD Schema to ASN.1.

UBL 2.1 ASN.1 Modules
asn/ASN.1-UBL-2.1-text.zip

After conversion from XSD, the generated ASN.1 was formatted by the PrettyPrint tool at the ASN.1 Information Site [http://asn1.elibel.tm.fr/] to produce the following HTML documentation file.

UBL 2.1 ASN.1 Specification
asn/ASN.1-UBL-2.1.html

F.2. UBL 2.1 RELAX NG Schemas

[RELAX NG] (compact syntax) versions of the UBL schemas contributed by Crane Softwrights [http://cranesoftwrights.com/] and used here by permission are located in the `rnc` directory. The Crane package includes RELAX NG schemas for both UBL 2.0 and 2.1, as detailed in the related documentation.

rnc/readme-rnc.html

The UBL 2.1 RELAX NG schemas are made accessible separately as listed below. In this release (PRD1), these schemas do not support the signature extension described in Section 5, "UBL XAdES Extension for Digital Signatures".

ApplicationResponse
rnc/versions/UBL-ApplicationResponse-2.1.rnc

AttachedDocument
rnc/versions/UBL-AttachedDocument-2.1.rnc

AwardedNotification
rnc/versions/UBL-AwardedNotification-2.1.rnc

BillOfLading
rnc/versions/UBL-BillOfLading-2.1.rnc

CallForTenders
rnc/versions/UBL-CallForTenders-2.1.rnc

Catalogue
rnc/versions/UBL-Catalogue-2.1.rnc

CatalogueDeletion
rnc/versions/UBL-CatalogueDeletion-2.1.rnc

CatalogueItemSpecificationUpdate
rnc/versions/UBL-CatalogueItemSpecificationUpdate-2.1.rnc

