

**This file, UBL-index-2.0.pdf, is automatically generated from UBL-index-2.0.html. The UBL-index-2.0.pdf file is included to comply with a procedural requirement of the current OASIS Technical Committee process and has no other function. It has no practical purpose and should be ignored. Please do not submit comments relating to the formatting or any other aspect of the UBL-index-2.0.pdf file. The hypertext UBL-index-2.0.html and the resources linked therefrom constitute the UBL 2.0 specification.**



# Universal Business Language 2.0 — Second Public Review Draft

## Publication Date

20 July 2006

## Document identifier

prd2-UBL-2.0

## Editorial Status

Second public review draft

## Location

<http://docs.oasis-open.org/ubl/prd2-UBL-2.0/>

## Downloadable Package Location

<http://docs.oasis-open.org/ubl/prd2-UBL-2.0.zip>

## Editors

Jon Bosak, G. Ken Holman, and Tim McGrath

## **Contributors**

Members of the OASIS UBL Technical Committee

## **Abstract**

This specification defines the Universal Business Language version 2.0.

## **Status**

This document is a draft of the OASIS Universal Business Language (UBL) Technical Committee. The UBL TC invites interested parties to comment on this release using the comment link on the UBL TC web page:

[http://www.oasis-open.org/committees/comments/form.php?wg\\_abbrev=UBL](http://www.oasis-open.org/committees/comments/form.php?wg_abbrev=UBL)

For legal reasons, only those comments that are submitted through the web page above can be considered in revising the current draft. Since the comment input form receives large amounts of spam, however, it is strongly recommended that comments submitted through the form above be copied to the UBL TC chair ([jon.bosak@sun.com](mailto:jon.bosak@sun.com)) and vice chair ([tmcgrath@portcomm.com.au](mailto:tmcgrath@portcomm.com.au)) to ensure that they are properly noted.

*Note that the purpose of this public review draft is to solicit input from the UBL user community. It will undergo further revision as a result of that input. Consequently, the current draft should not be implemented except for testing purposes.*

See [Appendix A: Release Notes](#) for more information regarding this release package.

## **Viewing this Document**

Older web browsers such as Internet Explorer may not correctly support the CSS styles and XHTML markup used for this document. For best results, the [Firefox 1.5](#) and [Opera 8.5](#) (or later) browsers are recommended.

# **Table of Contents**

[1 Introduction](#)

[2 Normative References](#)

[3 Terms and Definitions](#)

[4 Symbols and Abbreviations](#)

[5 UBL 2.0 Process Context](#)

[6 UBL 2.0 Schemas](#)

[Appendix A \(Informative\): Release Notes](#)

[Appendix B \(Informative\): Upgrading from UBL 1.0 to UBL 2.0](#)

[Appendix C \(Informative\): UBL 2.0 Data Models](#)

[Appendix D \(Informative\): UBL 2.0 Code Lists and Two-phase Validation](#)

[Appendix E \(Informative\): UBL 2.0 Naming and Design Rules](#)

[Appendix F \(Informative\): ASN.1](#)

[Appendix G: Notices](#)

# 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 201C;Address,201D; 201C;Item,201D; and 201C;Payment201D; 2014; the common data elements of everyday business documents.
- A set of XML schemas for common business documents such as 201C;Order,201D; 201C;Despatch Advice,201D; and 201C;Invoice201D; 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.

## Acknowledgements

The OASIS UBL TC 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, and GEFEG for its contribution of EDIFIX and technical expertise in the generation and quality review of UBL schemas.

## 2 Normative References

[ASN.1] ITU-T X.680-X.683: Abstract Syntax Notation One (ASN.1); ITU-T X.690-X.693: ASN.1 encoding rules

<http://www.itu.int/ITU-T/studygroups/com17/languages/X.680-X.693-0207w.zip>

<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 (ebCCTS)

<http://www.oasis-open.org/committees/download.php/6232/CEFACT-CCTS-Version-2pt01.zip>

[ISO11179] ISO/IEC 11179-1:1999 Information technology 2014; Specification and standardization of data elements 2014; 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](http://www.oasis-open.org/committees/download.php/6233/c002349_ISO_IEC_11179-1_1999%28E%29.pdf)

[RFC2119] Key words for use in RFCs to Indicate Requirement Levels

<http://www.faqs.org/rfcs/rfc2119.html>

<http://www.oasis-open.org/committees/download.php/6244/rfc2119.txt.pdf>

[SCH] Document Schema Definition Languages (DSDL) - Part 3: Rule-based validation (Schematron)

<http://www.iso.ch/iso/en/CatalogueDetailPage.CatalogueDetail?CSNUMBER=40833>

<http://www.schematron.com>

[UML] Unified Modeling Language Version 1.5 (formal/03-03-01)

<http://www.omg.org/docs/formal/03-03-01.pdf>

<http://www.oasis-open.org/committees/download.php/6240/03-03-01.zip>

[XML] Extensible Markup Language (XML) 1.0 (Second Edition), W3C Recommendation 6 October 2000

<http://www.w3.org/TR/2000/REC-xml-20001006>

<http://www.oasis-open.org/committees/download.php/6241/REC-xml-20001006.pdf>

[XSD1] XML Schema Part 1: Structures, W3C Recommendation 2 May 2001

<http://www.w3.org/TR/xmlschema-1/>

<http://www.oasis-open.org/committees/download.php/6248/xsd1.html>

[XSD2] XML Schema Part 2: Datatypes, W3C Recommendation 2 May 2001

<http://www.w3.org/TR/xmlschema-2/>

<http://www.oasis-open.org/committees/download.php/6247/xsd2.html>

[XSLT] XSL Transformations (XSLT) Version 1.0, W3C Recommendation 16 November 1999

<http://www.w3.org/TR/1999/REC-xslt-19991116>

<http://www.oasis-open.org/committees/download.php/18891/REC-xslt-19991116.html>

## 3 Terms and Definitions

### Assembly model

A tree-structured model of ABIEs that can be implemented as a document schema.

### Class diagram

A graphical notation used by [UML] to describe the static structure of a system, including object classes and their attributes and associations.

### Context

The circumstance or events that form the environment within which something exists or takes place.

### Document

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

### Spreadsheet model

A representation of an assembly model in tabular form.

## 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 [\[RFC2119\]](#).

# 4 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

## **CV2**

Credit Card Verification Numbering System

## **EDI**

Electronic Data Interchange

## **ISO**

International Organization for Standardization

## **NDR**

UBL Naming and Design Rules (see Appendix E)

## **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**

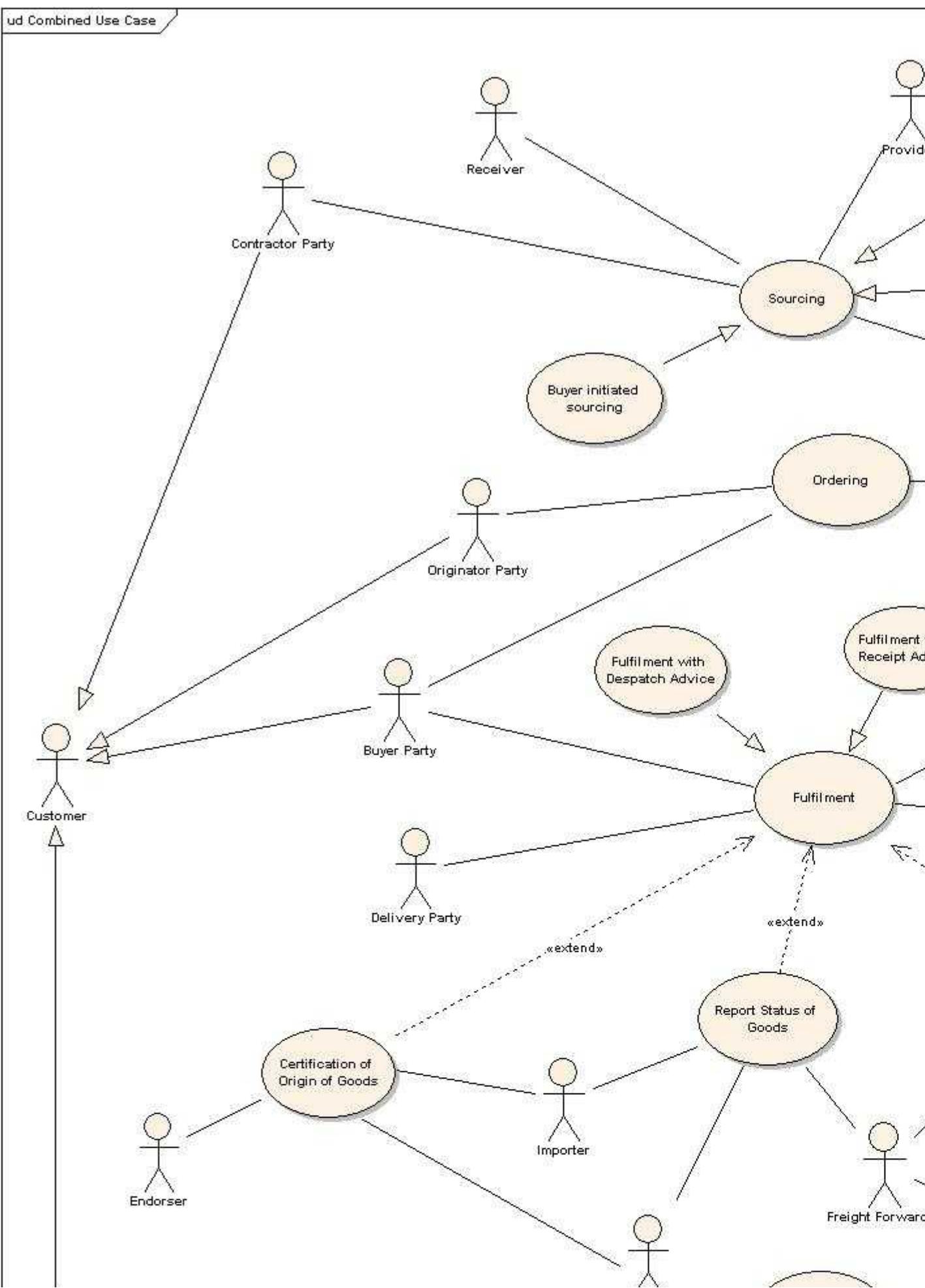
W3C XML Schema Language [\[XSD1\]](#) [\[XSD2\]](#)

# 5 Context of Use

This section defines the business process and contextual rules and requirements for the exchange of document types included in the UBL 2.0 release.

These processes (and the business rules associated with them) define a context for the use of the documents identified. 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 2.0 extends the order-to-invoice processs of UBL 1.0 to cover sourcing-to-payment and includes the commercial collaborations for international trade. The following use case diagram defines the scope of UBL 2.0 documents.



## **Figure 1. Use Case covered by UBL 2.0**

While this section describes the business rules and choreography of the generic process and the context played by each of the UBL 2.0 document types in that process, it is important to stress that these are indicative and demonstrative scenarios only and are not intended to limit the uses to which UBL documents can be put.

Also, it is important to note that the UBL 2.0 library is designed to support the construction of a wide variety of document types beyond those provided in the 2.0 package. It is expected that implementers will develop their own customized document types and that other UBL document types will be added as the library evolves.

## **5.1 General Business Rules**

This section describes the requirements and general business rules that are assumed for collaborations and document exchanges in UBL 2.0.

### **5.1.1 Items**

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

### **5.1.2 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.

### **5.1.3 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.

### **5.1.4 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 Base Price, in which case it is not specified. This makes a detailed response from the Seller necessary; see Order Response.

### **5.1.5 Hazardous Items**

Although ordered items may include Hazardous items, it is not necessary to specify related information 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.

### **5.1.6 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.

## 5.1.7 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 these textual components should not be in the same language.

## 5.2 Business Processes

The UBL 2.0 documents and library are designed to support the typical supply chain processes outlined in Figure 1.

The following sections describe each business collaboration in more detail. But first we should explain the roles that the parties involved in these processes may perform.

### 5.2.1 Party Roles

In the UBL procurement processes there are two main actors representing the key organizations or individuals involved in the process, Customer and Supplier. Each of these may play various roles. Processes may also involve supplementary roles which may be provided by different parties.

The actual role undertaken is dependant 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 transportation process is actually equal to the Despatch Party or Seller in procurement depends on different business cases.

The following table contains a description of the proposed procurement roles for the actor known as “Party”.

Actor	Role	Description	Example	Synonyms	Sends	Receives
Customer	Originator	<p>The party that had the original demand for the goods and/or services and therefore initiated the procurement transaction.</p> <p>The Originator participates in pre-ordering activity either through RFQ and Quotation or by receiving a Quotation as a response to a</p>	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

Actor	Role	Description	Example	Synonyms	Sends	Receives
		<p>punchout transaction on a marketplace or Seller's website.</p> <p>If the Originator subsequently places an Order, the Originator adopts the role of Buyer.</p> <p>The Originator is typically the contact point for queries regarding the original requirement and can be referred to in an Order Change, Order Cancellation, or Order Response.</p>				
Customer	Buyer	<p>The party that purchases the goods or services on behalf of the Originator.</p> <p>The Buyer can be referred to in Order Response, Despatch Advice, Invoice, Self Billed Invoice, Credit Note, and Account Statement.</p>	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, Despatch Advice, Invoice, Self Billed Invoice, Credit Note, and Account Statement
Customer	Delivery	<p>The party to whom goods should be delivered.</p> <p>The Delivery Party can be the same as the Originator.</p> <p>The Delivery Party must be referred to at line</p>	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, Invoice, Self Billed Invoice, Credit Note, and Account Statement

Actor	Role	Description	Example	Synonyms	Sends	Receives
		<p>item level in RFQ, Quotation, Order, Order change, Order Cancellation, and Order Response.</p> <p>The Delivery Party may be referred to at line level in Invoice, Self Billed Invoice, Credit Note, and Debit Note.</p> <p>The Delivery Party may be stipulated in a transport contract.</p>				
Customer	Accounting Customer	<p>The party responsible for making settlement relating to a purchase and resolving billing issues using a Debit Note.</p> <p>The Accounting Customer must be referred to in an Order and may be referred to in an Order Response.</p> <p>In a Self Billing scenario, the Accounting Customer is responsible for calculating and issuing tax invoices.</p>	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.	<p>Invoicee, Accounts Payable</p>	<p>In a traditional Billing scenario: Debit Note, Account Response, and Remittance Advice</p> <p>In a Self Billing scenario: Self Billed Invoice, Self Billed Credit Note, and Remittance Advice</p>	<p>In a traditional Billing scenario: Invoice, Note, and Statement</p> <p>In a Self Billing scenario: Note, Response, Statement, and Account</p>
Supplier	Seller	<p>The party responsible for handling Originator and Buyer services.</p> <p>The Seller party is</p>	The organization that sells wheelchairs to municipalities.	<p>Sales Point, Provider, Customer Manager</p>	<p>Quotation, Order Response, Order Simple, Catalogue, Catalogue</p>	<p>RFQ, Order Request, Order Change, Order Cancellation, Catalogue, Catalogue</p>

Actor	Role	Description	Example	Synonyms	Sends	Receives
		<p>legally responsible for providing the goods to the Buyer.</p> <p>The Seller party receives and quotes against RFQs and may provide information to the Buyer's requisitioning process through Catalogues and Quotations.</p>			Deletion, Catalogue Item Specification Update, Catalogue Pricing Update	
Supplier	Despatch	<p>The party where goods are to be collected from.</p> <p>The Despatch Party may be stipulated in a transport contract.</p>	<p>The wheelchair Supplier may store chairs at a local warehouse. The warehouse will actually despatch the chair to the Delivery Party.</p> <p>The local warehouse is then the Despatch Party.</p>	Despatch Point, Shipper, Sender	Despatch Advice	Receipt Advice
Supplier	Accounting Supplier	The party who claims the payment and is responsible for resolving billing issues and arranging settlement.	<p>There are cases where the Accounting Supplier is not the Seller party. For example, factoring, where the invoicing is outsourced to another company.</p>	Accounts Receivable, Invoice Issuer	<p>In a traditional Billing scenario: Invoice, Credit Note, and Statement of Account</p> <p>In a Self Billing scenario: Credit Note, Account Response and Statement of Account</p>	<p>In a traditional Billing scenario: Debit Note, Account Response and Remittance Advice</p> <p>In a Self Billing scenario: Billed In, Self Bill, Credit Note, Remittance Advice</p>
Supplier	Payee	The party to whom the Invoice is paid.	The Accounting Supplier may not be the party to be	Accounts Receivable		Remittance Advice

Actor	Role	Description	Example	Synonyms	Sends	Receives
			paid due to changes in the organization, e.g., a company merger.			
Party	Catalogue Managing	The party receiving a catalogue.  Catalogue items may never be ordered, so the recipient of the catalogue is not an Originator or a Buyer.	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 Specific Update, Catalogue Pricing U
Party	Information Content Owner	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.			
Party	Receiver	The party receiving a document.	A marketplace may receive an Application Response.			Application Response
Party	Sender	The party sending a document.	A marketplace may send an Application Response.		Application Response	
Party	Consignor	The party where goods are to be collected from.  The Consignor may be stipulated in a transport contract.  The Consignor will receive an invoice and make payments for the transport service provided.	The wheelchair Supplier may store chairs at 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	Despatch Point, Shipper, Sender	Forwarding Instruction, Packing List	Bill of L Waybill, Invoice

Actor	Role	Description	Example	Synonyms	Sends	Receives
			Forwarder.  The consignor will pay the freight forwarder for shipping the wheelchair to the destination.			
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	Forwarding Instruction, Freight Invoice	Bill of L Waybill
Party	Freight Forwarder	<p>The party arranging the carriage of goods, including connected services and/or associated formalities, on behalf of a Consignor or Consignee.</p> <p>The Freight Forwarder may also be the Carrier.</p> <p>The Freight Forwarder will create an invoice and bill to the consignor for the transportation service provided.</p>	<p>The Consignor may have a contract with this Freight Forwarder, which is a transport services provider, to arrange all their transport needs. This Freight Forwarder may then engage the Airline to transport the wheelchair. In this case, the Freight Forwarder is still the transport services provider while the Airline becomes the Carrier.</p> <p>The Freight Forwarder will then bill the consignor for shipping the wheelchairs.</p>	Shipping Agent, Broker, Courier	Forwarding Instruction, Freight Invoice	Bill of L Waybill, List
Party	Carrier	The party providing physical transport services.	The Freight Forwarder may engage the Airline to deliver the wheelchair. The Airline is	Freight Haulier, Shipper, Ships Agent, Shipping Company, Airline, Rail Operator, Road	Bill of Lading, Waybill	Forwarding Instruction

<b>Actor</b>	<b>Role</b>	<b>Description</b>	<b>Example</b>	<b>Synonyms</b>	<b>Sends</b>	<b>Receives</b>
			then the Carrier and delivers the chair to the Delivery Party.	Haulier		
Party	Exporter	The party supplies goods through the international purchase.	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 receiving a consignment of goods as stipulated in the transport contract.	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
Party	Freight Forwarder	The Freight Forwarder will create an Invoice and bill to the consignor for the transportation service provided.	Freight Forwarder is a transport services provider who arranges all the transport needed for transporting the wheelchairs. The freight forwarder will then bill the consignor for shipping the wheelchairs.	Shipping Agent, Broker, Courier	Freight Invoice	

Actor	Role	Description	Example	Synonyms	Sends	Receives
Party	Consignor	The consignor will receive an invoice and make payments for the transport service provided.	The consignor will pay the freight forwarder for shipping the wheelchair to the destination.	Despatch Point, Shipper, Sender		Freight I

**Table 1. Party Roles**

## 5.3 Sourcing

There are three kinds of sourcing:

- Catalogue provision
- Customer initiated sourcing
- Punchout

A Seller, Contractor Party, Originator Party, or Buyer Party can initiate sourcing.

