





urn:oasis:names:specification:UBL:schema:xsd:Statement-2  
urn:oasis:names:specification:UBL:schema:xsd:StockAvailabilityReport-2  
urn:oasis:names:specification:UBL:schema:xsd:Tender-2  
urn:oasis:names:specification:UBL:schema:xsd:TendererQualification-2  
urn:oasis:names:specification:UBL:schema:xsd:TendererQualificationResponse-2  
urn:oasis:names:specification:UBL:schema:xsd:TenderReceipt-2  
urn:oasis:names:specification:UBL:schema:xsd:TradeItemLocationProfile-2  
urn:oasis:names:specification:UBL:schema:xsd:TransportationStatus-2  
urn:oasis:names:specification:UBL:schema:xsd:TransportationStatusRequest-2  
urn:oasis:names:specification:UBL:schema:xsd:TransportExecutionPlan-2  
urn:oasis:names:specification:UBL:schema:xsd:TransportExecutionPlanRequest-2  
urn:oasis:names:specification:UBL:schema:xsd:TransportProgressStatus-2  
urn:oasis:names:specification:UBL:schema:xsd:TransportProgressStatusRequest-2  
urn:oasis:names:specification:UBL:schema:xsd:TransportServiceDescription-2  
urn:oasis:names:specification:UBL:schema:xsd:TransportServiceDescriptionRequest-2  
urn:oasis:names:specification:UBL:schema:xsd:UnawardedNotification-2  
urn:oasis:names:specification:UBL:schema:xsd:UtilityStatement-2  
urn:oasis:names:specification:UBL:schema:xsd:Waybill-2

**Abstract:**

This specification defines the Universal Business Language, version 2.1.

**Status:**







3.1.39 Product Activity .....	104
3.1.40 Quotation .....	105

## **Appendices**

A Release Notes (Non-Normative) .....	136
A.1 Availability .....	136
A.2 Status of this Release.....	136
A.3 Package Structure .....	136
A.4 Support .....	137
A.5 UBL Customization .....	137
A.6 Upgrading from UBL 2.0 to UBL 2.1 .....	137
A.7 Dictionary Entry Name Corrections in UBL 2.1.....	137
B Revision History .....	138



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.

**AdES**

Advanced Electronic Signature

**ASBIE**

Association Business Information Entity

**BBIE**

Basic Business Information Entity

**BIE**

Business Information Entity

**C14N**

Canonicalization

**CC**

Core Component

**CPFR**

Collaborative Planning, Forecasting, and Replenishment [[CPFR](#)]

**CV2**

Credit Card Verification Numbering System

**DSig**

Digital Signature

**EDI**

Electronic Data Interchange

**ISO**

International Organization for Standardization

**NDR**

UBL Naming and Design Rules

**QC**

Qualified Certificate

**QS**

Qualified Signature

**UML**

Unified Modeling Language [[UML](#)]

**UN/CEFACT**

United Nations Centre for Trade Facilitation and -12 Td[(United Nations Centre f)3United Nations Centre f

AdvancedXML Electronic Signature[CPFR](#)



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

[RFC3161] *Internet X.509 Public Key Infrastructure Time-Stamp Protocol (TSP), August 2001* [<http://www.faqs.org/rfcs/rfc3161.html>].

[UBL-2.1-ASN.1] *UBL 2.1 Abstract Syntax Notation 1 (ASN.1) AlternativeVersion 1.0.*  
Latest v <http://docs.oasis-open.org/ubl/UBL-2.1-ASN.1/v1.0/UBL-2.1-ASN.1-v1.0.html>.

[UBL-2.1-UML] *UBL 2.1 Unified Modeling Language (UML) AlternativeVersion 1.0.*  
Latest v <http://docs.oasis-open.org/ubl/UBL-2.1-UML/v1.0/UBL-2.1-UML-v1.0.html>.

[UN/CEFACT Rec. 37] *Signed Digital Evidence Interoperability Recommendation, 27 September 2010*  
[[http://www.unece.org/cefact/cf\\_plenary/plenary10/ECE\\_TRADE\\_C\\_CEFAC\\_2010\\_14E.pdf](http://www.unece.org/cefact/cf_plenary/plenary10/ECE_TRADE_C_CEFAC_2010_14E.pdf)].