CataloguePricingUpdate
rnc/versions/UBL-CataloguePricingUpdate-2.1.rnc

CatalogueRequest
rnc/versions/UBL-CatalogueRequest-2.1.rnc

CertificateOfOrigin
rnc/versions/UBL-CertificateOfOrigin-2.1.rnc

ContractAwardNotice
rnc/versions/UBL-ContractAwardNotice-2.1.rnc

ContractNotice
rnc/versions/UBL-ContractNotice-2.1.rnc

CreditNote
rnc/versions/UBL-CreditNote-2.1.rnc

DebitNote
rnc/versions/UBL-DebitNote-2.1.rnc

DespatchAdvice
rnc/versions/UBL-DespatchAdvice-2.1.rnc

DocumentStatus
rnc/versions/UBL-DocumentStatus-2.1.rnc

DocumentStatusRequest
rnc/versions/UBL-DocumentStatusRequest-2.1.rnc

ExceptionCriteria
rnc/versions/UBL-ExceptionCriteria-2.1.rnc

ExceptionNotification
rnc/versions/UBL-ExceptionNotification-2.1.rnc

Forecast
rnc/versions/UBL-Forecast-2.1.rnc

ForecastRevision
rnc/versions/UBL-ForecastRevision-2.1.rnc

ForwardingInstructions
rnc/versions/UBL-ForwardingInstructions-2.1.rnc

FreightInvoice
rnc/versions/UBL-FreightInvoice-2.1.rnc

GuaranteeCertificate
rnc/versions/UBL-GuaranteeCertificate-2.1.rnc

InstructionForReturns
rnc/versions/UBL-InstructionForReturns-2.1.rnc

InventoryReport
rnc/versions/UBL-InventoryReport-2.1.rnc

Invoice
rnc/versions/UBL-Invoice-2.1.rnc

ItemInformationRequest
rnc/versions/UBL-ItemInformationRequest-2.1.rnc

Order
rnc/versions/UBL-Order-2.1.rnc

OrderCancellation
rnc/versions/UBL-OrderCancellation-2.1.rnc

OrderChange
rnc/versions/UBL-OrderChange-2.1.rnc

OrderResponse
rnc/versions/UBL-OrderResponse-2.1.rnc

OrderResponseSimple
rnc/versions/UBL-OrderResponseSimple-2.1.rnc

PackingList
rnc/versions/UBL-PackingList-2.1.rnc

PerformanceHistory
rnc/versions/UBL-PerformanceHistory-2.1.rnc

PriorInformationNotice
rnc/versions/UBL-PriorInformationNotice-2.1.rnc

ProductActivity
rnc/versions/UBL-ProductActivity-2.1.rnc

Quotation
rnc/versions/UBL-Quotation-2.1.rnc

ReceiptAdvice
rnc/versions/UBL-ReceiptAdvice-2.1.rnc

Reminder
rnc/versions/UBL-Reminder-2.1.rnc

RemittanceAdvice
rnc/versions/UBL-RemittanceAdvice-2.1.rnc

RequestForQuotation
rnc/versions/UBL-RequestForQuotation-2.1.rnc

RetailEvent
rnc/versions/UBL-RetailEvent-2.1.rnc

SelfBilledCreditNote
rnc/versions/UBL-SelfBilledCreditNote-2.1.rnc

SelfBilledInvoice
rnc/versions/UBL-SelfBilledInvoice-2.1.rnc

Statement
rnc/versions/UBL-Statement-2.1.rnc

StockAvailabilityReport
rnc/versions/UBL-StockAvailabilityReport-2.1.rnc

Tender
rnc/versions/UBL-Tender-2.1.rnc

TenderReceipt
rnc/versions/UBL-TenderReceipt-2.1.rnc

TendererQualification
rnc/versions/UBL-TendererQualification-2.1.rnc

TendererQualificationResponse
rnc/versions/UBL-TendererQualificationResponse-2.1.rnc

TradeItemLocationProfile
rnc/versions/UBL-TradeItemLocationProfile-2.1.rnc

TransportExecutionPlan
rnc/versions/UBL-TransportExecutionPlan-2.1.rnc

TransportExecutionStatus
rnc/versions/UBL-TransportExecutionStatus-2.1.rnc

TransportOperationStatus
rnc/versions/UBL-TransportOperationStatus-2.1.rnc

TransportationStatus
rnc/versions/UBL-TransportationStatus-2.1.rnc

UnawardedNotification
rnc/versions/UBL-UnawardedNotification-2.1.rnc

UtilityStatement
rnc/versions/UBL-UtilityStatement-2.1.rnc

Waybill
rnc/versions/UBL-Waybill-2.1.rnc

Appendix G. UBL 2.1 Naming and Design Rules (Informative)

The XML Naming and Design Rules (NDRs) used in creating the UBL schemas in this specification are given in the checklist below. The entire NDR document (including explanatory prose) will be released following publication of UBL 2.1.

The checklist included in this release (UBL 2.1 PRD1) contains some known errors that will be corrected in PRD2.

The rule name categories are presented in alphabetical sequence, as follows:

Attribute Declaration Rules (ATD)
Code List Rules (CDL)
ComplexType Definition Rules (CTD)
ComplexType Naming Rules (CTN)
Documentation Rules (DOC)
Element Declaration Rules (ELD)
Element Naming Rules (ELN)
General Naming Rules (GNR)
General Type Definition Rules (GTD)
General XML Schema Rules (GXS)
Instance Document Rules (IND)
Modeling Constraints Rules (MDC)
Naming Constraints Rules (NMC)
Namespace Rules (NMS)
Root Element Declaration Rules (RED)
Schema Structure Modularity Rules (SSM)
Standards Adherence Rules (STA)
Versioning Rules (VER)

Code List Rules	
CDL1	All UBL codes MUST be part of a UBL or externally maintained code list.
CDL2	The UBL Library SHOULD identify and use external standardized code lists rather than develop its own UBL-native code lists.
CDL3	The UBL Library MAY design and use an internally maintained code list where an existing externally maintained code list needs to be extended, or where no suitable externally maintained code list exists.

ComplexType Definition rules	
CTD1	For every class identified in the UBL model, a named xsd:complexType MUST be defined.
CTD2	Every CCTS ABIE xsd:complexType definition content model MUST contain an xsd:sequence element containing the appropriate global element declarations.
CTD3	Every CCTS BBIE Property xsd:complexType definition content model MUST contain an xsd:simpleContent element.
CTD4	Every CCTS BBIE Property xsd:complexType content model xsd:simpleContent element MUST consist of an xsd:extension element.
CTD25	For every CCTS BBIE Property identified in the UBL model, a named xsd:complexType MUST be defined.
CTD26	Every CCTS BBIE Property xsd:complexType xsd:base attribute value MUST be the UBL Unqualified Datatype.