### 5.3.1 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 be a Customer.

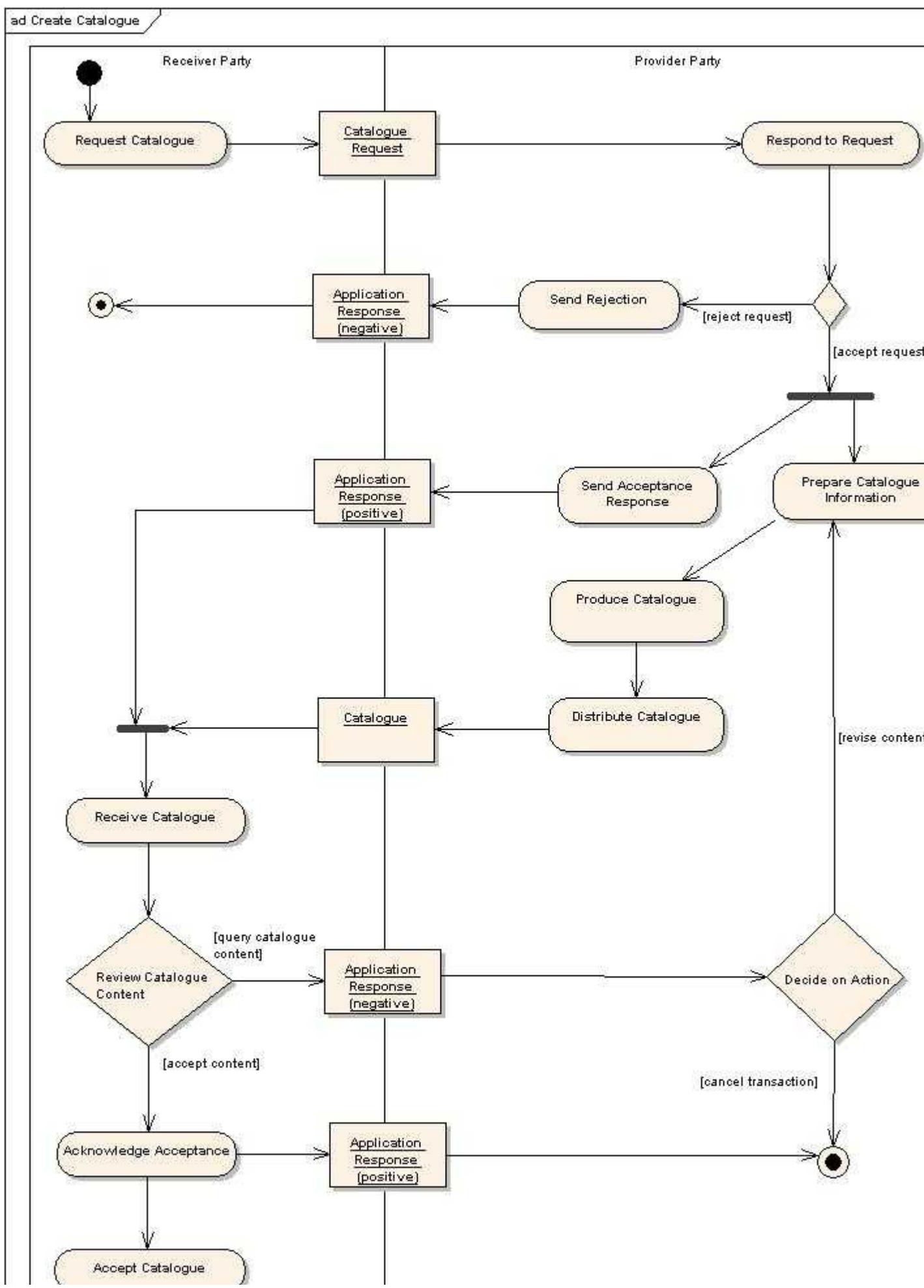
#### 5.3.1.1 Sourcing Business Rules

- Any conditions specified in the contract will overrule those stated in the common catalogue.
- A catalogue exchange is between one Provider and one Receiver Party.
- A classification system may have its own set of properties.
- A classification scheme will 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 can explicitly specify the framework contract reference.
- A catalogue can refer to a DPS contract number.
- When a catalogue item is updated, the item is 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 can 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 is at the discretion of the Provider Party.
- If an updated catalogue is issued, then an action code will define the status of the items in the catalogue.

### 5.3.1.2 Create Catalogue

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

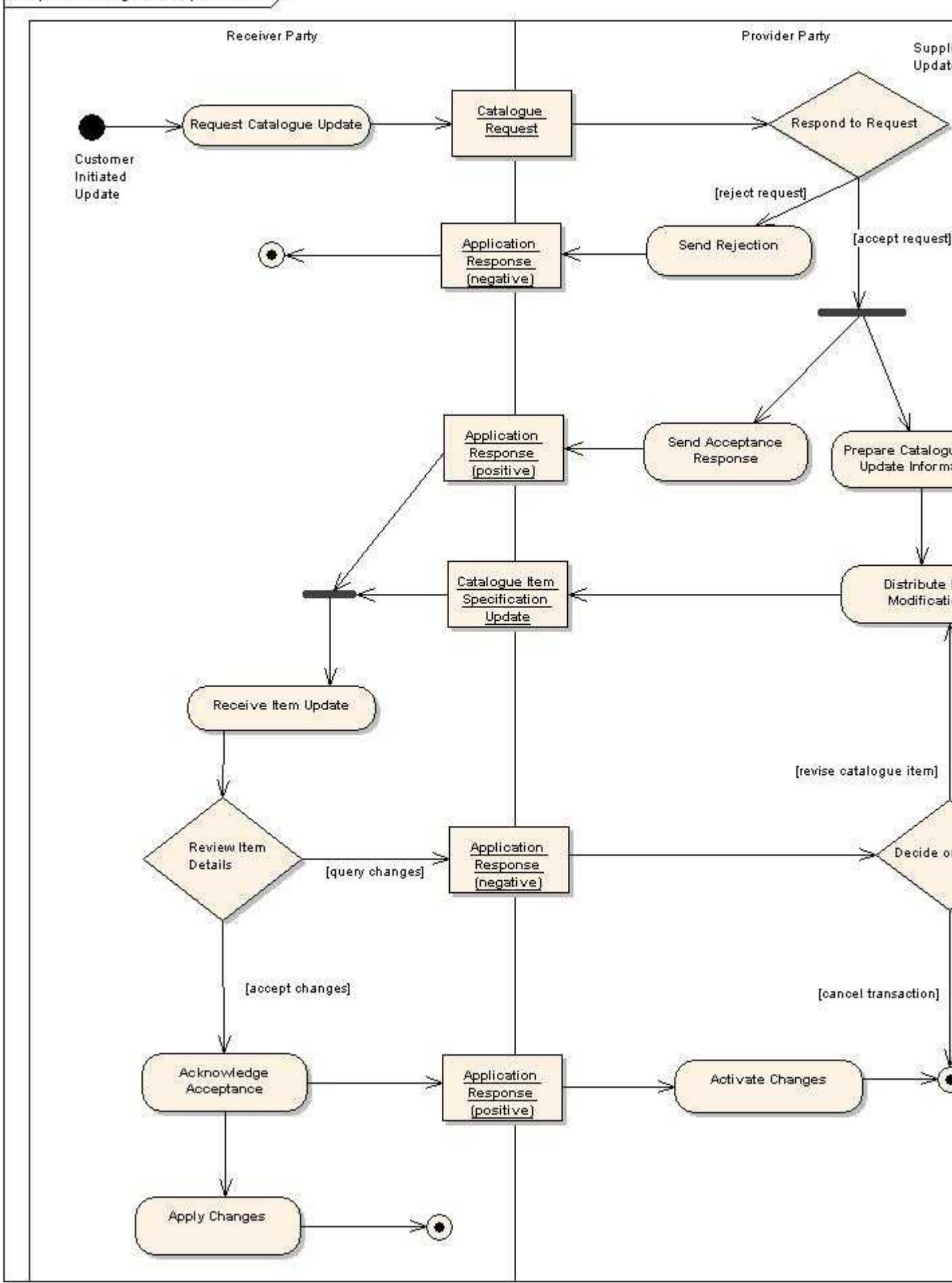


## **Figure 2. Create Catalogue Process Model**

### **5.3.1.3 Update Catalogue Item Specification**

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

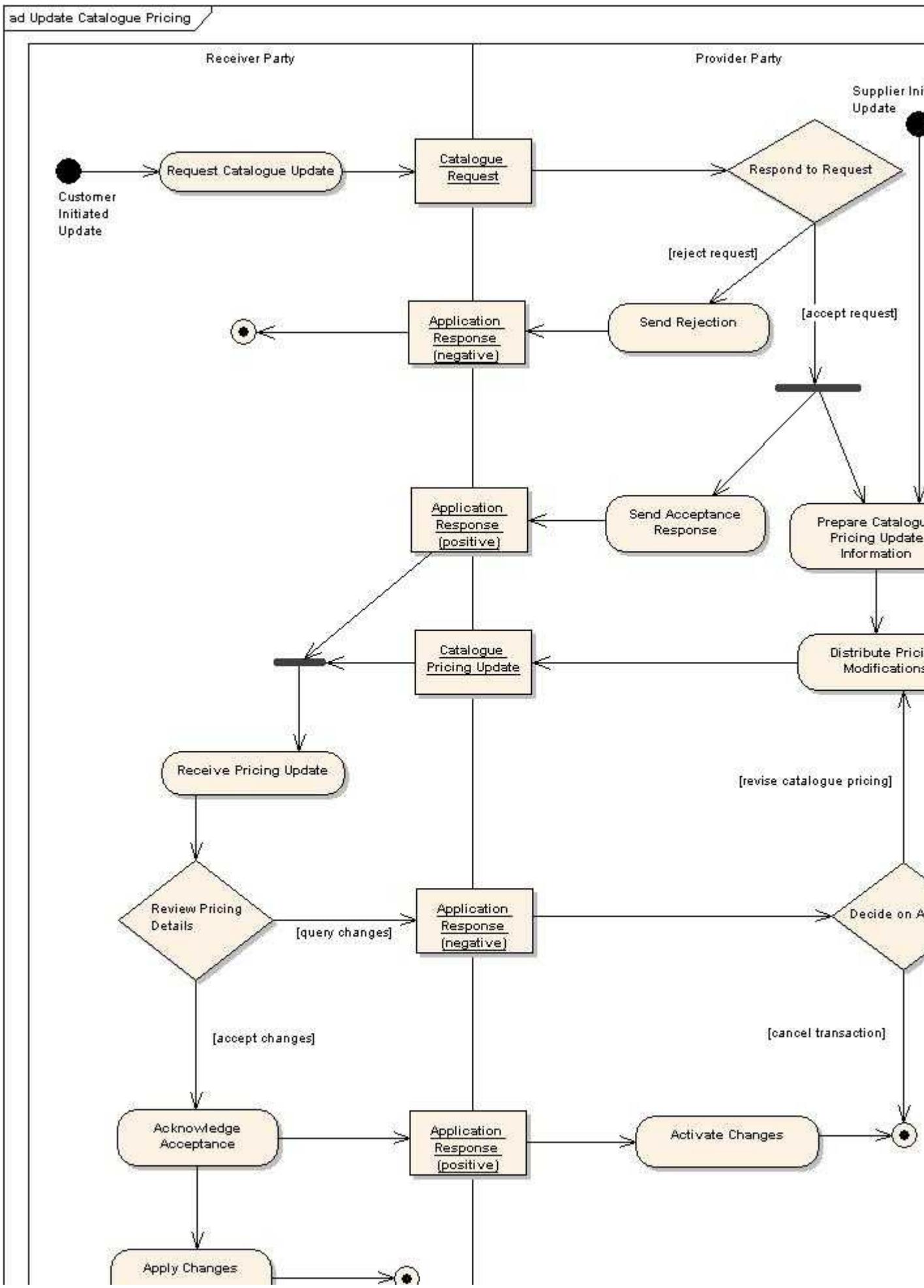
ad Update Catalogue Item Specification



### **Figure 3. Update Item Specification Process Model**

#### **5.3.1.4 Update Catalogue Pricing**

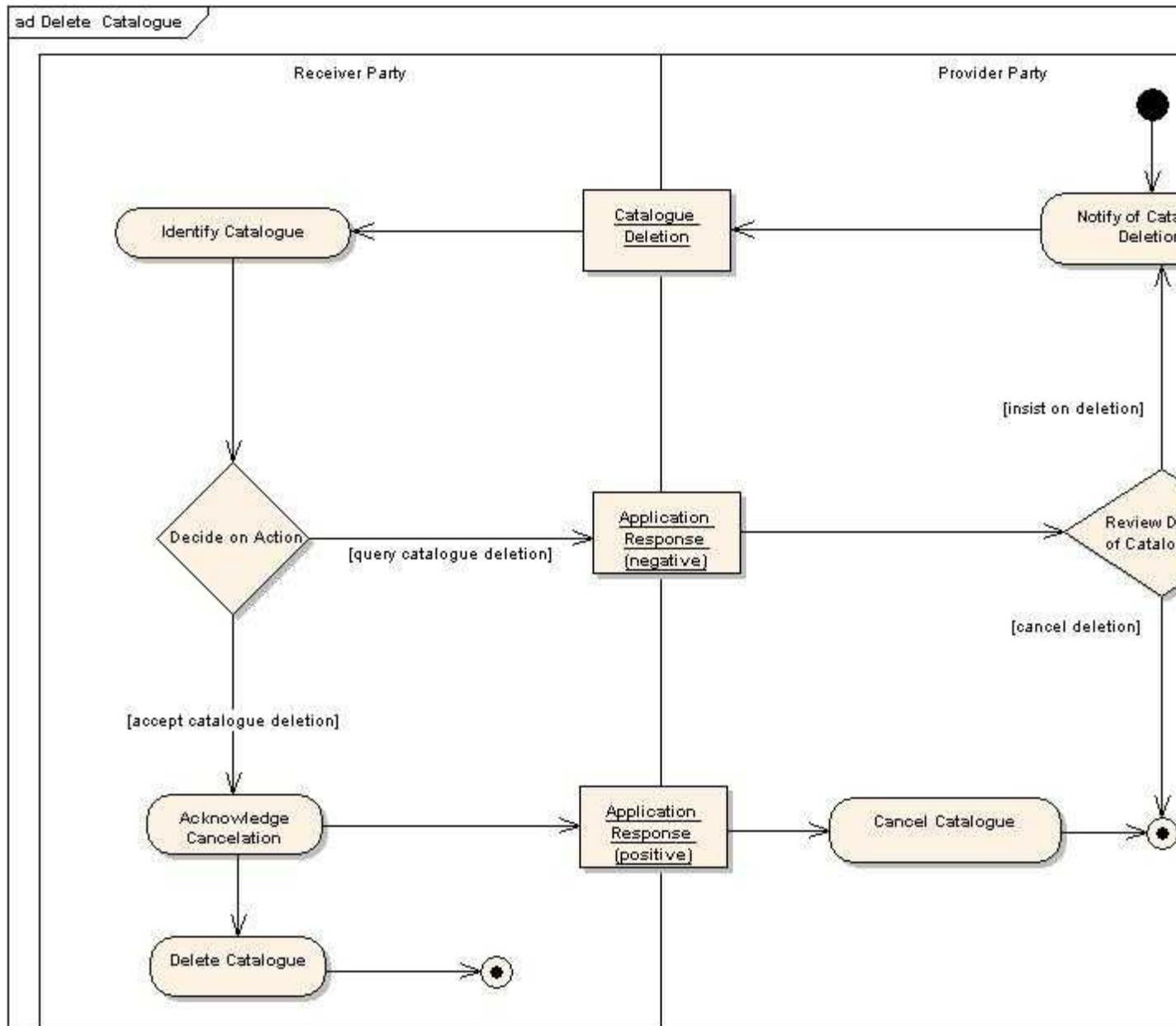
The process of updating Catalogue pricing is shown in the following activity diagram.



**Figure 4. Update Catalogue Pricing Process Model**

### 5.3.1.5 Delete Catalogue

Deletion of a Catalogue is shown in the following activity diagram.

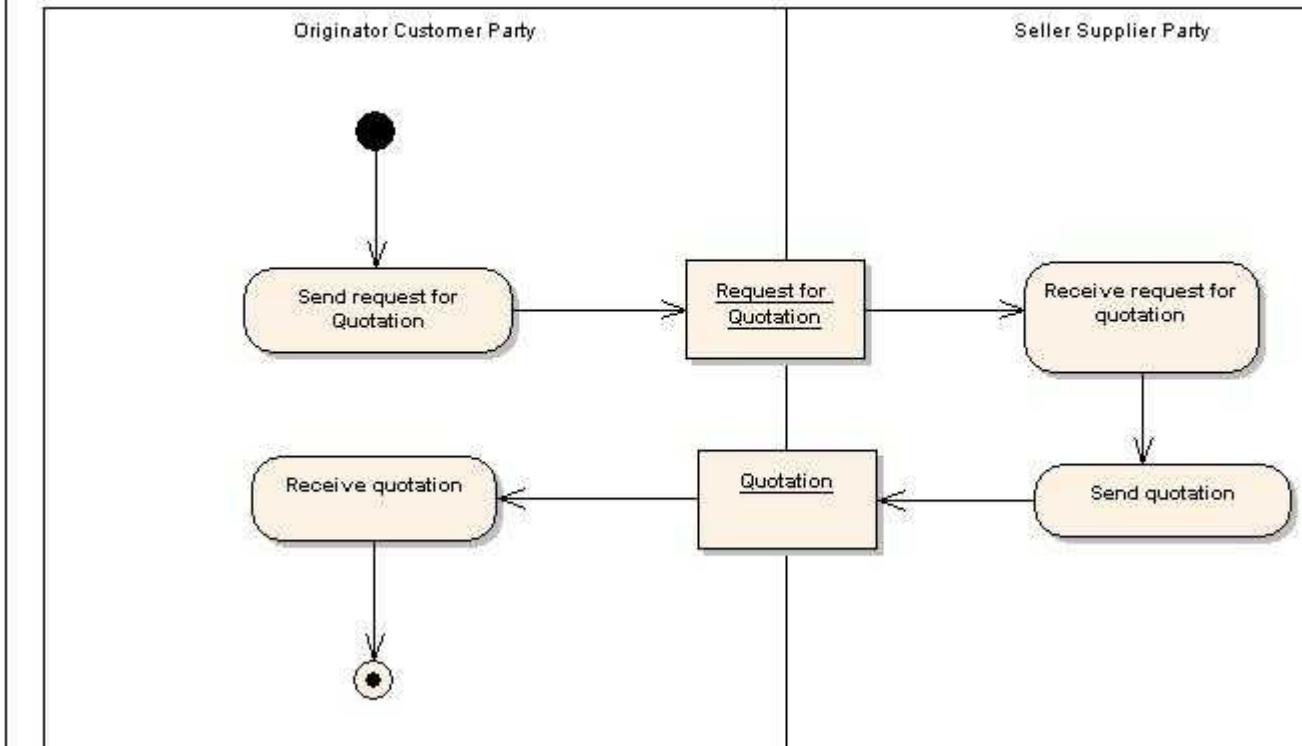


**Figure 5. Delete Catalogue Process Model**

### 5.3.2 Customer Initiated Sourcing

Customer initiated sourcing is the case where the Originator asks for a quotation, as shown in the following activity diagram.