[XAdES (ISO)] *ISO 14533-2:2012 Processes, data elements and documents I( commerce)15(, I(distr)-30(y)JTJ-0.55*  
*AdvXAdES) [http://www.iso.org/iso/*  
*catalogue\_detail?csnumber=56025]*

[XML C14N] John Boyer, *Canonical XML Version 1.0, 15 March 2001* [<http://www.w3.org/TR/2001/REC-xml-c14n-20010315>].

[XPath 2.0] Anders Berglund, et al., *XML Path Language (XPath) Version 2.0, 23 January 2007* [<http://www.w3.org/TR/2007/REC-xpath20-20070123>].

[XPointer] Steve, et al., *XML Pointer Language (XPointer) Version 1.0 Working Draft, 16 August 2002* [<http://www.w3.org/TR/xptr/>].

[XSLT 2.0] Michael Kay, *XSL TransformXSLT) Version 2.0, 2007-01-23* [<http://www.w3.org/TR/xslt20/>].

---

## The fine line

cord the use of UBL 2.1 business documents. Thoroughly examined and extended the general business supply chain processes of UBL.

## 2 UBL 2.1 Business Objects



### **2.1.3 Item Instances**

Certain Items may be identified and ordered as individual, unique objects—for example, a specific car

*tained in the line.* Thus, despite the name, a `LineItem` is not an `Item` but rather a complex data structure that *contains* an `Item`

Often, however, a single contractual shipment is split up into separate physical consignments that may be received on separate schedules, as shown in [Figure 3, “Split Fulfilment”](#). The shipper may use multiple

So far, the shipper (here a supplier) remains the only consignor and the recipient (here the buyer) the only consignee. But sometimes the division of a shipment into consignments takes place "behind the scenes" through the involvement of a freight forwarder, who becomes both a second consignee and a second consignor ([Figure 4, "Intermediary Fulfilment"](#)

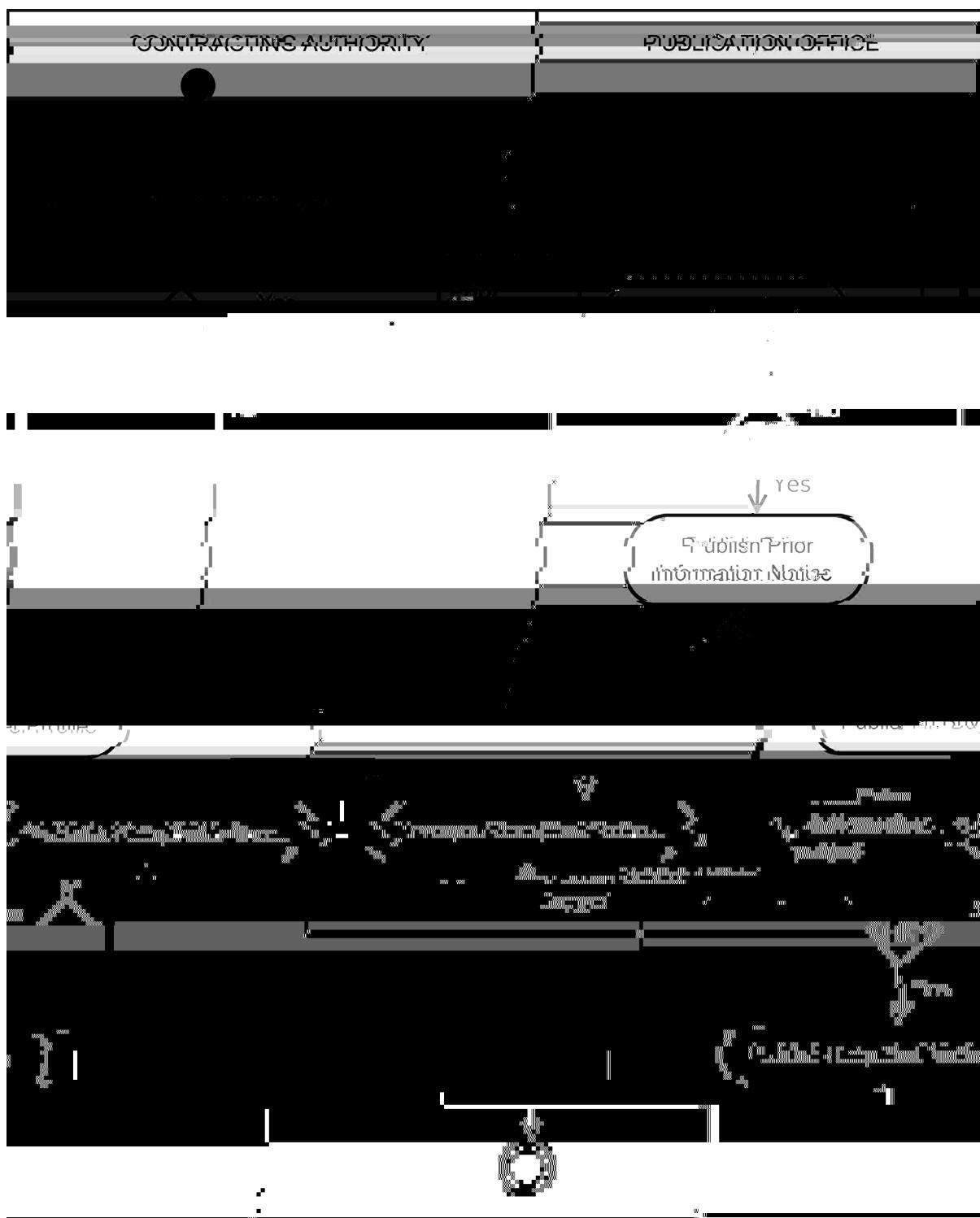
Another layer of complexity is introduced when pieces of different, possibly unrelated shipments are consolidated into a single consignment to make the physical process more efficient (to share space in the same shipping container, for example, which optimizes transport by ensuring that the container is