ComplexType Definition rules	
CTD27	Every BBIE property with the representation term Code MUST be based on the UBL unqualified code data type.

Complex Type Naming rules	
CTN1	A UBL xsd:complexType name based on a CCTS ABIE MUST be the CCTS Dictionary Entry Name with the separators removed and with the "Details" suffix replaced with "Type".
CTN2	A UBL xsd:complexType name based on a CCTS BBIE Property MUST be the CCTS Dictionary Entry Name shared Property Term and its qualifiers and the Representation Term of the BBIE with the separators removed and with the "Type" suffix appended after the Representation Term.
CTN6	A UBL xsd:complexType name based on a CCTS BBIE Property and with a CCTS BBIE Representation Term of "Text" MUST have the word "Text" removed from the end of its name.
CTN7	A UBL xsd:complexType name based on a CCTS BBIE Property and with a CCTS BBIE Representation Term of "Identifier" MUST replace "Identifier" with "ID" at the end of its name.
CTN8	A UBL xsd:complexType name based on a CCTS BBIE Property MUST remove all duplication of words that occurs as a result of duplicate Property Terms and Representation Terms.

Documentation rules	
DOC1	The xsd:documentation element for every data type MUST contain a set of annotations in the following order (as defined in CCTS Section 7):
DOC2	A datatype definition MAY contain one or more Content Component Restrictions to provide additional information on the relationship between the datatype and its corresponding Core Component Type. If used, the Content Component Restrictions MUST contain a set of annotations in the following order:
DOC3	A datatype definition MAY contain one or more Supplementary Component Restrictions to provide additional information on the relationship between the datatype and its corresponding Core Component Type. If used, the Supplementary Component Restrictions MUST contain a set of annotations in the following order:
DOC4	The xsd:documentation element for every BBIE MUST contain a set of annotations in the following order:
DOC5	The xsd:documentation element for every ABIE MUST contain a set of annotations in the following order:
DOC6	The xsd:documentation element for every ASBIE element declaration MUST contain a set of annotations in the following order:
DOC8	The xsd:documentation element for every Supplementary Component attribute declaration MUST contain a set of annotations in the following order:
DOC9	The xsd:documentation element for every Supplementary Component attribute declaration containing restrictions MUST include the following additional information appended to the information required by DOC8:

Element Declaration rules	
ELD2	All element declarations MUST be global.
ELD3	For every class and property identified in the UBL model, a global element bound to the corresponding xsd:complexType MUST be declared.
ELD4	When a CCTS ASBIE is unqualified, it is bound via reference to the global CCTS ABIE element with which it is associated.

Element Declaration rules	
ELD7	Empty elements MUST not be declared, except in the case of extension where the UBL Extensions element is used.
ELD11	When a CCTS ASBIE is qualified, a new element MUST be declared and bound to the xsd:complexType of its associated CCTS ABIE.
ELD12	The UBLExtensions element MUST be declared as the first child of the document element with xsd:minOccurs="0".
ELD13	The UBLProfileID element MUST be declared immediately following the UBL Extensions element with xsd:minOccurs="0".
ELD14	The UBLSubsetID element MUST be declared immediately following the UBLProfileID element with xsd:minOccurs="0".

Element Naming rules	
ELN1	A UBL global element name based on a CCTS ABIE MUST be the same as the name of the corresponding xsd:complexType to which it is bound, with the word "Type" removed.
ELN2	A UBL global element name based on a CCTS BBIE Property MUST be the same as the name of the corresponding xsd:complexType to which it is bound, with the word "Type" removed.
ELN3	A UBL global element name based on a CCTS ASBIE MUST be the CCTS ASBIE Dictionary Entry Name Property Term and its qualifiers and the Object Class Term and qualifiers of its associated CCTS ABIE. All CCTS Dictionary Entry Name separators MUST be removed.