#### cd Sourcing Buyer Initiated

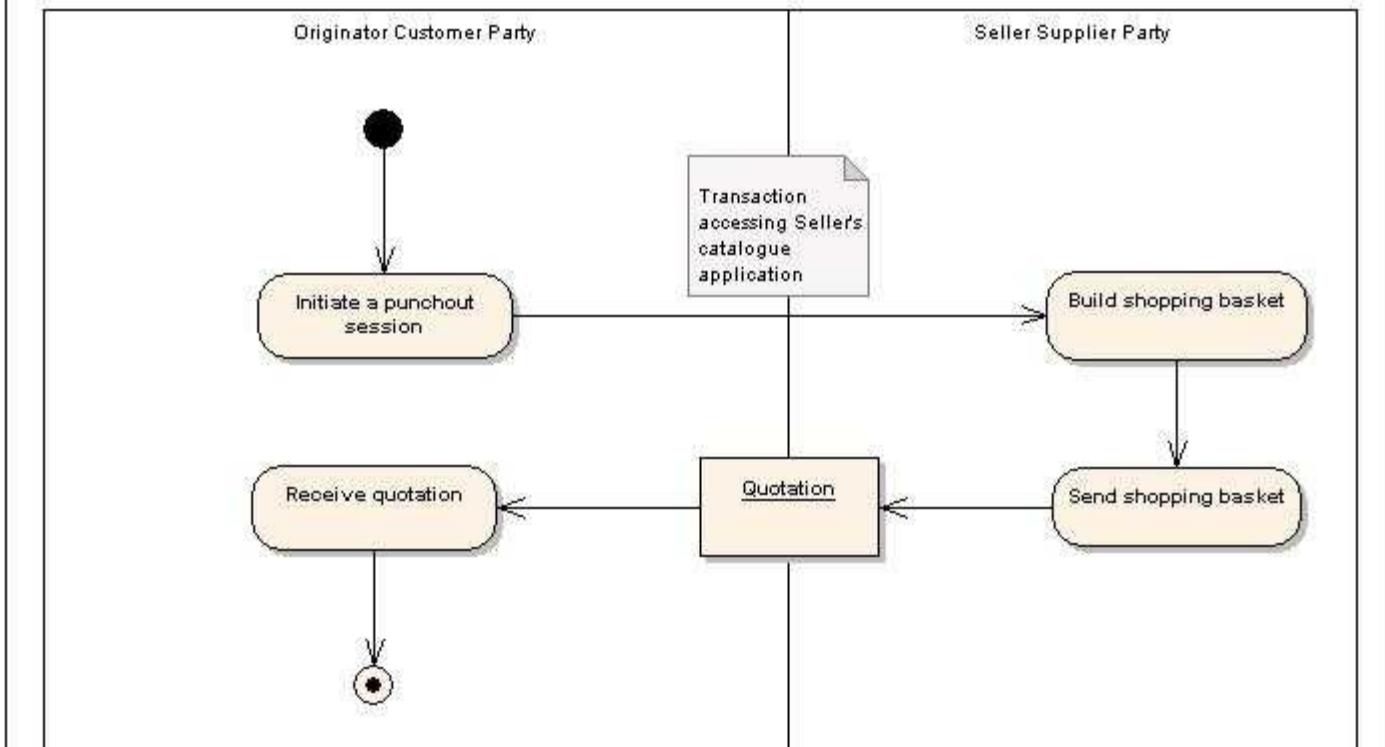


**Figure 6. Customer Initiated Sourcing Process Model**

### 5.3.3 Sourcing Punchout

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

#### cd Sourcing punch out



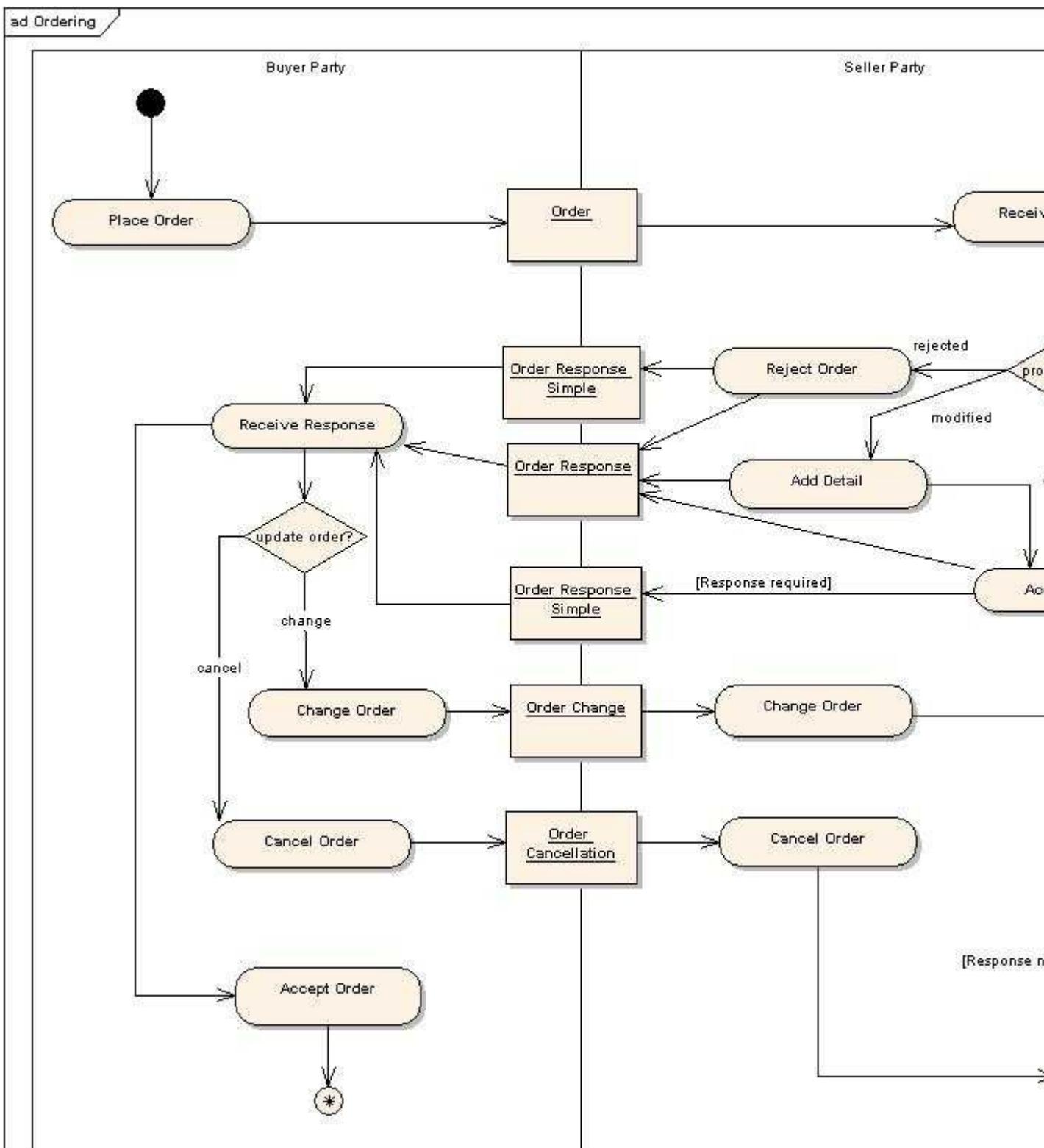
## **Figure 7. Punchout Sourcing Process Model**

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 considered outside the scope of UBL 2.0.

## **5.4 Ordering**

Ordering is the collaboration that creates a contractual obligation between the Seller and the Buyer.



**Figure 8. Ordering Process Model**

### 5.4.1 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 can be placed “on account” against a trading credit account held by the Seller, or against a credit/debit card account, or 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 can 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 (5.4.3).

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 alternatives that are acceptable.

## 5.4.2 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.

## 5.4.3 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 or substitutes which will be made, or changes necessary, using the Order Response.

## 5.4.4 Order Change

The Buyer can 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 5.4.5) followed by a new, complete replacement Order.

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

Buyers can initiate a change to a previously accepted order for various reasons, such as changing

ordered items, quantity, delivery date, ship-to address, etc. Suppliers can accept or reject the Order Change using either Order Response or Order Response Simple.

## 5.4.5 Order Cancellation

At any point of the process, a Buyer can cancel an established order transaction using the Order Cancellation document. Legal contracts, trading partner agreements, and business rules will restrict at what point 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.

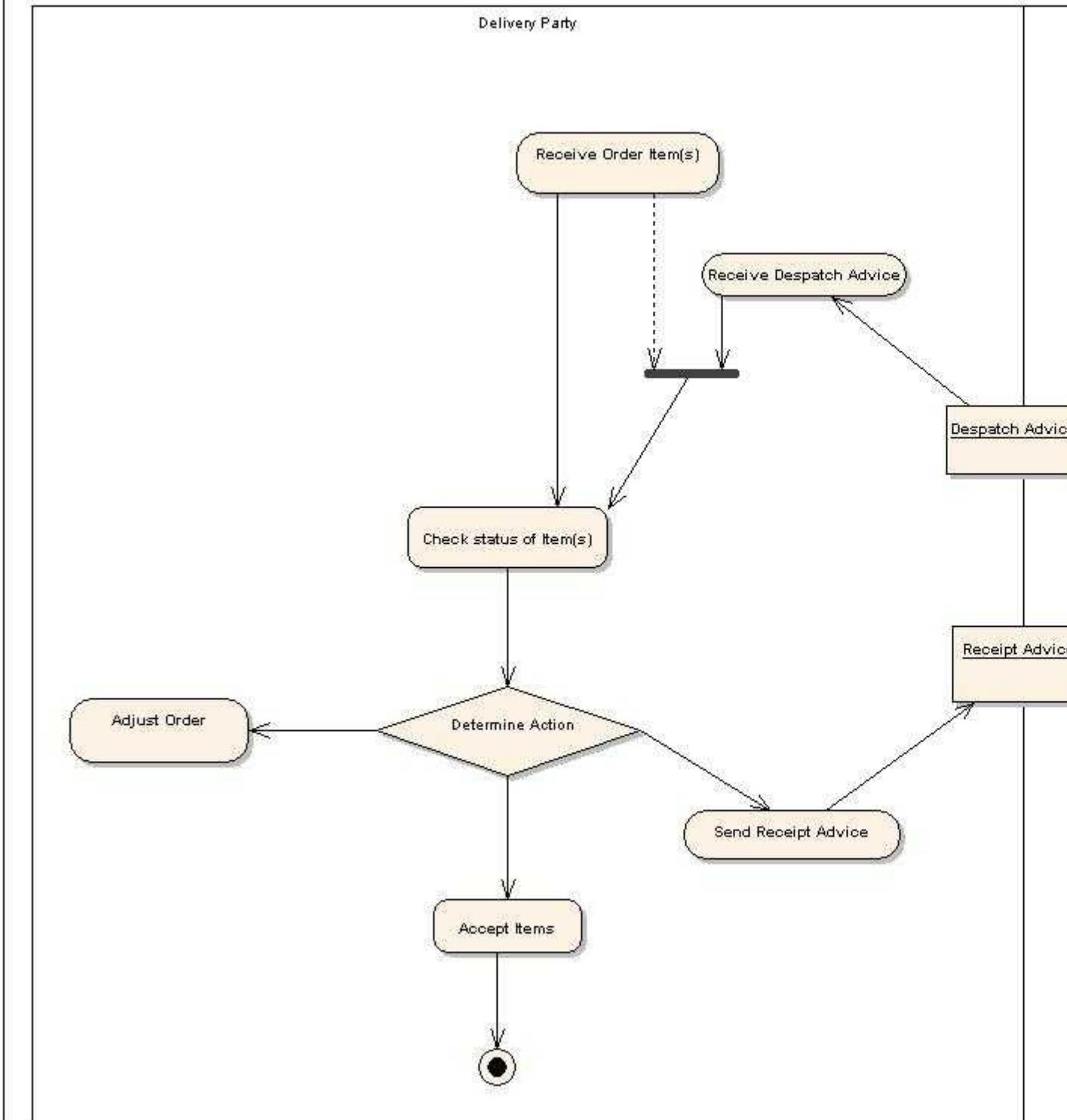
## 5.5 Fulfilment

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

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 5.4, Ordering).

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



**Figure 9. Fulfilment with Despatch Advice Process Model**

### 5.5.1 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:

- Organization of the delivery set of items by Transport Handling Unit(s) so that the Receiver can

check the Transport Handling Unit and then 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.

- 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 can advise:

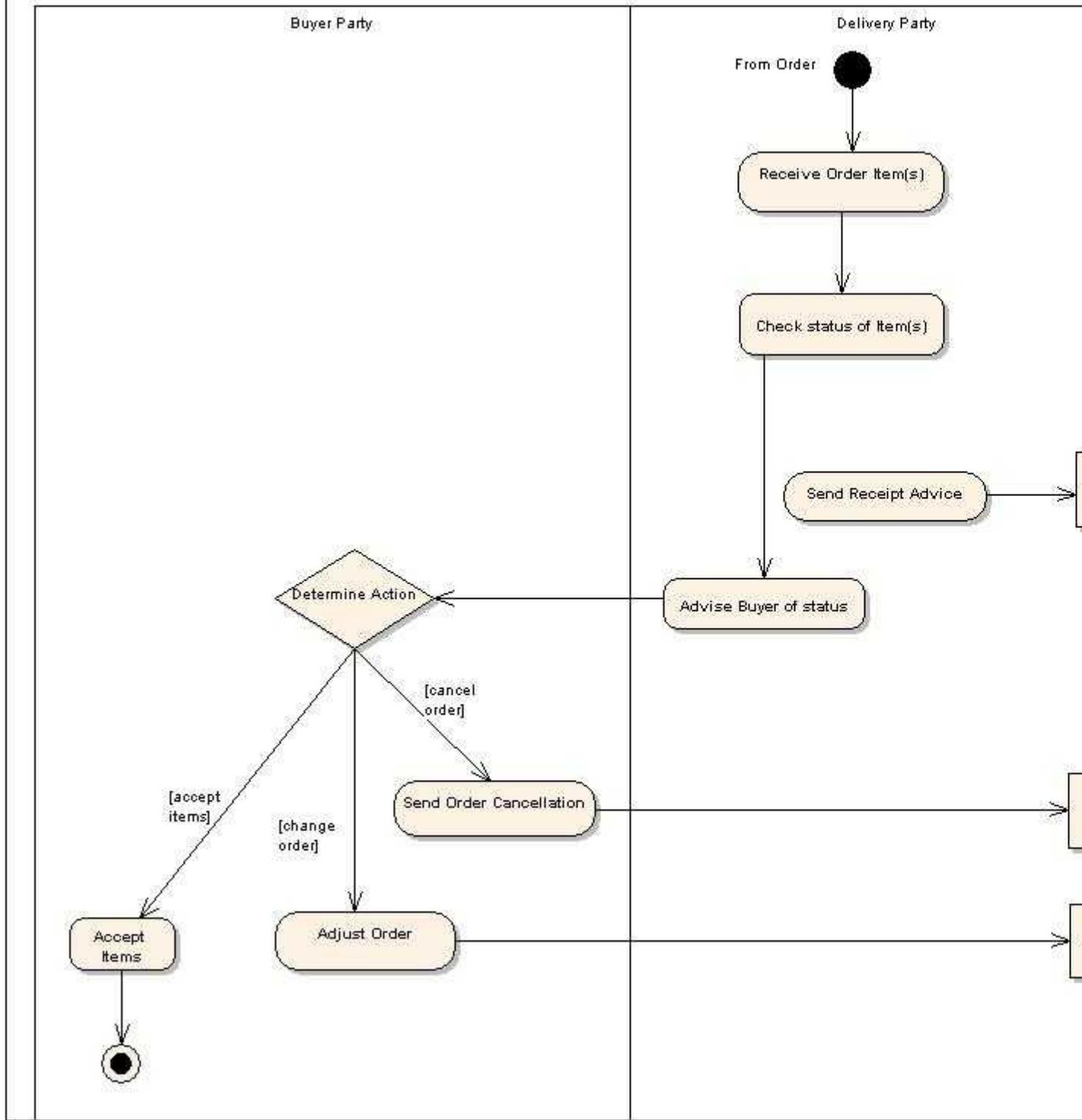
- Full Despatch — advising the Recipient and/or Buyer that all the items on the order will be, or are being, delivered in one complete consignment on a given date.
- Partial Despatch — advising the Recipient and/or Buyer that the items on the order will be, or are being, partially delivered in a consignment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

## 5.5.2 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 10. Fulfilment with Receipt Advice Process Model**

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:

- 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.
- 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.

## 5.6 Billing

In the Billing collaboration, 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 can be billed.

For UBL 2.0, we propose the following methods:

1. Traditional Billing

- Using Credit Note
- Using Debit Note

2. Self Billing (also known as billing on receipt)

- Using Credit Note
- Using Self billed Credit Note

### 5.6.1 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). The OASIS TaxML Technical Committee ([http://www.oasis-open.org/committees/tc\\_home.php?wg\\_abbrev=tax](http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=tax)) is developing UBL implementation profiles for various tax regimes, such as those required by the European Community.

Charges can 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.

## 5.6.2 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 can be created at the time of despatch or when the Delivery Party acknowledges that the goods have been received (using a Receipt Advice).