## **2.1.12 Financial Information**

UBL 2.1 has been enhanced to support the financial information required for downstream processing



Figure 7. Contract Information Preparation





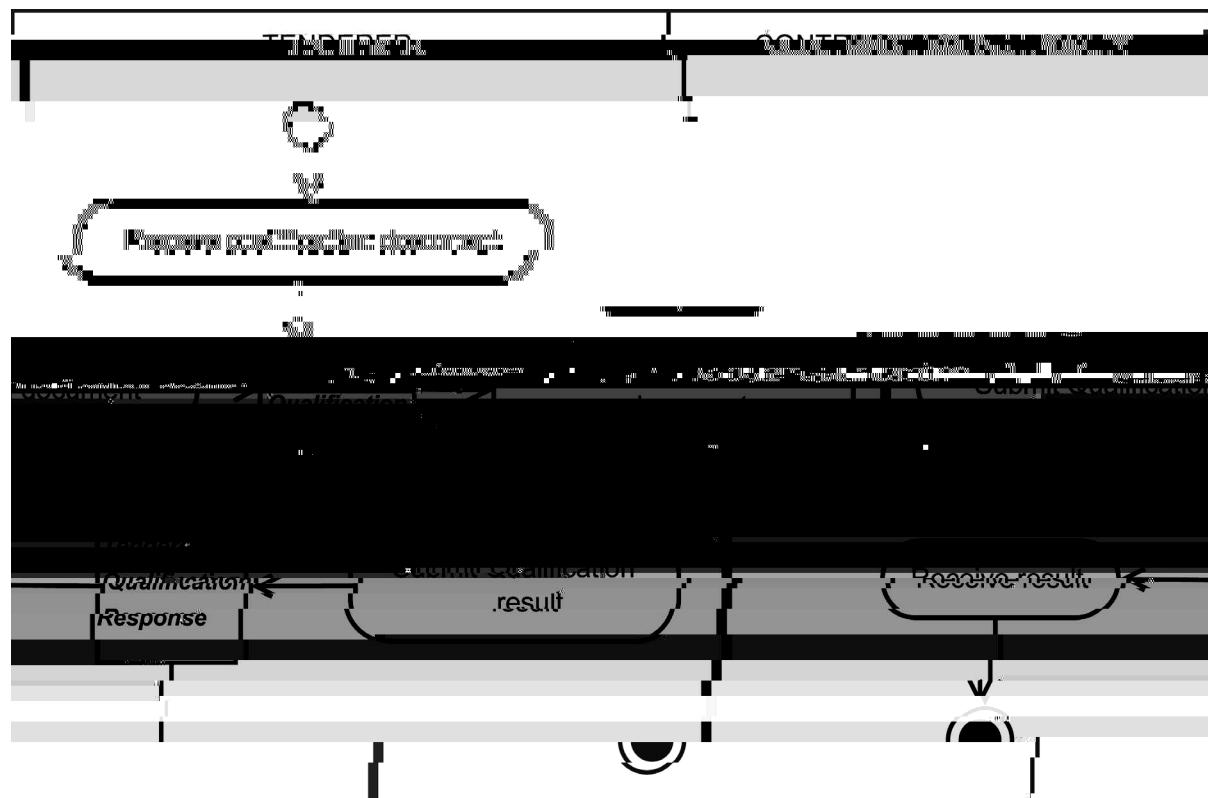
### **2.3.3 Invitation to Tender**

In some procedures, the Contracting Authority invites economic operators to participate in a contest by sending them an invitation to tender us510<sup>2010</sup> tender A

## 2.3.4 Submission of Qualification Information

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

*Figure 10. Submission of Qualification Information*