General Naming rules	
GNR1	UBL XML element and type names MUST be in the English language, using the primary English spellings provided in the Oxford English Dictionary.
GNR2	UBL XML element and type names MUST be consistently derived from CCTS conformant Dictionary Entry Names.
GNR3	UBL XML element and type names constructed from CCTS Dictionary Entry Names MUST NOT include periods, spaces, other separators, or characters not allowed by XSD.
GNR4	UBL XML element names and simple and complex type names MUST NOT use acronyms, abbreviations, or other word truncations, except those in the list of exceptions maintained and published by the UBL TC.
GNR6	The acronyms and abbreviations listed in the UBL-approved list MUST always be used in place of the word or phrase they represent.
GNR7	UBL XML element and type names MUST be in singular form unless the concept itself is plural.
GNR8	The UpperCamelCase (UCC) convention MUST be used for naming elements and types.
GNR9	The lowerCamelCase (LCC) convention MUST be used for naming attributes.
GNR10	Acronyms and abbreviations at the beginning of an attribute name MUST appear in all lower case. Acronyms and abbreviations elsewhere in an attribute name MUST appear in upper case.
GNR11	Acronyms and abbreviations MUST appear in all upper case for all element and type names.

General Type Definition Rules	
GTD1	All types MUST be named.
GTD2	The predefined XML schema type xsd:anyType MUST NOT be used.

General XML Schema Rules	
GXS1	Except in the case of extension, where the "UBL Extensions" element is used, UBL schemas SHOULD conform to the following physical layout as applicable: See .
GXS2	UBL MUST provide two schemas for each transaction. One normative schema shall be fully annotated. One non-normative schema shall be a run-time schema devoid of documentation.
GXS3	Built-in xsd:simpleTypes SHOULD be used wherever possible.
GXS4	All XSD constructs in UBL schema and schema modules MUST contain the following namespace declaration on the xsd:schema element:
GXS5	The xsd:substitutionGroup feature MUST NOT be used.
GXS6	The xsd:final attribute MUST be used to control extensions where there is a desire to prohibit further extensions.
GXS7	xsd:notation MUST NOT be used.
GXS8	xsd:all MUST NOT be used.
GXS9	The xsd:choice element SHOULD NOT be used where customization and extensibility are a concern.
GXS10	xsd:include can only be used when the including schema is in the same namespace as the included schema.
GXS11	The xsd:union technique MUST NOT be used except for code lists.
GXS12	UBL schemas SHOULD NOT use xsd:appinfo. If used, xsd:appinfo MUST be used only to convey non-normative information.
GXS15	Each xsd:schemaLocation attribute declaration MUST contain a system-resolvable URL, which at the time of release from OASIS shall be a relative URL referencing the location of the schema or schema module in the release package.
GXS16	The built in xsd:nillable attribute MUST NOT be used for any UBL declared element.
GXS14	xsd:any MUST NOT be used except within the ExtensionContentType type definition, and with xsd:processContents= "skip" for non-UBL namespaces.
GXS13	Complex type extension or restriction MAY be used where appropriate.

Modelling constraint rules	
MDC0	The sequence of the business information entities that is expressed in the UBL model MUST be preserved in the schema.
MDC1	UBL libraries and schemas MUST only use CCTS Core Component Types, except in the case of extension, where the UBLExtensions element is used.
MDC2	XML mixed content MUST NOT be used except where contained in an xsd:documentation element.

Naming constraint rules	
NMC1	Each Dictionary Entry Name MUST define one and only one fully qualified path (FQP) for an element or attribute.

Namespace Rules	
NMS1	Every UBL-defined or -used schema module, except internal schema modules, MUST declare a namespace using the xsd:targetNamespace attribute.
NMS2	Every UBL-defined or -used major version schema set MUST have its own unique namespace.
NMS3	UBL namespaces MUST only contain UBL developed schema modules.