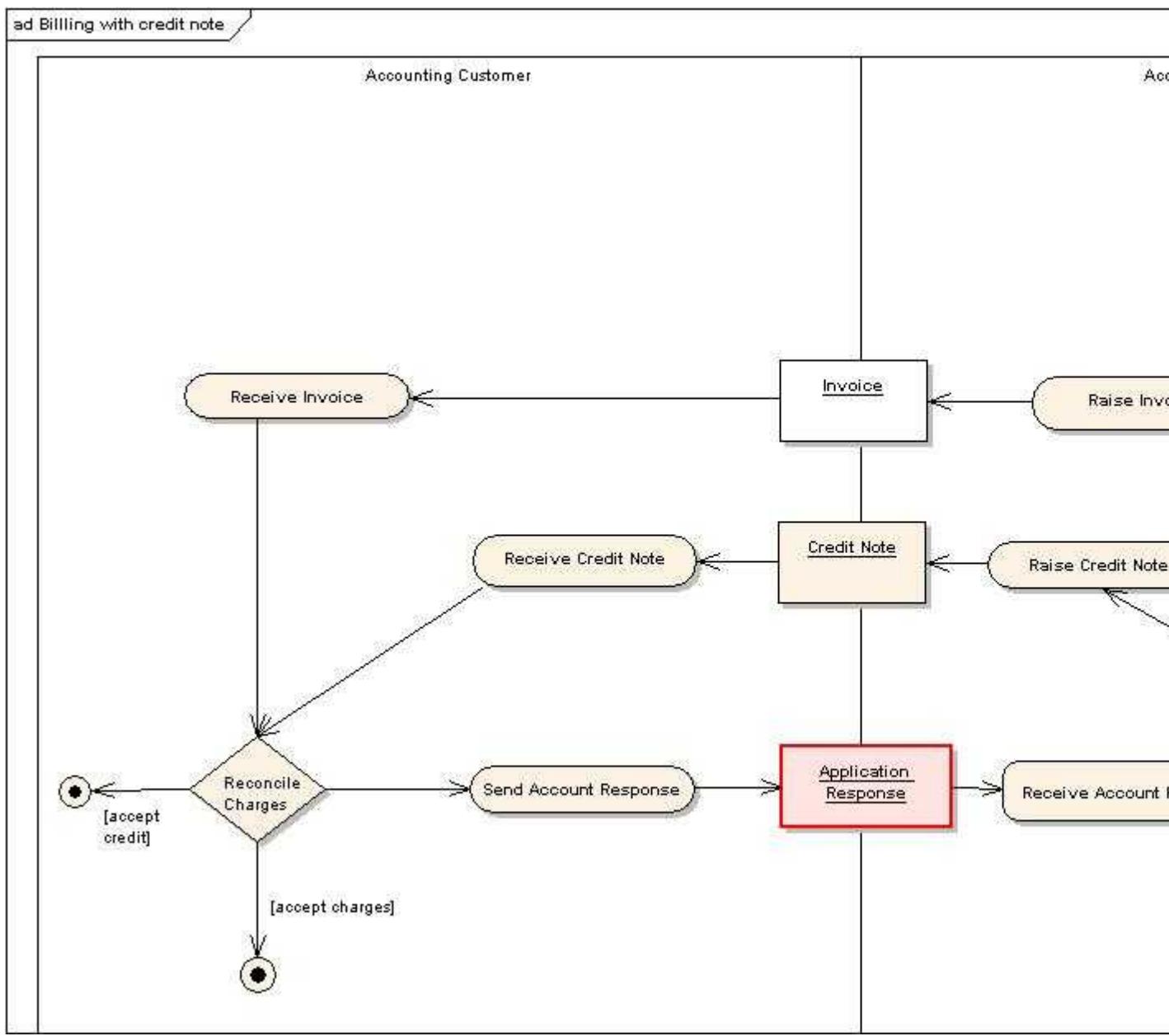
When there are discrepancies between the Despatch Advice, Receipt Advice, and/or the Invoice and the goods actually received, or the goods are rejected for quality reasons, the customer may send an Application Response or a Debit Note to the supplier. The supplier may then issue a Credit Note or another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

### 5.6.2.1 Billing using Credit Notes

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

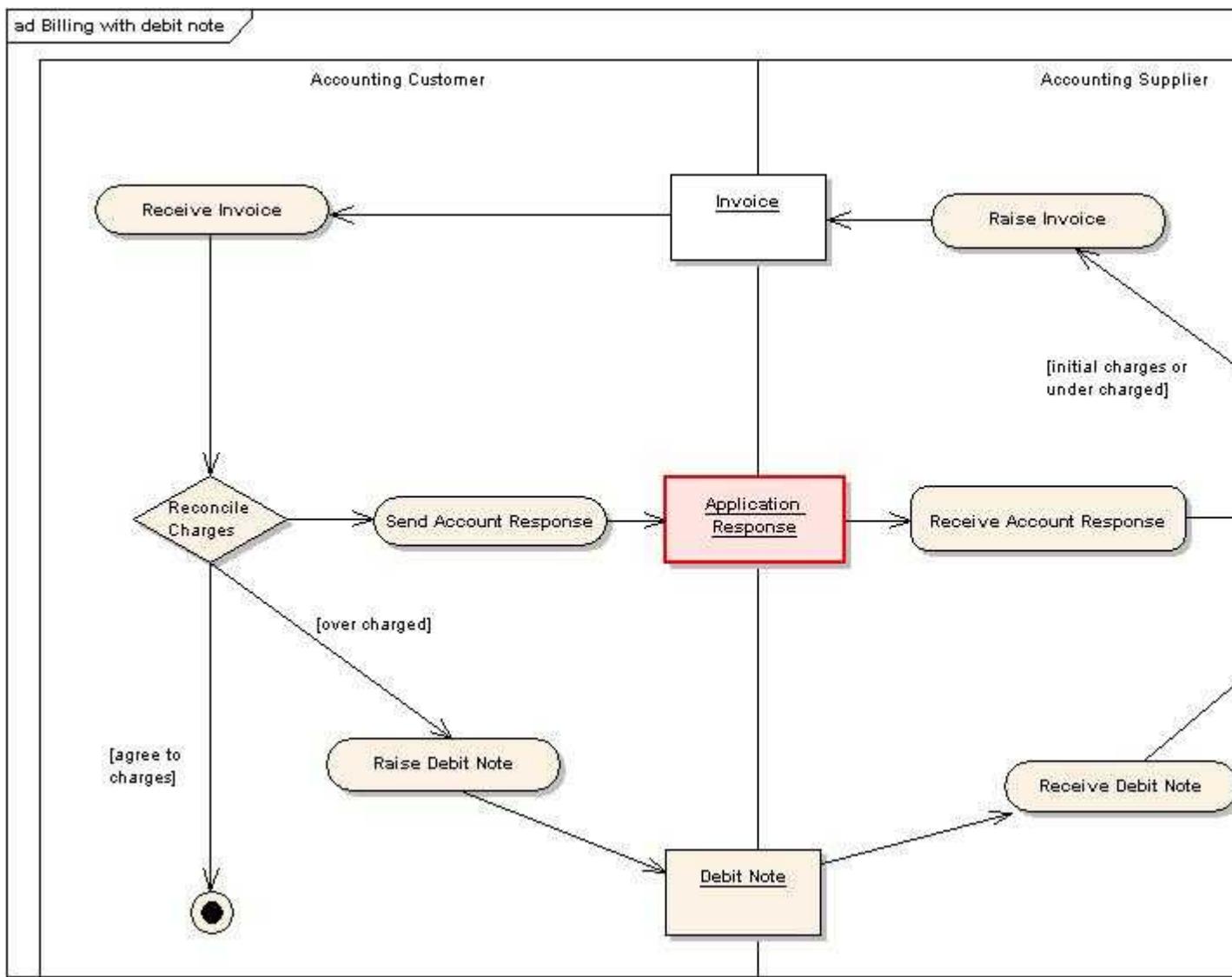


**Figure 11. Billing with Credit Note Process Model**

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

#### 5.6.2.2 Billing using Debit Notes

Billing using Debit Notes is shown in the following activity diagram.



**Figure 12. Billing with Debit Note Process Model**

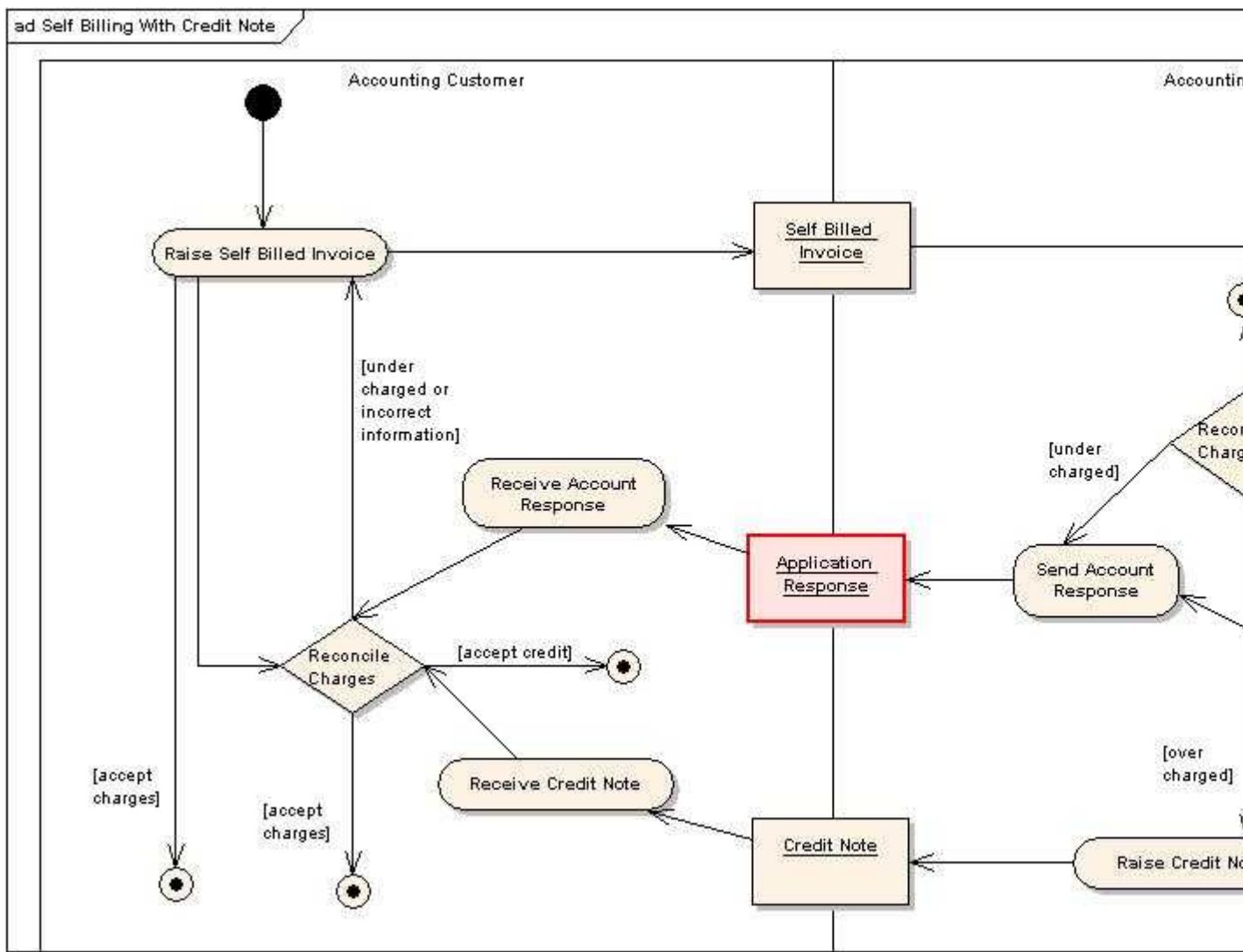
When using Debit Notes, both the supplier (in their Accounting role) and the customer (in their Accounting role) are responsible for providing taxation information.

### 5.6.3 Self Billing

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

#### 5.6.3.1 Self Billing using Credit Notes

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

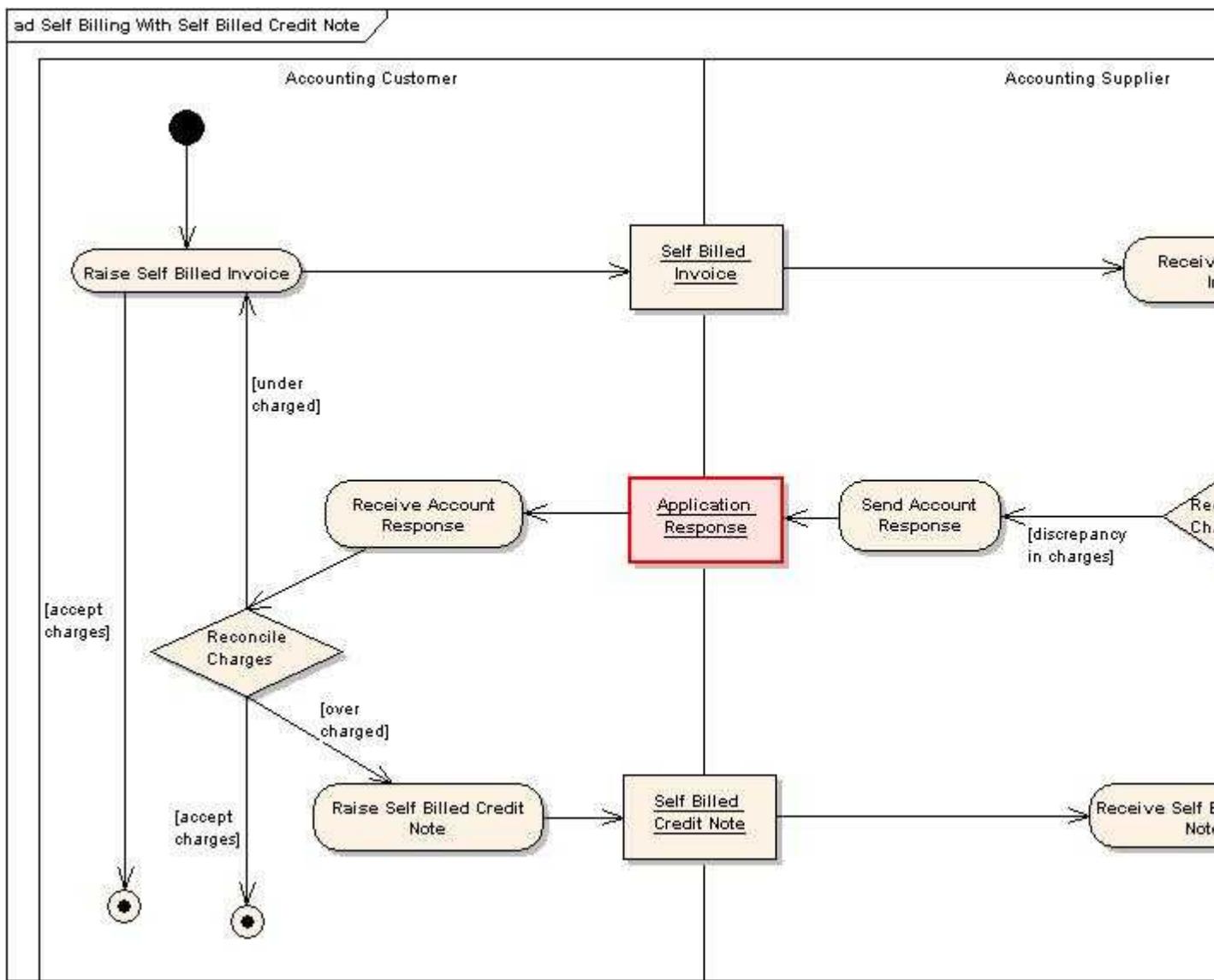


**Figure 13. Self Billing with Credit Note Process Model**

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, they 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 billing credit note.

#### 5.6.3.2 Self Billing using Self Billed Credit Notes

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



**Figure 14. Self Billing with Self Billed Credit Note Process Model**

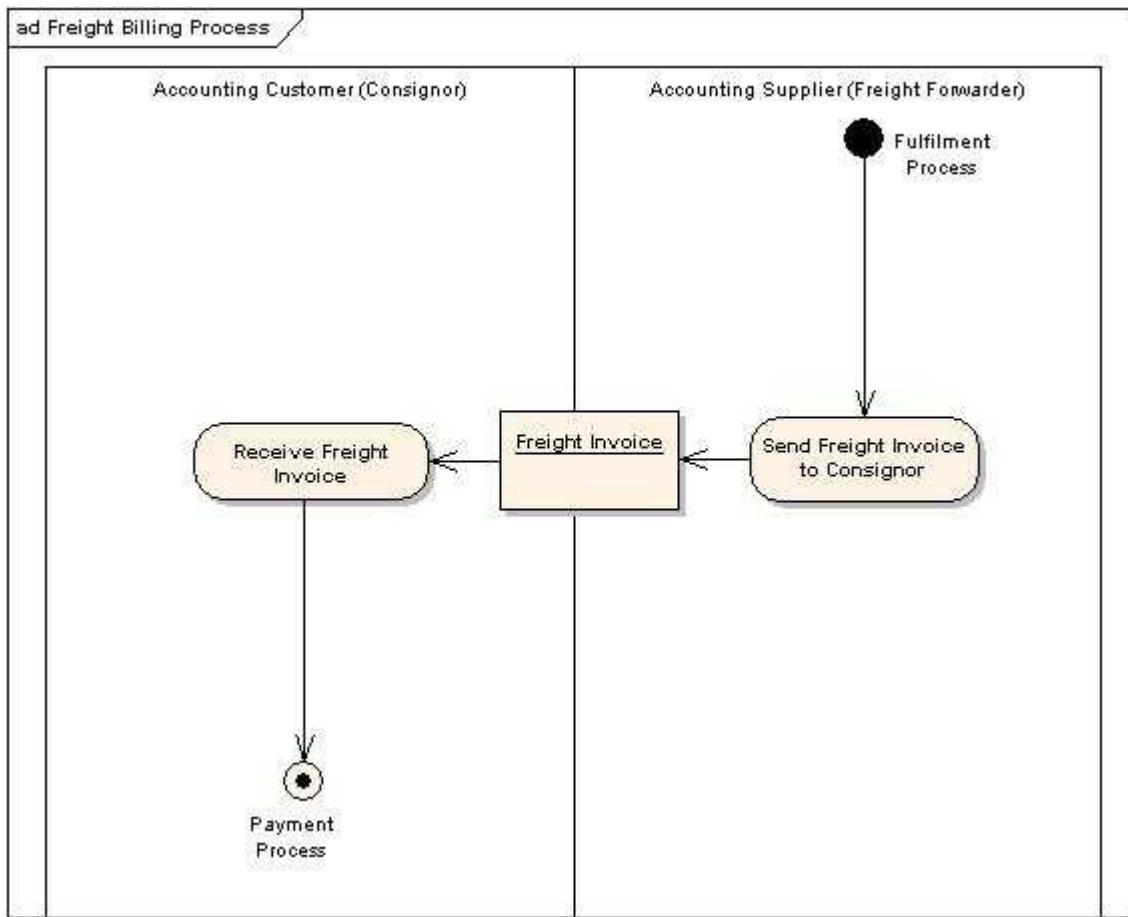
When using Self Billed Credit Notes, the customer is raising the Self Billed Credit Note *in the name and on behalf of* the supplier. Therefore the supplier and the customer are still both responsible for providing taxation information.

## 5.6.4 Freight Billing

An extension of the Billing process is that of Freight Billing. This use case represents the billing process between the Consignor and the Freight Forwarder through the use of an Invoice for freight charges.

The party acting the role of Freight Forwarder initiates the process of billing the Consignor for logistic services.

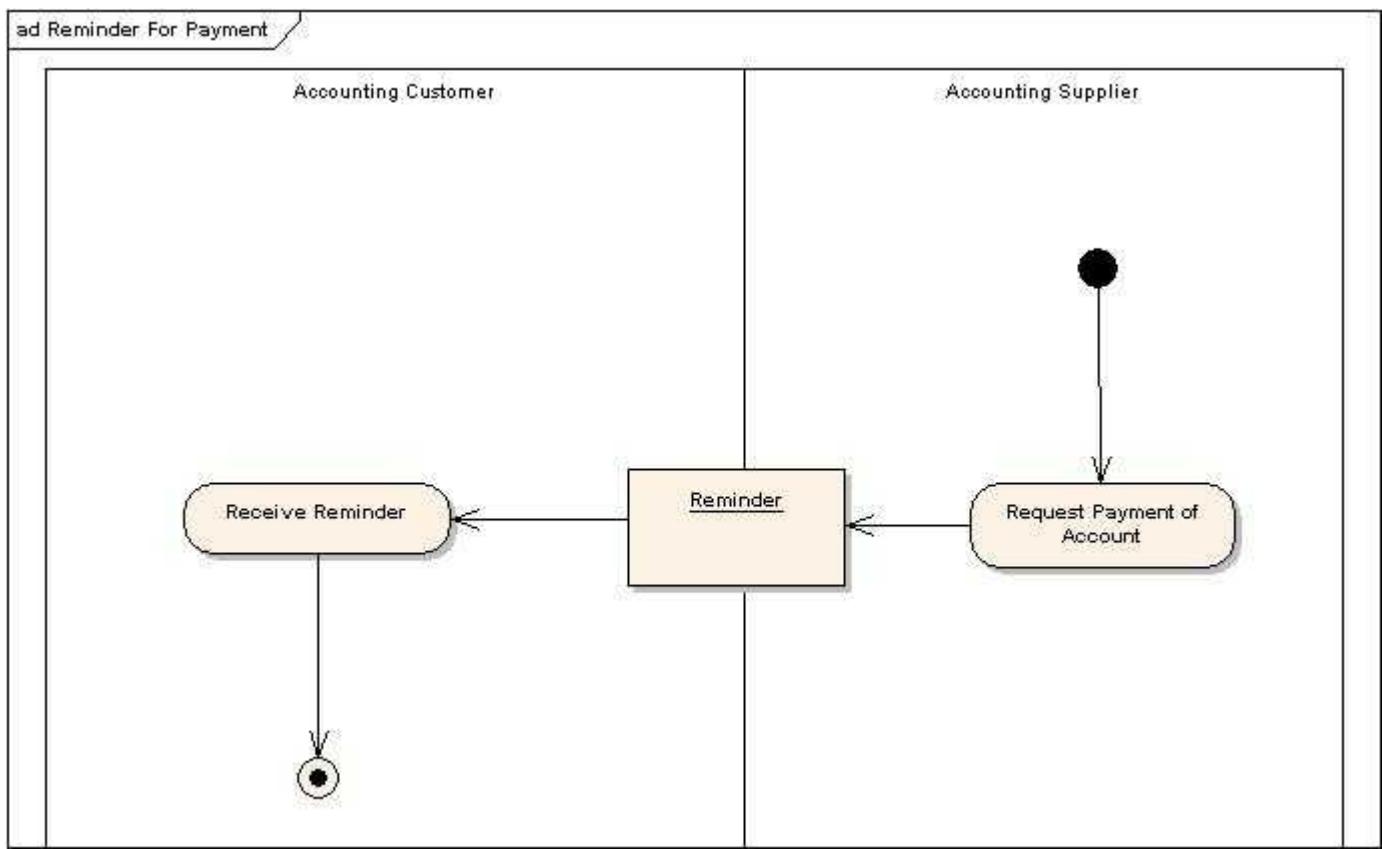
A Freight Invoice is issued by the party who provides the logistics services to the party who requests the service. The Freight Invoice lists the charges incurred in order to fulfill the agreed service.