## **2.3.5 Submission of Tenders**

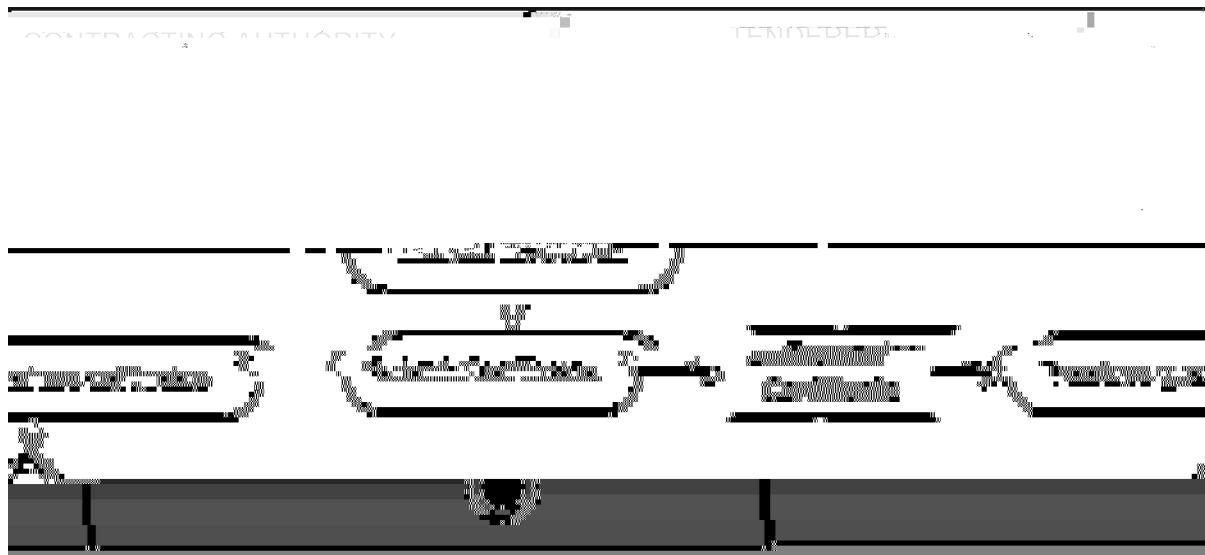
Tenderer is required to submit one or more electronic documents that offer a tender to the Contracting Authority for bid. The Contracting Authority responds with a [Tender Receipt](#) to *notify the reception of the tender* for a tendering process. The date and time of the Tender Receipt are significant, because tendering procedures usually have strict deadlines for tender presentation.

*Figure 11. Submission of Tenders*



Finally, the Tenderer sends a [Guarantee Certificate](#) to *notify the deposit of a guarantee*.