Namespace Rules	
NMS4	The namespace names for UBL schemas holding committee draft status MUST be of the form
NMS5	The namespace names for UBL schemas holding OASIS Standard status MUST be of the form
NMS6	UBL published namespaces MUST never be changed.
NMS7	The UBL Common Aggregate Components schema module MUST reside in its own namespace.
NMS8	The UBL Common Aggregate Components schema module namespace MUST be represented by the namespace prefix "cac" when referenced in other schemas.
NMS9	The UBL Common Basic Components schema module MUST reside in its own namespace.
NMS10	The UBL Common Basic Components schema module namespace MUST be represented by the namespace prefix "cbc" when referenced in other schemas.
NMS15	The UBL Qualified Datatypes schema module MUST reside in its own namespace.
NMS16	The UBL Qualified Datatypes schema module namespace MUST be represented by the namespace prefix "qdt" when referenced in other schemas.
NMS18	The CommonExtensionComponents schema module namespace MUST be represented by the namespace prefix "ext" when referenced in other schemas.
NMS19	The CCTS Core Component Type schema module must be represented by the namespace prefix "ccts-cct".
NMS20	The UBL Unqualified Datatypes schema module namespace MUST be represented by the prefix "udt" when referenced in other schemas.

Root element declaration rules	
RED2	The root element MUST be the only global element declared in the document schema.

Schema structure modularity rules	
SSM1	UBL schema expressions MAY be split into multiple schema modules.
SSM2	A schema in one UBL namespace that is dependent upon type definitions or element declarations in another schema namespace MUST only import that schema.
SSM3	A schema in one UBL namespace that is dependent upon type definitions or element declarations defined in another schema namespace MUST NOT import the internal schema modules of that schema.
SSM6	All UBL internal schema modules MUST be in the same namespace as their corresponding document schema.
SSM7	Each UBL internal schema module MUST be named <ParentSchemaModuleName><InternalSchemaModuleFunction>
SSM8	UBL schema modules MAY be created for reusable components.
SSM9	A schema module defining all UBL Common Aggregate Components MUST be created.
SSM10	The UBL Common Aggregate Components schema module MUST be identified as CommonAggregateComponents in the document name within the schema header.
SSM11	A schema module defining all UBL Common Basic Components MUST be created.
SSM12	The UBL Common Basic Components schema module MUST be identified as CommonBasicComponents in the document name within the schema header.
SSM18	A schema module without any declarations must exist.
SSM19	The UBL Qualified Datatypes schema module MUST be identified as QualifiedDatatypes in the document name in the schema header.

Schema structure modularity rules	
SSM21	The UBL extension schema module MUST be identified as CommonExtensionComponents in the document name within the schema header.
SSM22	The UBL Qualified Datatypes schema module MUST import the UBL Unqualified Datatypes schema module.

Versioning rules	
VER2	Every UBL schema module major version MUST have an RFC 3121 document-id of the form
VER4	Every minor version release of a UBL schema module MUST have a document-id of the form
VER5	For UBL minor version changes, the namespace name MUST not change.
VER6	Every UBL schema module major version number MUST be a sequentially assigned integer greater than zero.
VER7	Every UBL schema module minor version number MUST be a sequentially assigned, non-negative integer.
VER12	Every major version release of a UBL schema module MUST capture its version number in the xsd:version attribute of the xsd:schema element in the form
VER14	Every minor version release of a UBL schema module MUST capture its version information in the xsd:version attribute in the form
VER15	Every UBL document schema MUST declare an optional element named UBLVersionID immediately following the optional UBL Extensions element.

Appendix H. Acknowledgements

The OASIS UBL Technical Committee thanks Altova for its contribution of XML Spy licenses for use in UBL schema design; Sparx Systems for its contribution of Enterprise Architect licenses for use in developing UML content models; Syncro Soft for its contribution of oXygen licenses used in DocBook authoring of UBL documentation; RenderX for its contribution of XEP licenses used in generating PDF documents from DocBook originals; SRDC for developing iSurf eDoCreator (supported by European Commission FP7 ICT-213031 iSurf Project) and providing technical assistance in its use as an online repository and editing environment for UBL 2.1 document models and the generation of UBL schemas; OSS Nokalva for its contribution of ASN.1 UBL 2.1 specifications; and Crane Softwrights for permission to include a copy of their publicly-available HTML reports and UBL 2.1 RELAX NG schema resources.