**Figure 15. Freight Billing Process Model**

### 5.6.5 Reminder For Payment

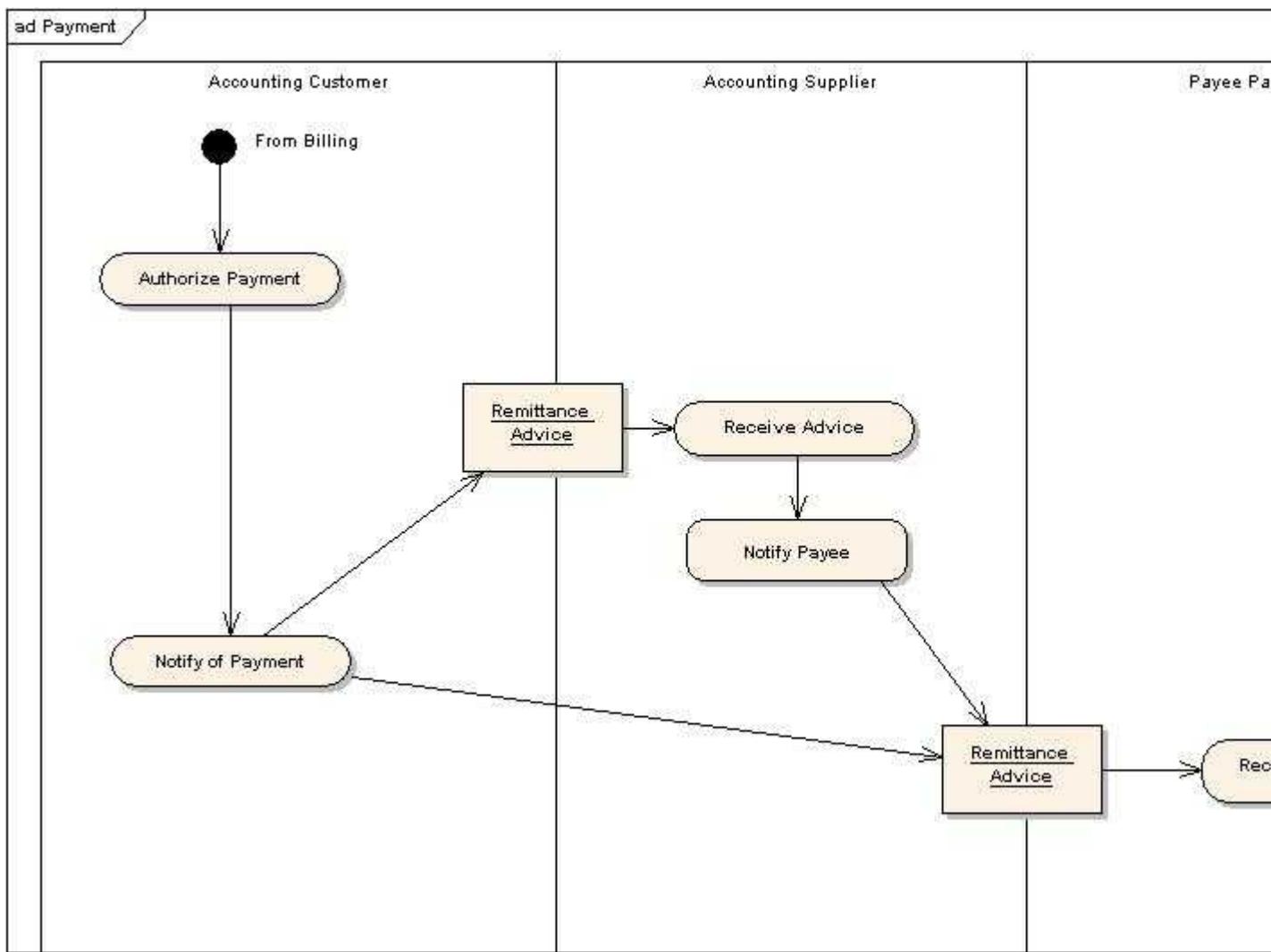
A Reminder can be used to notify the Customer of accounts due to be paid.



**Figure 16.** Reminder for Payment Process Model

## 5.7 Payment

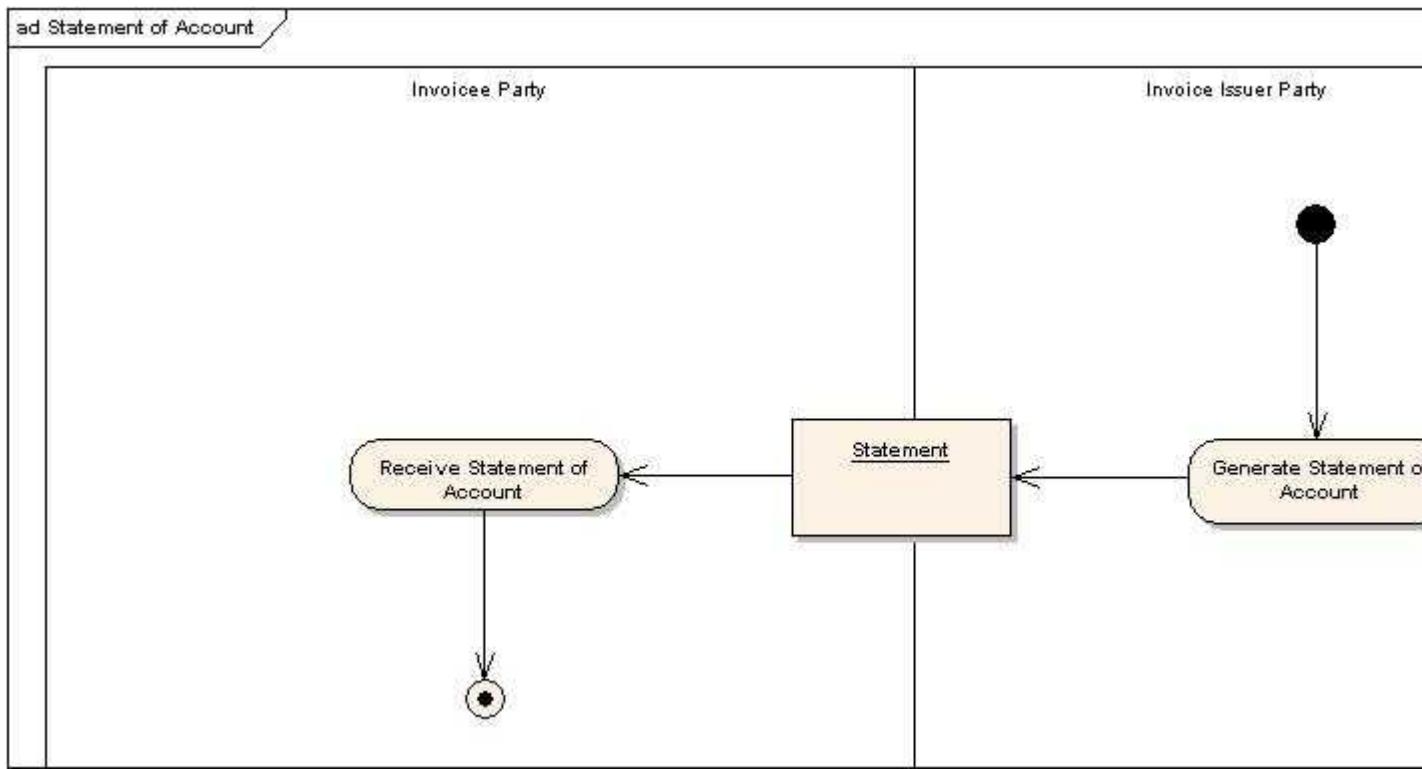
The payment collaboration is where the Payee (who is most often the Accounting Customer) is notified of any funds transferred against the account of the Accounting Supplier using a Remittance Advice.



**Figure 17. Payment Process Model**

### 5.7.1 Report State of Accounts

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

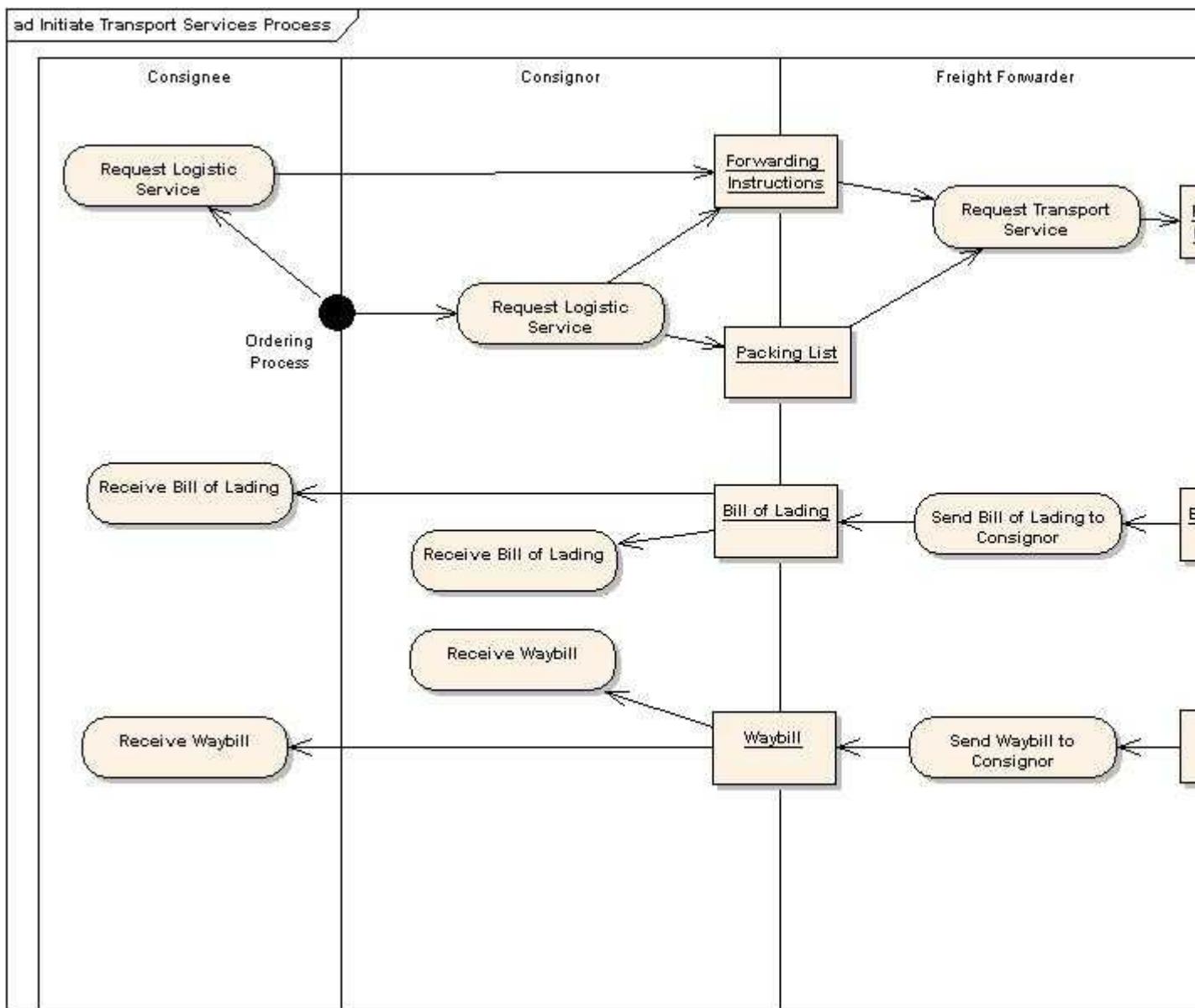


**Figure 18. Statement Process Model**

## 5.8 Initiate Transport Services

This use case defines the ordering of logistical services for international trade. With receipt of an order and acknowledgement by the Consignor that the goods are available and ready to be shipped, the Consignor initiates the transportation arrangements (although this may also be arranged by the Consignee). This includes booking the consignment with a Transportation Provider such as the Freight Forwarder or the Carrier and advising the Delivery Party of the arrangements as needed.

It should be noted that this use case does not cover regulatory notifications such as Customs declarations or arrangements for the physical movement of goods.



**Figure 19. Initiate Transport Services Process Model**

There are four types of documents involved in the process, namely:

- Forwarding Instruction
- Bill of Lading
- Waybill
- Packing List

### 5.8.1 Forwarding Instructions

Forwarding Instructions is normally 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. It can 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 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 “shipper”. This document can be used to arrange for the transportation:

- of different types of goods or cargoes
- whether containerized or non-containerized
- through different modes of transport, including multi-modal, and
- from any origin to any destination.

## 5.8.2 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.

It can 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 does not provide the physical transportation service.

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

## 5.8.3 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 shipper and consignee.

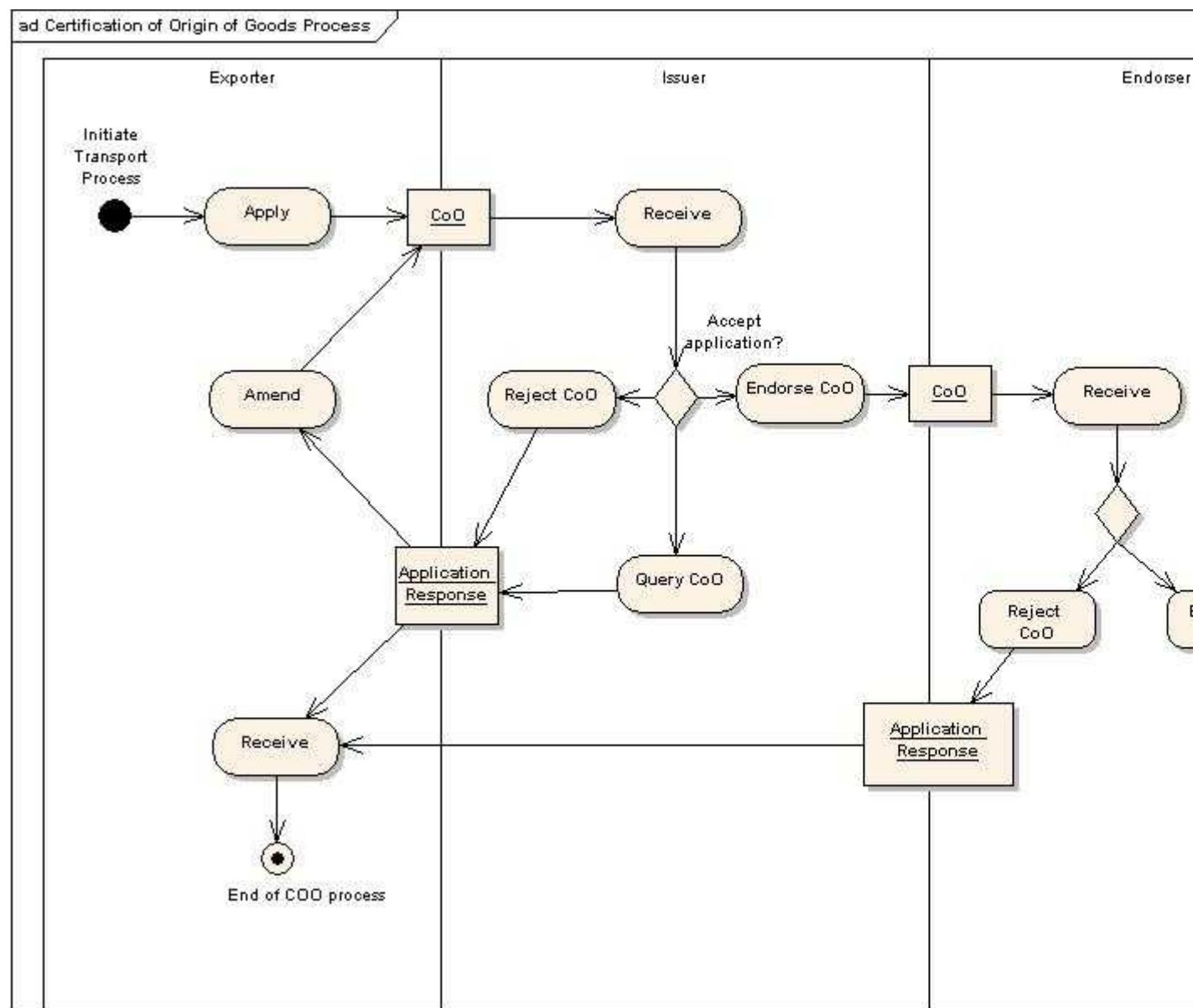
## 5.8.4 Packing List

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.

## **5.9 Certification of Origin of Goods**

A Certificate of Origin (CO) 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 apply it to a local chamber of commerce or any designated government agency/board, i.e., 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 that the Exporter claims the goods originated in that country. Finally, the issued Certificate of Origin is received by the Importer.



## **Figure 20. Certification Of Origin Of Goods Process Model**

The party acting the role of Exporter initiates the process of certification of origin of goods. It is the responsibility of the party acting the role of Exporter to initiate the process by submitting an application to an authorized Issuer.

## **5.10 Document Types**

The following table lists all the documents used in each collaboration.

Document	Description	Use case(s) involved
Catalogue Request	A document to request a Catalogue from a seller. May be either an entire new Catalogue or an update (at the discretion of the Seller).	Create Catalogue, Update Item Specification, Update Pricing
Catalogue	A document produced by a party in the procurement chain that describes items and prices. The document typically enables the transmission of information regarding pricing and catalogue details for goods and services offered by a seller to a buyer.	Create Catalogue
Catalogue Deletion	A document to cancel an entire Catalogue. All previous Catalogue information becomes obsolete.	Delete Catalogue
Catalogue Item Specification Update	A document to update information about Items in an existing Catalogue.	Update Catalogue Item Specification
Catalogue Pricing Update	A document to update information about Prices in an existing Catalogue.	Update Catalogue Pricing
Request For Quotation	A document to request pricing and availability information about goods or services. The document may requesting a quote on specified goods or services.	Sourcing
Quotation	A document to specify pricing and availability information about goods or services. The document which, with a view to concluding a contract, sets out the conditions under which the goods are offered.	Sourcing
Order	A document that contains information directly relating to the economic event of ordering products. The document by means of which a customer initiates a	Ordering

Document	Description	Use case(s) involved
	transaction with a supplier for the supply of goods or services as specified, according to conditions set out in an offer, or otherwise known to the customer.	
Order Response	A document responding to the customer to indicate detailed responses against a single order already received.	Ordering
Order Response Simple	A document responding to the customer to indicate simple acceptance or rejection of an entire order. The document acknowledging an undertaking to fulfil an order and confirming conditions or acceptance of conditions.	Ordering
Order Change	A document that contains information directly relating to the economic event of changing an order already sent.	Ordering, Fulfilment
Order Cancellation	A document that advises either party of the cancellation of an Order.	Ordering, Fulfilment
Despatch Advice	A document that describes the content of goods shipped. Document/message by means of which the seller or consignor informs the consignee about the despatch of goods.	Fulfilment
Receipt Advice	A document that advises the goods received and accepted by the buyer. The document acknowledges the receipt of goods and in addition may indicate receiving conditions.	Fulfilment
Invoice	A document claiming payment for goods or services supplied under conditions agreed between the supplier and the customer. In most cases this document describes the actual financial commitment of goods or services ordered from the supplier.	Billing
Self Billed Invoice	A document provided by a customer, in the name and on behalf of the supplier, describing the claim for payment for goods or services supplied under conditions agreed between the supplier and the customer.	Billing
Credit Note	A document for a supplier to specify a	Billing

Document	Description	Use case(s) involved
	reduced payment. The document for providing credit information to the relevant party.	
Debit Note	A document for a customer to specify a reduced payment. The document for providing debit information to the relevant party.	Billing
Self Billed Credit Note	A document for a customer to specify a reduced payment in a Self Billing environment. The document indicates that the customer is claiming credit in a self billing environment.	Billing
Statement	A document to list the financial transactions between customer and supplier and notify of their status.	Billing
Remittance Advice	A document to specify that funds have been transferred from the customer to the supplier. The document advising of the remittance of payment.	Payment
Forwarding Instructions	The document 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. The document issued to a freight forwarder, giving instructions regarding the action to be taken by the forwarder for the forwarding of goods described therein.	Initiate Transport Services
Bill of Lading	A document issued by the party who acts as an agent for the carrier or other agents,	Initiate Transport Services

Document	Description	Use case(s) involved
	<p>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 Instructions. 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. A negotiable document that evidences a contract of carriage by sea and the taking over or loading of goods by carrier, and by which carrier undertakes to deliver goods against surrender of the document. A provision in the document that goods are to be delivered to the order of a named person, or to order, or to bearer, constitutes such an undertaking.</p>	
Waybill	<p>A document 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 may not provide the physical transportation service. It corresponds to the information on the Forwarding Instructions. 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. A Waybill is a non-</p>	Initiate Transport Services

Document	Description	Use case(s) involved
	negotiable document evidencing the contract for the transport of cargo. It provides information similar to Bill of Lading but is not negotiable and cannot be assigned to a third party.	
Packing List	A document stating the detail of how goods are packed. The document specifies the distribution of goods in individual packages (in trade environment the despatch advice message is used for the packing list).	Initiate Transport Services
Freight Invoice	A document issued by a transport operation specifying freight costs and charges incurred for a transport operation and stating conditions of payment.	Freight Billing
Certificate of Origin	A document required by governments, declaring that goods in a particular international shipment are of a certain origin. Customs offices will use this document to determine whether or not a preferential duty rate applies on the products being imported and whether a shipment may be legally imported during a specific quota period. The document identifies which authority or body authorized to issue it certifies expressly that the goods to which the certificate relates originate in a specific country. The word "country" may include a group of countries, a region or a part of a country. This certificate may also include a declaration by the manufacturer, producer, supplier, exporter or other competent person.	Certification of Origin of Goods
Application Response	A document to indicate the application's response to a transaction at the business application level concerning the processing of a document.	All
Reminder	A document used to request payment.	Reminder for Payment
Transportation Status	A message to report the transport status and/or change in the transportation status (i.e. event) between agreed parties.	Report Status of Goods
Attached Document	In effect a 'wrapper' UBL document that can contain anything. This allows a	All

Document	Description	Use case(s) involved
	referenced document to be included in the package of documents being exchanged.	

**Table 2. Documents Used in Each Transaction**

## 6 UBL 2.0 Schemas

The UBL 2.0 XSD schemas are the only normative representations of the UBL 2.0 document types and library components.

All of the UBL 2.0 XSD schemas are contained in the `xsd/` subdirectory of the UBL 2.0 release package (see Appendix A for more information regarding the structure of the 2.0 release package and Section 6.4 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) `201C;runtime201D;` set with the annotation elements stripped out is provided in the `xsdrt/` directory.

### 6.1 UBL 2.0 Document Schemas

XSD schemas defining the 31 UBL 2.0 document types are located in the `xsd/maindoc/` directory, as listed below.

**ApplicationResponse**

<xsd/maindoc/UBL-ApplicationResponse-2.0.xsd>

**AttachedDocument**

<xsd/maindoc/UBL-AttachedDocument-2.0.xsd>

**BillOfLading**

<xsd/maindoc/UBL-BillOfLading-2.0.xsd>

**Catalogue**

<xsd/maindoc/UBL-Catalogue-2.0.xsd>

**CatalogueDeletion**

<xsd/maindoc/UBL-CatalogueDeletion-2.0.xsd>

**CatalogueItemSpecificationUpdate**

<xsd/maindoc/UBL-CatalogueItemSpecificationUpdate-2.0.xsd>

**CataloguePricingUpdate**

<xsd/maindoc/UBL-CataloguePricingUpdate-2.0.xsd>

**CatalogueRequest**

<xsd/maindoc/UBL-CatalogueRequest-2.0.xsd>

**CertificateOfOrigin**

<xsd/maindoc/UBL-CertificateOfOrigin-2.0.xsd>

**CreditNote**

[xsd/main/doc/UBL-CreditNote-2.0.xsd](#)

**DebitNote**

[xsd/main/doc/UBL-DebitNote-2.0.xsd](#)

**DespatchAdvice**

[xsd/main/doc/UBL-DespatchAdvice-2.0.xsd](#)

**ForwardingInstructions**

[xsd/main/doc/UBL-ForwardingInstruction-2.0.xsd](#)

**FreightInvoice**

[xsd/main/doc/UBL-FreightInvoice-2.0.xsd](#)

**Invoice**

[xsd/main/doc/UBL-Invoice-2.0.xsd](#)

**Order**

[xsd/main/doc/UBL-Order-2.0.xsd](#)

**OrderCancellation**

[xsd/main/doc/UBL-OrderCancellation-2.0.xsd](#)

**OrderChange**

[xsd/main/doc/UBL-OrderChange-2.0.xsd](#)

**OrderResponse**

[xsd/main/doc/UBL-OrderResponse-2.0.xsd](#)

**OrderResponseSimple**

[xsd/main/doc/UBL-OrderResponseSimple-2.0.xsd](#)

**PackingList**

[xsd/main/doc/UBL-PackingList-2.0.xsd](#)

**Quotation**

[xsd/main/doc/UBL-Quotation-2.0.xsd](#)

**ReceiptAdvice**

[xsd/main/doc/UBL-ReceiptAdvice-2.0.xsd](#)

**Reminder**

[xsd/main/doc/UBL-Reminder-2.0.xsd](#)

**RemittanceAdvice**

[xsd/main/doc/UBL-RemittanceAdvice-2.0.xsd](#)

**RequestForQuotation**

[xsd/main/doc/UBL-RequestForQuotation-2.0.xsd](#)

**SelfBilledCreditNote**

[xsd/main/doc/UBL-SelfBilledCreditNote-2.0.xsd](#)

**SelfBilledInvoice**

[xsd/main/doc/UBL-SelfBilledInvoice-2.0.xsd](#)

**Statement**

[xsd/main/doc/UBL-Statement-2.0.xsd](#)

**TransportationStatus**

[xsd/main/doc/UBL-TransportationStatus-2.0.xsd](#)

**Waybill**

[xsd/main/doc/UBL-Waybill-2.0.xsd](#)

## 6.2 UBL Common Schemas

The `xsd/common` directory contains schemas referenced by the document schemas in `xsd/maindoc`. The name of each schema file together with a brief description of its contents is given below.

### 6.2.1 Reusable BIE Schemas

#### **CommonBasicComponents**

[xsd/common/UBL-CommonBasicComponents-2.0.xsd](#)

This schema defines the global Basic Business Information Entities (BBIEs) that are used throughout UBL, serving, in effect, as a 201C;global BBIE type database201D; for constructing documents. BBIEs are the 201C;leaf nodes201D; of UBL documents.

#### **CommonAggregateComponents**

[xsd/common/UBL-CommonAggregateComponents-2.0.xsd](#)

This schema defines the Aggregate Business Information Entities (ABIEs) that are used throughout UBL, serving, in effect, as an 201C;ABIE type database201D; for constructing the main documents.

### 6.2.2 Reusable Datatype Schemas

#### **CCTS\_CCT\_SchemaModule**

[xsd/common/CCTS\\_CCT\\_SchemaModule-2.0.xsd](#)

This schema provides Core Component Types as defined by [\[CCTS\]](#). These types are used to construct higher-level datatypes in a standardized and consistent manner. This schema is defined by UN/CEFACT and should not be modified.

#### **UnqualifiedDataTypeSchemaModule**

[xsd/common/UnqualifiedDataTypeSchemaModule-2.0.xsd](#)

This schema defines Unqualified Data Types for primary and secondary representation terms as specified by [\[CCTS\]](#). Derived from Core Component Types, these XSD complexType structures are the basic data types from which all other data types must derive. This schema is defined by UN/CEFACT and should not be modified.

#### **QualifiedDatatypes**

[xsd/common/UBL-QualifiedDatatypes-2.0.xsd](#)

This schema provides Qualified Data Types as defined by [\[CCTS\]](#). These XSD complexType structures are derived from Unqualified Data Types by extension, restriction, and other contextual constraints, such as facets. The Qualified Datatypes have been customized for UBL and may be further extended to support additional datatypes required for other business contexts.

### 6.2.3 Documentation Metadata Schema

#### **CoreComponentParameters**

[xsd/common/UBL-CoreComponentParameters-2.0.xsd](#)

This 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 of customized extensions.

## 6.2.4 Imported Code List Schemas

Four standard code list schemas imported for use in UBL 2.0 are included in the `xsd/common` directory. These are defined by UN/CEFACT and should not be modified.

### **CodeList\_CurrencyCode**

[xsd/common/CodeList\\_CurrencyCode\\_ISO\\_7\\_04.xsd](#)

### **CodeList\_MIMEMediaTypeCode**

[xsd/common/CodeList\\_MIMEMediaTypeCode\\_IANA\\_7\\_04.xsd](#)

### **CodeList\_UnitCode**

[xsd/common/CodeList\\_UnitCode\\_UNECE\\_7\\_04.xsd](#)

### **CodeList\_LanguageCode**

[xsd/common/CodeList\\_LanguageCode\\_ISO\\_7\\_04.xsd](#)

This code list is not currently used in any UBL 2.0 documents. It is provided here to support customized implementation of textual content in different languages. For example, where a `TextType` component allows multiple occurrences, each different occurrence may be expressed in a different language. The actual language used can be identified using this code list.

Appendix D contains a description of the structure and form of representation used for UBL code lists.

## 6.2.5 Extension Content Schemas

See Section B.3.5 for information regarding UBL extension.

### **CommonExtensionComponents**

[xsd/common/UBL-CommonExtensionComponents-2.0.xsd](#)

This 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-2.0.xsd](#)

This schema specifies the actual structure of the `extension` element containing the foreign non-UBL content. This is delivered as an unconstrained element and may be replaced by users to specify the validation of their foreign vocabulary in a customized UBL document.

## 6.3 Schema Dependencies

The following diagram shows the dependencies among the schema modules comprising a UBL 2.0 document schema.

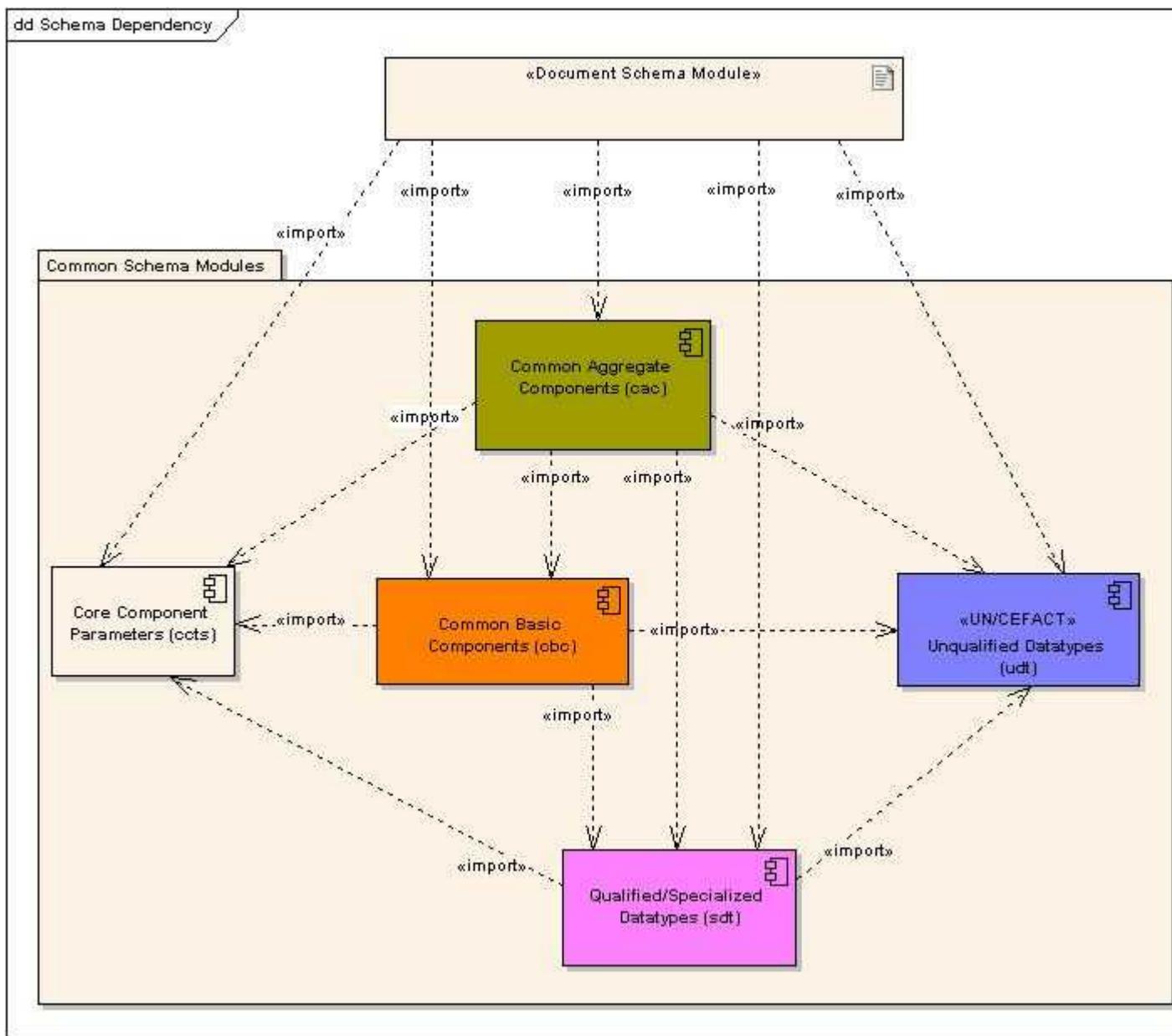


Figure 21. UBL Schema Dependencies

## Appendix A (Informative): Release Notes

### A.1 Availability

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

### A.2 Package Structure

This Public Review Draft of the UBL 2.0 specification is published as a zip archive named prd2-UBL-2.0.zip. Unzipping this archive creates a directory named prd2-UBL-2.0 containing a master hypertext

document (this document, UBL-index-2.0.html) and a number of subdirectories. The files in these subdirectories, linked to from UBL-index-2.0.html, contain the various normative and informational pieces of the 2.0 release. A description of each subdirectory is given below.

**art/**  
Diagrams and illustrations used in this specification

**asn/**  
ASN.1 UBL 2.0 schema; see Appendix F

**c1/**  
Code list specification files; see Appendix D

**doc/**  
Documents included with this release

**mod/**  
Spreadsheet data models; see Appendix C

**uml/**  
UML class diagrams of the UBL 2.0 data models; see Appendix C

**val/**  
Test harness for demonstrating UBL 2.0 two-phase validation; see Appendix D

**xml/**  
Sample UBL 2.0 instances

**xsd/**  
XSD schemas; see Section 6

**xsdrt/**  
201C;Runtime201D; XSD schemas; see Section 6

This draft package also contains a PDF file, UBL-index-2.0.pdf, that is automatically generated from UBL-index-2.0.html. The UBL-index-2.0.pdf file is included to comply with a procedural requirement of the current OASIS Technical Committee process and has no other function. It has no practical purpose and should be ignored. Please do not submit comments relating to the formatting or any other aspect of the UBL-index-2.0.pdf file.

## A.3 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>

## A.4 Known Issues

As the purpose of this release is to gain early input from interested parties, the focus has been on

exposing the UBL 2.0 data models and derived schemas rather than achieving perfection in editorial aspects of the specification. The present draft therefore exhibits the imperfections that would be expected at this stage of development. A list of known issues can be found at

<http://www.eccnet.com/UniversalBusinessLanguage/IssuesList.xml>

See Appendices C and D below for known issues regarding code lists and the UBL Naming and Design rules.

A technical review of the current schemas against the UBL NDRs will take place as part of the initial review cycle. Since not all recent NDR changes have been incorporated into EDIFIX programming, it is expected that a few inconsistencies will be found during the review.

## A.5 Supporting Materials

As an aid to deployment, the standard XML schemas in UBL 1.0 were accompanied by a large quantity of supporting materials, most of them included in the UBL 1.0 release package as informative appendices and the remainder available from sites referenced in the release package. These materials included descriptions of example implementations, sample instances of each of the UBL documents used in the example implementations, formatting specifications for the United Nations Layout Keys corresponding to each of the UBL basic business document types, and an ASN.1 specification to enable the transmission of UBL messages in binary form. Also included were documents describing the UBL 1.0 Naming and Design Rules, the UBL 1.0 code list mechanism, and UBL 1.0 customization guidelines.

Due to the greatly increased scope of UBL 2.0, the supporting documents and informative materials corresponding to those in the UBL 1.0 Standard are being provided in a separate UBL 2.0 Support Package in order to reduce scheduling dependencies between the normative and informative parts of the specification. The Support Package is being developed in parallel with the UBL 2.0 specification and will be made available after ratification of UBL 2.0 as an OASIS Standard.

## A.6 Taxation Rules

UBL 2.0 does not provide documents for tax reporting. It provides the structures on which tax information is based rather than implementations of specific tax rules.

To implement support for specific tax regimes, the OASIS UBL Technical Committee intends to work with the OASIS TaxXML Technical Committee to provide guidelines that explain how specific tax requirements (e.g., audit trails) are implemented using UBL.

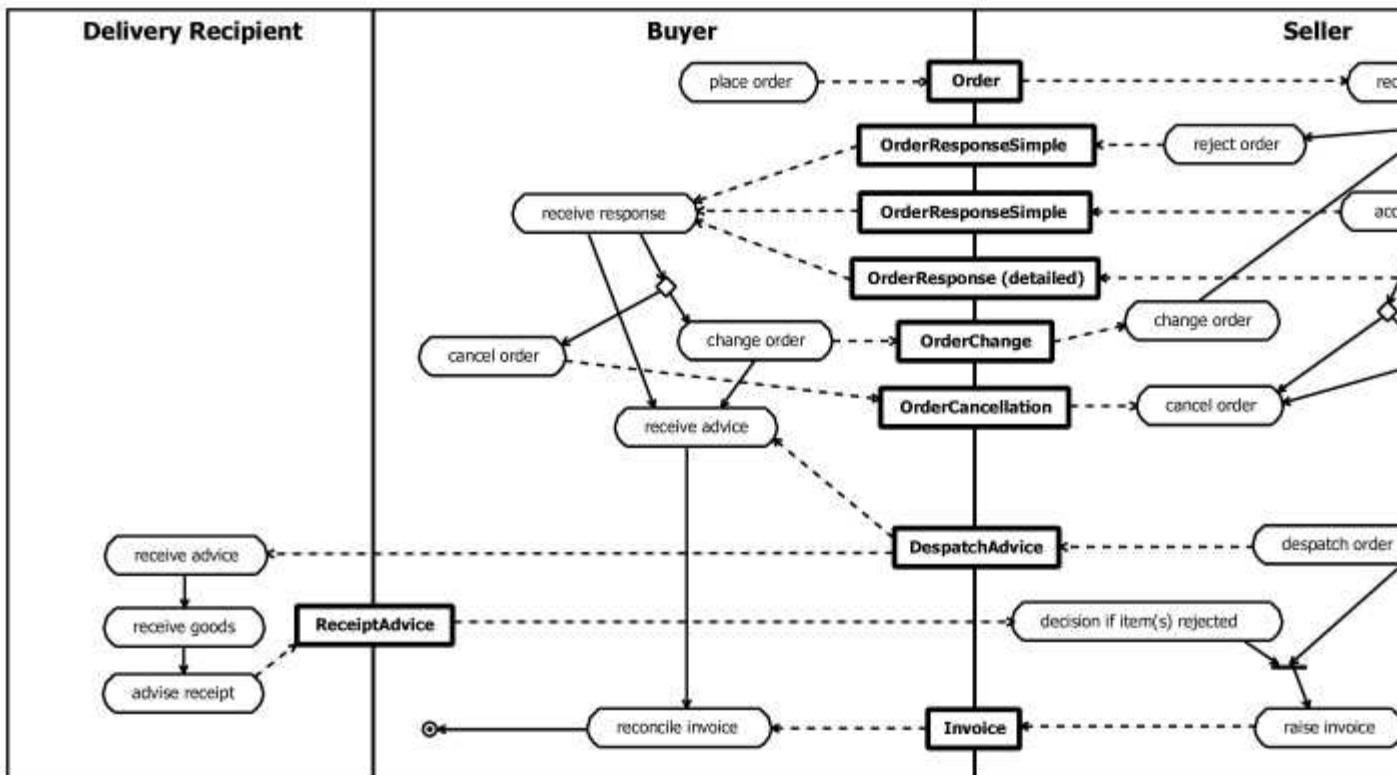
## A.7 UBL Customization

Recommendations for the development of derivative implementations such as subsets, extensions, and contextualized profiles of UBL will be provided as part of the UBL 2.0 Support Package.

# Appendix B (Informative): Upgrading from UBL 1.0 to UBL 2.0

## B.1 The Original UBL 1.0 Order-to-Invoice Process

UBL 2.0 builds upon the basic procurement process established in UBL 1.0. That process, based on eight basic document types shown in bold outline, is illustrated in the diagram below. (See Section 5 for the process contexts assumed for UBL 2.0.)



**Figure 22. UBL 1.0 Order-to-Invoice Business Process**

Though apparently limited in scope, the eight document types provided in UBL 1.0 are applicable to a very large number of real-world use cases and have been widely deployed.

## B.2 New in UBL 2.0

Adoption of UBL 1.0 following ratification as an OASIS Standard in November 2004 has resulted in major inputs of new 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 the United States have resulted in greatly expanded pre-order and post-invoice capabilities together with the addition of several transport-related document types. These additions have increased the number of UBL document types from eight in UBL 1.0 to 31 in UBL 2.0.

**Original UBL 1.0 order-to-invoice document types:** Order, OrderResponse, OrderResponseSimple, OrderChange, OrderCancellation, DespatchAdvice, ReceiptAdvice, Invoice

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

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

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

**New UBL 2.0 document types for payment:** RemittanceAdvice, Statement

**New UBL 2.0 supplementary document types:** ApplicationResponse, AttachedDocument

The role of the 23 new UBL 2.0 document types is described in further detail below.

## B.3 Other Differences between UBL 1.0 and UBL 2.0

While every effort has been made to keep UBL 2.0 backward-compatible with UBL 1.0, several changes resulting from experience with 1.0 have proven extensive enough to make this a major release instead of a minor version update. These changes must be considered in upgrading existing UBL-based systems to take advantage of the greatly expanded applicability of UBL 2.0.

### B.3.1 Global Scoping

In UBL 1.0, the great majority of element types were globally scoped, the only exceptions being identifiers and codes. In UBL 2.0, *all* types are globally scoped.

### B.3.2 Code Lists

The UBL mechanism for specifying and validating code lists has been completely revamped using the power of XSLT [[XSLT](#)] (a W3C Recommendation) and Schematron [[SCH](#)] (ISO/IEC 19757-3) to make it easier to modify code lists and perform basic business rule checking. See Appendix D, UBL 2.0 Code Lists and Two-phase Validation.

### B.3.3 Data Elements

A number of elements have been changed to reflect actual practice, as shown in the following two tables.

Aggregate BIE	Basic or Association BIE	Change in UBL 2.0
Address		

Aggregate BIE	Basic or Association BIE	Change in UBL 2.0
	AddressLine	Changed cardinality to 0..n
AddressLine		
	Line	Changed cardinality to 1
AllowanceCharge		
	CurrencyCode	Removed
BasePrice		
	MaximumQuantity	Removed
	MinimumQuantity	Removed
	MaximumAmount	Removed
	MinimumAmount	Removed
BuyerParty		Renamed to CustomerParty
	BuyerAssignedAccountID	Renamed to CustomerAssignedAccountID
	SellerAssignedAccountID	Renamed to SupplierAssignedAccountID
Delivery		
	DespatchAddress	Replaced by Despatch
DespatchLine		
	Delivery	Replaced by Shipment
	DeliveryTerms	Replaced by Shipment
	Item	
	TransportHandlingUnit	Replaced by Shipment
DocumentReference		
	CopyIndicator	Changed cardinality to 1
InvoiceLine		
	LineStatusCode	Removed

Aggregate BIE	Basic or Association BIE	Change in UBL 2.0
Item		
	SalesConditions	Renamed to TransactionConditions
	TaxCategory	Renamed to ClassifiedTaxCategory
	BasePrice	Removed
LineItem		
	BuyersID	Renamed to ID
	SellersID	Renamed to SalesOrderID
OrderLineReference		
	BuyersLineID	Renamed to LineID and changed cardinality to 1
	SellersLineID	Renamed to SalesOrderLineID
OrderReference		
	BuyersID	Renamed to ID and changed cardinality 1..1
	SellersID	Renamed to SalesOrderID
Party		
	PartyName	Changed cardinality to 0..n
PartyName		
	Name	Changed cardinality to 1
ReceiptLine		
	LineStatusCode	Removed
	Delivery	Replaced by Shipment
	TransportHandlingUnit	Replaced by Shipment
	OrderedItemIdentification	Replaced by Item
SalesConditions		Renamed to TransactionConditions

Aggregate BIE	Basic or Association BIE	Change in UBL 2.0
SellerParty		Renamed to SupplierParty
	BuyerAssignedAccountID	Renamed to CustomerAssignedAccountID
	SellerAssignedAccountID	Renamed to SupplierAssignedAccountID
Shipment		
	TransportEquipment	Replaced by TransportHandlingUnit
TaxCategory		
	ExemptionReason	Removed
TaxScheme		
	JurisdictionAddress	Renamed to JurisdictionRegionAddress
TaxTotal		
	TotalTaxAmount	Renamed to TaxAmount
TransportEquipment		
	Dimension	Renamed to MeasurementDimension
TransportEquipmentSeal		
	IssuerTypeCode	Renamed to SealIssuerTypeCode
TransportHandlingUnit		
	UnitTypeCode	Renamed to TransportHandlingUnitTypeCode

**Table 3. Changes to Library Elements in UBL 2.0**

Aggregate BIE	Basic or Association BIE	Change in UBL 2.0
DespatchAdvice		
	CopyIndicator	Changed cardinality to 1
	BuyerParty	Renamed to BuyerCustomerParty
	SellerParty	Renamed to SellerSupplierParty

Aggregate BIE	Basic or Association BIE	Change in UBL 2.0
	FreightForwarderParty	Replaced by Shipment
	Delivery	Replaced by Shipment
	DeliveryTerms	Replaced by Shipment
	DespatchedTransportHandlingUnit	Replaced by Shipment
	ActualShipment	Replaced by Shipment
Invoice		
	CopyIndicator	Changed cardinality to 1
	InvoiceCurrencyCode	Renamed to DocumentCurrencyCode and changed cardinality to 1
	LineItemCountNumeric	Removed
	BuyerParty	Renamed to BuyerCustomerParty and changed cardinality to 0..1
	SellerParty	Renamed to SellerSupplierParty and changed cardinality to 0..1
	PaymentMeans	Changed cardinality to 0..n
	ExchangeRate	Renamed to TransactionExchangeRate
Order		
	BuyersID	Renamed to ID and changed cardinality to 1
	SellersID	Renamed to SalesOrderID
	CopyIndicator	Changed cardinality to 1
	AcknowledgementResponseCode	Removed
	TransactionCurrencyCode	Renamed to DocumentCurrencyCode and changed cardinality to 1
	EarliestDate	Removed

Aggregate BIE	Basic or Association BIE	Change in UBL 2.0
	ExpiryDate	Removed
	ValidityDurationMeasure	Removed
	TaxTotalAmount	Removed
	LineExtensionTotalAmount	Removed
	TotalPackagesQuantity	Removed
	GrossWeightMeasure	Removed
	NetWeightMeasure	Removed
	NetNetWeightMeasure	Removed
	GrossVolumeMeasure	Removed
	NetVolumeMeasure	Removed
	LineItemCountNumeric	Removed
	ContractDocumentReference	Replaced by Contract
	QuoteDocumentReference	Renamed to QuotationDocumentreference
	BuyerParty	Renamed to BuyerCustomerParty
	SellerParty	Renamed to SellerSupplierParty
	OriginatorParty	Renamed to OriginatorCustomerParty
	SalesConditions	Renamed to TransactionConditions
OrderCancellation		
	CopyIndicator	Changed cardinality to 1
	IssueDateTime	Renamed to IssueDate and changed to Date datatype
	DocumentStatusCode	Removed
	ResponseRequiredIndicator	Removed
	AcceptedIndicator	Removed

Aggregate BIE	Basic or Association BIE	Change in UBL 2.0
	BuyerParty	Renamed to BuyerCustomerParty
	SellerParty	Renamed to SellerSupplierPArty
OrderChange		
	BuyersID	Renamed to ID
	SellersID	Renamed to SalesOrderID
	CopyIndicator	Changed cardinality to 1
	DocumentStatusCode	Removed
	AcknowledgementResponseCode	Removed
	TransactionCurrencyCode	Renamed to DocumentCurrencyCode and changed cardinality to 1
	EarliestDate	Removed
	ExpiryDate	Removed
	ValidityDurationMeasure	Removed
	TaxTotalAmount	Removed
	LineExtensionTotalAmount	Removed
	TotalPackagesCountQuantity	Removed
	GrossWeightMeasure	Removed
	NetWeightMeasure	Removed
	NetNetWeightMeasure	Removed
	GrossVolumeMeasure	Removed
	NetVolumeMeasure	Removed
	LineItemCountNumeric	Removed
	OrderReference	Changed cardinality to 1
	ContractDocumentReference	Replaced by Contract

Aggregate BIE	Basic or Association BIE	Change in UBL 2.0
	QuoteDocumentReference	Renamed to QuotationDocumentReference
	BuyerParty	Renamed to BuyerCustomerParty
	SellerParty	Renamed to SellerSupplierParty
	OriginatorParty	Renamed to OriginatorCustomerParty
	SalesConditions	Renamed to TransactionConditions
OrderResponse		
	BuyersID	Renamed to ID and changed cardinality to 1
	SellersID	Renamed to SalesOrderID
	CopyIndicator	Changed cardinality to 1
	DocumentStatusCode	Removed
	EarliestDate	Removed
	ExpiryDate	Removed
	ValidityDurationMeasure	Removed
	TaxTotalAmount	Removed
	LineExtensionTotalAmount	Removed
	TotalPackagesCountQuantity	Renamed to TotalPackagesQuantity
	LineItemCountNumeric	Removed
	BuyerParty	Renamed to BuyerCustomerParty
	SellerParty	Renamed to SellerSupplierParty
	OriginatorParty	Renamed to OriginatorCustomerParty
	SalesConditions	Renamed to TransactionConditions
	RespondedOrderLine	Renamed to OrderLine

Aggregate BIE	Basic or Association BIE	Change in UBL 2.0
OrderResponseSimple		
	CopyIndicator	Changed cardinality to 1
	DocumentStatusCode	Removed
	BuyerParty	Renamed to BuyerCustomerParty
	SellerParty	Renamed to SellerSupplierParty
ReceiptAdvice		
	CopyIndicator	Changed cardinality to 1
	BuyerParty	Renamed to BuyerCustomerParty
	SellerParty	Renamed to SellerSupplierParty
	FreightForwarderParty	Replaced by Shipment
	Delivery	Replaced by Shipment
	ReceivedTransportHandlingUnit	Replaced by Shipment

**Table 4. Changes to Document Elements in UBL 2.0**

### B.3.4 Attributes

Several attribute names have been changed to implement the UN/CEFACT Core Component Type schemas, as shown in the following table.

Type	Attribute	Change in UBL 2.0
AmountType		
	amountCurrencyID	Renamed to CurrencyID
	amountCurrencyCodeListVersionID	Removed
BinaryObjectType		
	format	Added
	mimeCode	Added
	encodingCode	Added

Type	Attribute	Change in UBL 2.0
	uri	Added
	filename	Added
GraphicType		
	format	Added
	mimeCode	Added
	encodingCode	Added
	uri	Added
	filename	Added
	characterSetCode	Removed
PictureType		
	format	Added
	mimeCode	Added
	encodingCode	Added
	uri	Added
	filename	Added
	characterSetCode	Removed
SoundType		
	format	Added
	mimeCode	Added
	encodingCode	Added
	uri	Added
	filename	Added
	characterSetCode	Removed
VideoType		

Type	Attribute	Change in UBL 2.0
	format	Added
	mimeCode	Added
	encodingCode	Added
	uri	Added
	filename	Added
	characterSetCode	Removed
CodeType		
	codeListID	Renamed to listID
	codeListAgencyID	Renamed to listAgencyID
	codeListAgencyName	Renamed to listAgencyName
	codeListName	Renamed to listName
	codeListVersionID	Renamed to listVersionID
	codeListURI	Renamed to listURI
	codeListSchemeURI	Renamed to listSchemeURI
IdentifierType		
	identificationSchemeID	Renamed to schemeID
	identificationSchemeName	Renamed to schemeName
	identificationSchemeAgencyID	Renamed to schemeAgencyID
	identificationSchemeAgencyName	Renamed to schemeAgencyName
	identificationSchemeVersionID	Renamed to schemeVersionID
	identificationSchemeURI	Renamed to schemeURI
	identificationSchemeDataURI	Renamed to schemeDataURI
MeasureType		
	measureUnitCode	Renamed to unitCode

Type	Attribute	Change in UBL 2.0
	measureUnitCodeListVersionID	Renamed to unitCodeListVersionID
QuantityType		
	quantityUnitCode	Renamed to unitCode
	quantityUnitCodeListID	Removed
	quantityUnitCodeListAgencyID	Removed
	quantityUnitCodeListAgencyName	Removed

**Table 5. Changes to Attributes in UBL 2.0**

### B.3.5 Extension Element

An optional container element named UBLExtensions may now appear as the first child of any UBL 2.0 document. UBLExtensions was provided to meet user demand for an area in which to include non-UBL data elements, in particular, elements containing data whose inclusion is mandated by law for certain business documents in certain regulatory environments. Note that unlike every other data element in UBL, UBLExtensions has no associated business semantics in itself and is therefore not derived from a CCTS data type.

Each ext:UBLExtension child element of the ext:UBLExtensions container element contains the metadata and content associated with a single extension. To accommodate the widest range of possible extensions, the ext:ExtensionContent element is specified in xsd/common/UBL-ExtensionContentDatatype-2.0.xsd as having a single child element of type xsd:any with a processContents value of "skip". This means, in essence, that any well-formed XML element (and all of its children and descendants) from any vocabulary can be the one child of the ext:ExtensionContent element. The metadata recorded for an extension is part of the UBL vocabulary, specified in xsd/common/UBL-CommonExtensionComponents-2.0.xsd as optional elements that are siblings to the ext:ExtensionContent element.

Injudicious use of UBLExtensions will obviously have damaging consequences for interoperability of UBL documents. UBLExtensions should be used with great care and should never be used for data that is properly conveyed in standard UBL elements allowed elsewhere in the document. In general, UBLExtensions should be used only as a last resort for data that cannot be accommodated by the constructs provided in the standard. Practical use of UBLExtensions will require out-of-band agreements among specific trading partner communities together with publication and maintenance procedures outside the scope of Standard UBL.

## B.4 UBL Development Methodology

UBL 2.0 has been developed based on the principles of the ebXML Core Components Technical Specification [\[CCTS\]](#). This entailed developing conceptual models of Basic Information Entities.

To gather requirements and business rules, several workshops were held involving participants from a range of industries and countries.

Based on the defined processes and business rules for the context of use, Business Information Entities were identified and aggregated using normalization techniques to maximize re-use and clarify meanings. This resulted in a comprehensive model of all BIEs relevant to the UBL 2.0 use case. The design strategy has been to provide an 80/20 solution — describing 80 percent of the required components with 20 percent of the complexity. This meant that in some cases, components less commonly used or used only in particular contexts were dropped on the understanding that specific implementations may extend UBL to satisfy these requirements.

The resultant conceptual model of BIE components forms the library from which all UBL document models are assembled.

Every UBL 2.0 conceptual model of a document (known as a document assembly model) contains the “root” ABIE for the document. This itself contains several BBIEs (individual pieces of information) and ASBIEs (associations to other ABIEs in one of the three libraries). This creates the hierachical structure necessary to represent an XML document schema.

The artefacts used to express these conceptual models are UML Class Diagrams and UBL-specific spreadsheets. The spreadsheets are the artefacts used to automatically generate schemas based on the UBL Naming and Design rules. As was the case in UBL 1.0, the UBL 2.0 schemas were generated by EDIFIX (see Appendix C, UBL 2.0 Data Models).

## Appendix C (Informative): UBL 2.0 Document Models

Conceptual models of UBL documents and their component assemblies are expressed using both UML class diagrams and spreadsheets.

Class diagrams are provided as useful graphical guides to the overall UBL structures.

To assist those migrating from UBL 1.0 to UBL 2.0, these diagrams use pink boxes to represent ABIEs that existed in UBL 1.0 and red lines for ASBIEs that existed in UBL 1.0. BBIEs that existed in UBL 1.0 are marked with a "#" symbol. An XMI copy of the UBL 2.0 UML model will be provided as part of the UBL 2.0 Support Package.

Spreadsheets provide the supplementary metadata required by [\[CCTS\]](#). Their format has been developed by UBL and extends the spreadsheet format used for UBL 1.0. They are provided in Open Document (ods) format as well as proprietary xls format. The OASIS Open Document Format for Office Applications ([OpenDocument](#)) is supported by [OpenOffice](#), a free multiplatform, multilingual, open-source office suite.

UBL spreadsheet models can be used to generate XML Schemas using the EDIFIX software tool from [GEFEG](#).

## C.1 The Reusable Library

UBL has been designed as a reusable library of Business Information Entities.

The entire library is provided as a single spreadsheet model:

- [mod/common/UBL-CommonLibrary-2.0.ods](#)
- [mod/common/UBL-CommonLibrary-2.0.xls](#)

To aid readability of the UML class diagrams, this library is graphically presented using three views, based on the primary contexts of use for the given business areas. The diagrams can be navigated using

the  and  arrows:

- [A Common Library view](#) containing ABIEs used throughout the various document types.
- [A Procurement view](#) containing ABIEs used mainly for documents associated with a supply chain.
- [A Transportation view](#) containing ABIEs used mainly for documents associated with the commercial aspects of transporting goods.

## C.2 Document Assembly Models

Every UBL 2.0 document assembly model defines a 'root' ABIE (aggregate container) for the document. This itself contains several BBIEs (individual pieces of information) and ASBIEs (associations to other ABIEs). This creates the hierarachical structure necessary to represent an XML document.

The hierarchy as shown in the UML class diagrams can be navigated using the  and  arrows.

### Application Response

- [Application Response Class Diagram](#)  
[mod/maindoc/UBL-ApplicationResponse-2.0.ods](#)  
[mod/maindoc/UBL-ApplicationResponse-2.0.xls](#)

### Attached Document

- [Attached Document Class Diagram](#)  
[mod/maindoc/UBL-AttachedDocument-2.0.ods](#)  
[mod/maindoc/UBL-AttachedDocument-2.0.xls](#)

### Bill Of Lading

- [Bill Of Lading Class Diagram](#)  
[mod/maindoc/UBL-BillOfLading-2.0.ods](#)  
[mod/maindoc/UBL-BillOfLading-2.0.xls](#)

### Catalogue

- [Catalogue Class Diagram](#)

[mod/maindoc/UBL-Catalogue-2.0.ods](#)  
[mod/maindoc/UBL-Catalogue-2.0.xls](#)

### **Catalogue Deletion**

[Catalogue Deletion Class Diagram](#)  
[mod/maindoc/UBL-CatalogueDeletion-2.0.ods](#)  
[mod/maindoc/UBL-CatalogueDeletion-2.0.xls](#)

### **Catalogue Item Specification Update**

[Catalogue Item Specification Update Class Diagram](#)  
[mod/maindoc/UBL-CatalogueItemSpecificationUpdate-2.0.ods](#)  
[mod/maindoc/UBL-CatalogueItemSpecificationUpdate-2.0.xls](#)

### **Catalogue Pricing Update**

[Catalogue Pricing Update Class Diagram](#)  
[mod/maindoc/UBL-CataloguePricingUpdate-2.0.ods](#)  
[mod/maindoc/UBL-CataloguePricingUpdate-2.0.xls](#)

### **Catalogue Request**

[Catalogue Request Class Diagram](#)  
[mod/maindoc/UBL-CatalogueRequest-2.0.ods](#)  
[mod/maindoc/UBL-CatalogueRequest-2.0.xls](#)

### **Certificate Of Origin**

[Certificate Of Origin Class Diagram](#)  
[mod/maindoc/UBL-CertificateOfOrigin-2.0.ods](#)  
[mod/maindoc/UBL-CertificateOfOrigin-2.0.xls](#)

### **Credit Note**

[Credit Note Class Diagram](#)  
[mod/maindoc/UBL-CreditNote-2.0.ods](#)  
[mod/maindoc/UBL-CreditNote-2.0.xls](#)

### **Debit Note**

[Debit Note Class Diagram](#)  
[mod/maindoc/UBL-DebitNote-2.0.ods](#)  
[mod/maindoc/UBL-DebitNote-2.0.xls](#)

### **Despatch Advice**

[Despatch Advice Class Diagram](#)  
[mod/maindoc/UBL-DespatchAdvice-2.0.ods](#)  
[mod/maindoc/UBL-DespatchAdvice-2.0.xls](#)

### **Forwarding Instruction**

[Forwarding Instruction Class Diagram](#)  
[mod/maindoc/UBL-ForwardingInstructions-2.0.ods](#)  
[mod/maindoc/UBL-ForwardingInstructions-2.0.xls](#)

### **Freight Invoice**

[Freight Invoice Class Diagram](#)  
[mod/maindoc/UBL-FreightInvoice-2.0.ods](#)  
[mod/maindoc/UBL-FreightInvoice-2.0.xls](#)

### **Invoice**

[Invoice Class Diagram](#)  
[mod/maindoc/UBL-Invoice-2.0.ods](#)  
[mod/maindoc/UBL-Invoice-2.0.xls](#)

### **Order**

[Order Class Diagram](#)

[mod/maindoc/UBL-Order-2.0.ods](#)

[mod/maindoc/UBL-Order-2.0.xls](#)

### **Order Cancellation**

[Order Cancellation Class Diagram](#)

[mod/maindoc/UBL-OrderCancellation-2.0.ods](#)

[mod/maindoc/UBL-OrderCancellation-2.0.xls](#)

### **Order Change**

[Order Change Class Diagram](#)

[mod/maindoc/UBL-OrderChange-2.0.ods](#)

[mod/maindoc/UBL-OrderChange-2.0.xls](#)

### **Order Response**

[Order Response Class Diagram](#)

[mod/maindoc/UBL-OrderResponse-2.0.ods](#)

[mod/maindoc/UBL-OrderResponse-2.0.xls](#)

### **Order Response Simple**

[Order Response Simple Class Diagram](#)

[mod/maindoc/UBL-OrderResponseSimple-2.0.ods](#)

[mod/maindoc/UBL-OrderResponseSimple-2.0.xls](#)

### **Packing List**

[Packing List Class Diagram](#)

[mod/maindoc/UBL-PackingList-2.0.ods](#)

[mod/maindoc/UBL-PackingList-2.0.xls](#)

### **Quotation**

[Quotation Class Diagram](#)

[mod/maindoc/UBL-Quotation-2.0.ods](#)

[mod/maindoc/UBL-Quotation-2.0.xls](#)

### **Receipt Advice**

[Receipt Advice Class Diagram](#)

[mod/maindoc/UBL-ReceiptAdvice-2.0.ods](#)

[mod/maindoc/UBL-ReceiptAdvice-2.0.xls](#)

### **Reminder**

[Reminder Class Diagram](#)

[mod/maindoc/UBL-Reminder-2.0.ods](#)

[mod/maindoc/UBL-Reminder-2.0.xls](#)

### **Remittance Advice**

[Remittance Advice Class Diagram](#)

[mod/maindoc/UBL-RemittanceAdvice-2.0.ods](#)

[mod/maindoc/UBL-RemittanceAdvice-2.0.xls](#)

### **Request For Quotation**

[Request For Quotation Class Diagram](#)

[mod/maindoc/UBL-RequestForQuotation-2.0.ods](#)

[mod/maindoc/UBL-RequestForQuotation-2.0.xls](#)

### **Self Billed Credit Note**

[Self Billed Credit Note Class Diagram](#)

[mod/maindoc/UBL-SelfBilledCreditNote-2.0.ods](#)

[mod/maindoc/UBL-SelfBilledCreditNote-2.0.xls](#)

### **Self Billed Invoice**

[Self Billed Invoice Class Diagram](#)

[mod/maindoc/UBL-SelfBilledInvoice-2.0.ods](#)  
[mod/maindoc/UBL-SelfBilledInvoice-2.0.xls](#)

#### **Statement**

[Statement Class Diagram](#)  
[mod/maindoc/UBL-Statement-2.0.ods](#)  
[mod/maindoc/UBL-Statement-2.0.xls](#)

#### **Transportation Status**

[Transportation Status Class Diagram](#)  
[mod/maindoc/UBL-TransportationStatus-2.0.ods](#)  
[mod/maindoc/UBL-TransportationStatus-2.0.xls](#)

#### **Waybill**

[Waybill Class Diagram](#)  
[mod/maindoc/UBL-Waybill-2.0.ods](#)  
[mod/maindoc/UBL-Waybill-2.0.xls](#)

## **Appendix D (Informative): UBL 2.0 Code Lists and Two-phase Validation**

### **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 are specified directly in the UBL schemas as enum (enumeration) constraints. This allows 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 makes 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 assumes a two-phase validation model. In the first validation phase, the UBL instance is checked for structure and vocabulary against a standard UBL 2.0 XSD schema using a generic XSD validator (or custom-built software performing the same function). This is exactly the same procedure used in UBL 1.0, except that the UBL 2.0 schemas (with a few exceptions noted below) do not contain default code list values. In the second validation phase, new in UBL 2.0, code list values in the instance are checked against external code list configuration files using an XSLT 1.0 processor driven by an XSLT 1.0 stylesheet. The default values assumed by the UBL 2.0 specification are provided in a file named `defaultCodeList.xsl` located in the `val` directory, as described in more detail below.

The separation of structural and vocabulary checking from code value checking 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 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.

The process for creating custom XSLT code list files to enable this context-specific functionality is described in a separate specification called the UBL Code List Methodology, a copy of which can be obtained from the UBL TC web site at OASIS. A set of support files to aid implementors in creating custom XSLT code list files will be found in a separate support package available from the same site.

## 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 perform default validation of UBL 2.0 documents is included in the `val` directory of this release package. The default validation assumes a Linux or Windows XP 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 test UBL 2.0 default validation:

1. Change to the `val` directory.
2. From within that directory, enter the test command

`test.bat` (XP)

or

`./test.sh` (Linux)

The output, which is explained in the next section, should resemble the following (the spacing has been adjusted to make this easier to read):

```
#####
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. You can now validate any UBL document against the UBL 2.0 runtime schemas by executing commands of the form

```
validate <appropriate-schema> <UBL-document>
```

where `<UBL-document>` is the path of a document to be validated and `<appropriate-schema>` is the UBL 2.0 schema for that document type (Order, Invoice, etc.). For example, the script `testeg.bat` (or `testeg.sh`) shows 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 in D.2 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` (XP) 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 2.0 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.

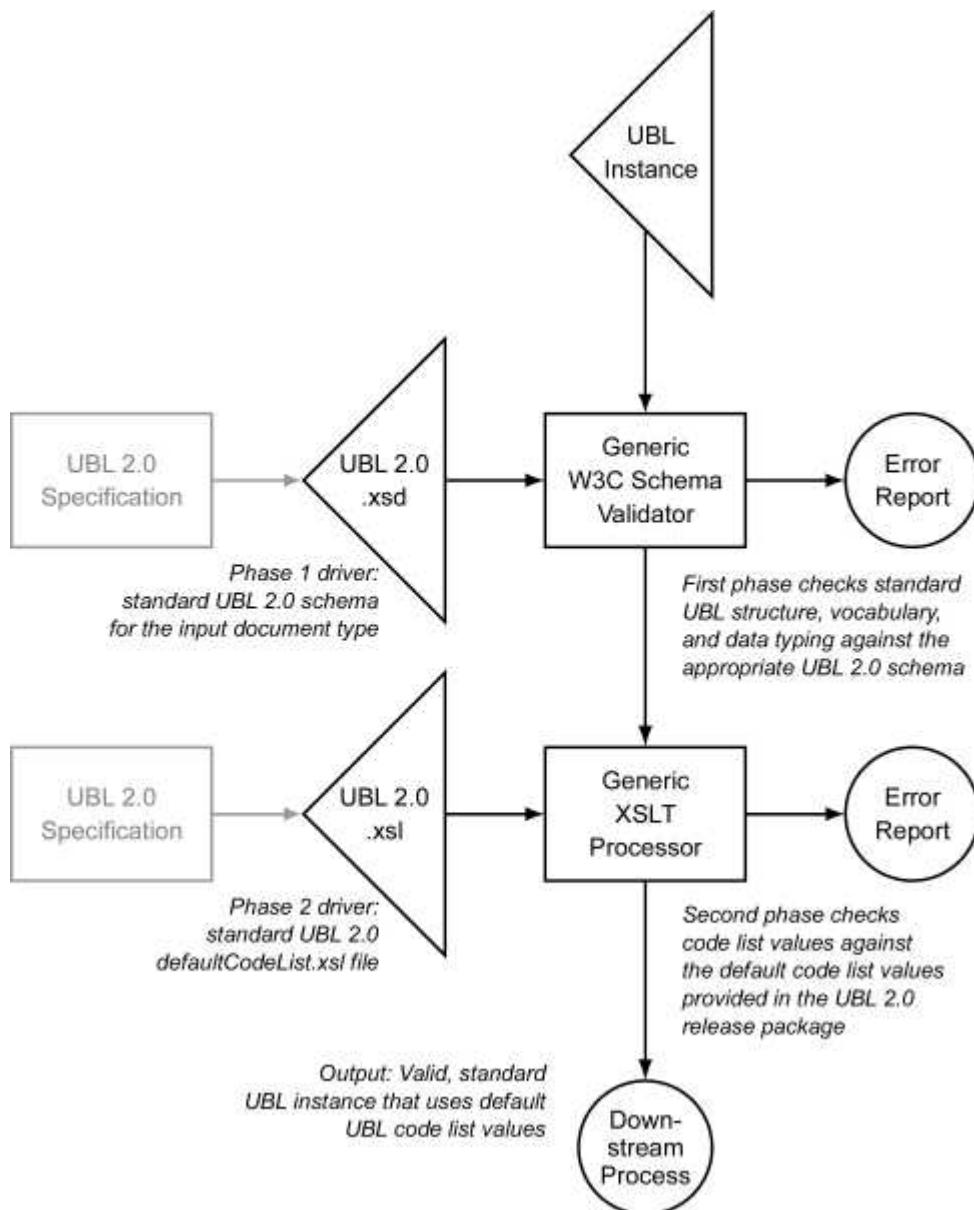
The second phase of validation uses a standard XSLT 1.0 engine, saxon, to verify that the values of various codes used in the UBL document to be tested (country codes, currency codes, etc.) are valid in terms of the default UBL 2.0 code list values provided in `defaultCodeList.xsl`. Here the output line “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 the values in `defaultCodeList.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 `defaultCodeList.xsl`. This process is illustrated in the following diagram.



**Figure 23. Two-phase default UBL 2.0 validation**

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, associate different versions of the same code list with different parts of the same UBL document, and even perform fairly sophisticated business rule checking, simply by building additional logic into the XSLT file that drives the second validation phase — and without changing the standard UBL 2.0 schemas. Schematron-based techniques for creating a custom XSLT file to take the place of `defaultCodeList.xsl` are explained in the UBL Codelist Methodology, the latest draft of which is available from the UBL TC web site. Using these techniques, the business analyst can offload a large proportion of input filtering from the backend business application to a simpler input processing area. And, of course, 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 sophisticated 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 you can update these components as later releases become available.

- The files `resolver.jar` and `xercesImpl.jar` are taken 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 taken from the `xjparse` 1.0 distribution at  
<http://nwalsh.com/java/xjparse/>
- The file `saxon.jar` is taken from the `saxon` 6.5.5 distribution at  
<http://prdownloads.sourceforge.net/saxon/saxon6-5-5.zip>

## D.6 Code list documentation

While the `defaultCodeList.xsl` file is what actually drives the second validation phase where the code list values get checked, it doesn't function well as documentation of those values. For listings of the default codes, it's better to consult the separate code list files from which `defaultCodeList.xsl` was compiled.

These files, which can be found in the `c1/gc` directory, use an XML format called genericode that is specially designed to represent code lists. The version of genericode adopted for this release is an early draft that is now being worked on by another OASIS technical committee. While still unfinished, this version provides all of the functionality needed for UBL and is the one intended for use in the UBL 2.0 Code List Support Package.

The genericode files are separated into three subdirectories as follows:

### D.6.1 cl/gc/default

These code lists contain most of the default codes represented in `defaultCodeList.xsl`. Note that the majority of these code lists are “placebos” or placeholders included to provide extension points for users wishing to assign their own code values when generating custom XSLT files. The files in this directory that contain actual default code values are:

```
AllowanceChargeReasonCode  
ChannelCode  
ChipCode  
ConditionCode  
DocumentStatusCode  
IdentificationCode  
LatitudeDirectionCode  
LineStatusCode  
LongitudeDirectionCode  
OperatorCode  
PackagingTypeCode  
PaymentMeansCode  
SubstitutionStatusCode  
TransportModeCode
```

### D.6.2 cl/gc/cefact

This directory contains genericode versions of three standard code lists (currency codes, unit codes, and MIME content codes) specified by UN/CEFACT (United Nations Centre for Trade Facilitation and Electronic Business). Unlike all other code values in UBL 2.0, the values documented in these three files are “hardwired” into the UBL schemas as a result of UBL’s adoption of the UN/CEFACT unqualified data type (UDT) module. Consequently, these codes are actually checked twice — once during the first validation phase against the codes bound into the UBL schemas via the UDT module, and then once again against the same codes compiled into `defaultCodeList.xsl`. Of course, any nonstandard value used for one of these codes will end the validation in the first phase.

The practical result of this is that code values can be removed from any of these three code lists (for example, the set of acceptable currencies could be narrowed down to just the currencies used by a company’s trading partners), but no values can be added. This is because customizing the XSLT file so that a given code list has *fewer* values will trap the omitted values in the second validation phase, but customizing the same file to give the code list *additional* values will have no effect, because an instance of one of the new values will be trapped in the first validation phase before the second phase can be applied.

In summary: the three code lists in the `cl/cefact` directory can only be subsetted; they cannot be extended. As in the case of the default UBL code lists, the genericode files containing the UN/CEFACT code lists also serve as documentation of the code values. The schema modules from which these “hardwired” values are actually imported into the UBL document schemas can be found in the `xsd/common` directory in files whose names begin `CodeList_`.

### D.6.3 cl/gc/special-purpose

This directory contains genericode versions of three code lists that are used only in certain application contexts. Due to the large size of these lists, they are not included in `defaultCodeList.xsl`, but are

provided here so that they can be incorporated into custom XSLT scripts.

The files in this directory are:

ContainerSizeCodeType.gc  
PortCodeType.gc  
UnitOfMeasureCodeType.gc

## D.6.4 cl/xsdcl

This directory contains two directories of XSD schema fragments expressing enumeration constraints mirroring the coded values in the genericode files described in sections D.6.1 and D.6.3. These are provided here only as a convenience for users who may wish to modify their schema expressions to incorporate enumeration constraints. These files do not comprise part of standard UBL.

# Appendix E (Informative): UBL 2.0 Naming and Design Rules

The XML Naming and Design Rules (NDRs) used in creating the UBL schemas in this draft specification are given in the checklist at [NDR-checklist-2006-07-19.pdf](#). The entire NDR document (including explanatory prose) will be released following publication of UBL 2.0.

# Appendix F (Informative): ASN.1 Specification

The UBL ASN.1 specification referenced 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 in Section 6 that constitute the normative definitions of valid UBL documents. The UBL ASN.1 XML schema 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.

### UBL 2.0 ASN.1 Specification

[asn/ASN.1-UBL-2.0.html](#)

The ASN.1 UBL 2.0 specification was 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. After conversion, the generated ASN.1 was formatted by the PrettyPrint tool at the [ASN.1 Information Site](#) (<http://asn1.elibel.tm.fr>) to produce the HTML file included in this package.

# Appendix G: Notices

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's procedures with respect to rights in OASIS specifications can be found at 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 implementors or users of this specification, can be obtained from the OASIS Executive Director.

OASIS invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights which may cover technology that may be required to implement this specification. Please address the information to the OASIS Executive Director.

Copyright © OASIS Open 2006. All Rights Reserved.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this paragraph are included on all such copies and derivative works. However, this document itself may not be modified in any way, such as by removing the copyright notice or references to OASIS, except as needed for the purpose of developing OASIS specifications, in which case the procedures for copyrights defined in the OASIS Intellectual Property Rights document must be followed, or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by OASIS or its successors or assigns.

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 RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.