*Figure 14. Guarantee Deposit*

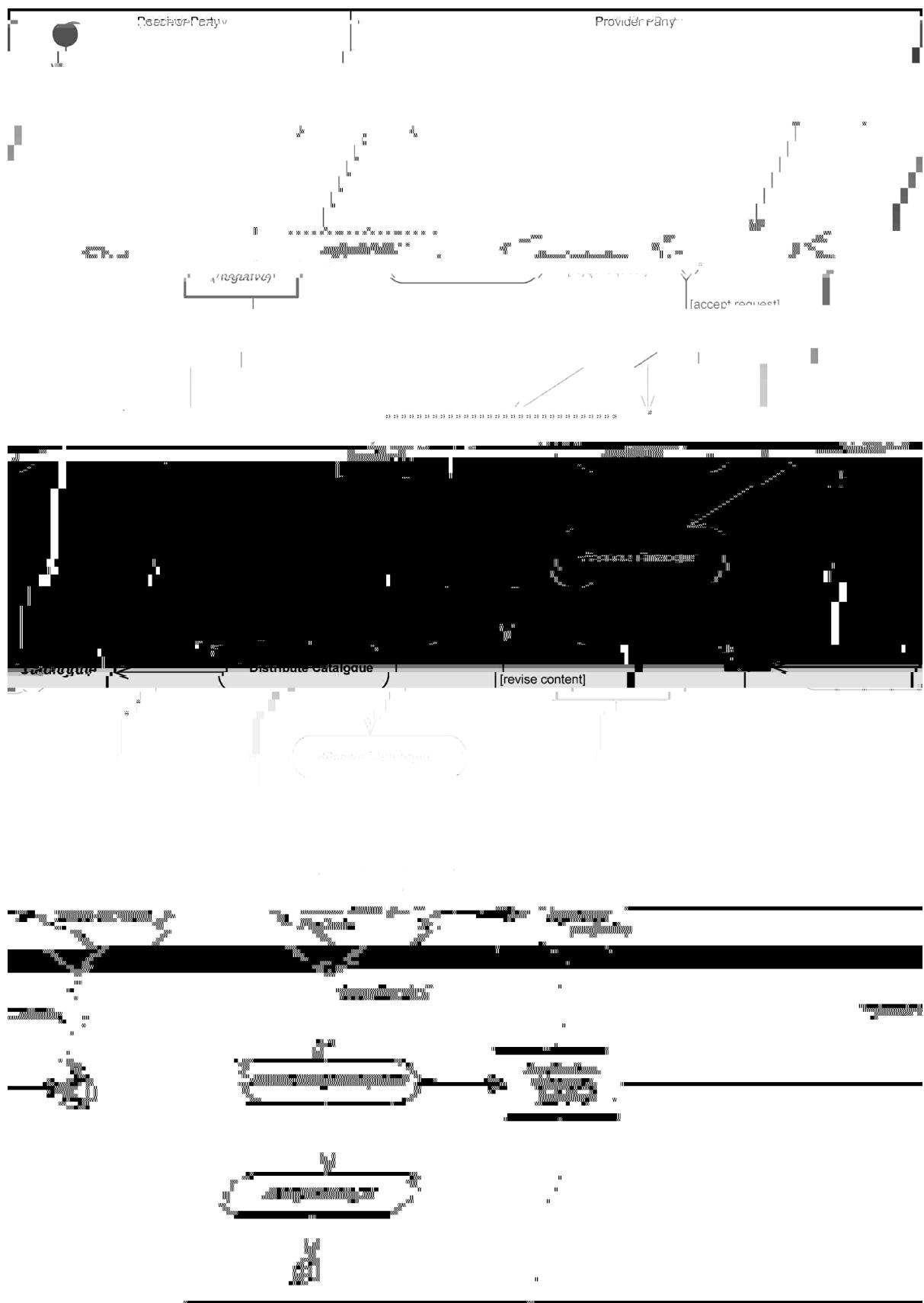




## **2.4.2 Catalogue Provision**

Catalogue provision is the case where a vendor sends information regarding items available for purchase.

*Figure 15. Create Catalogue Process*



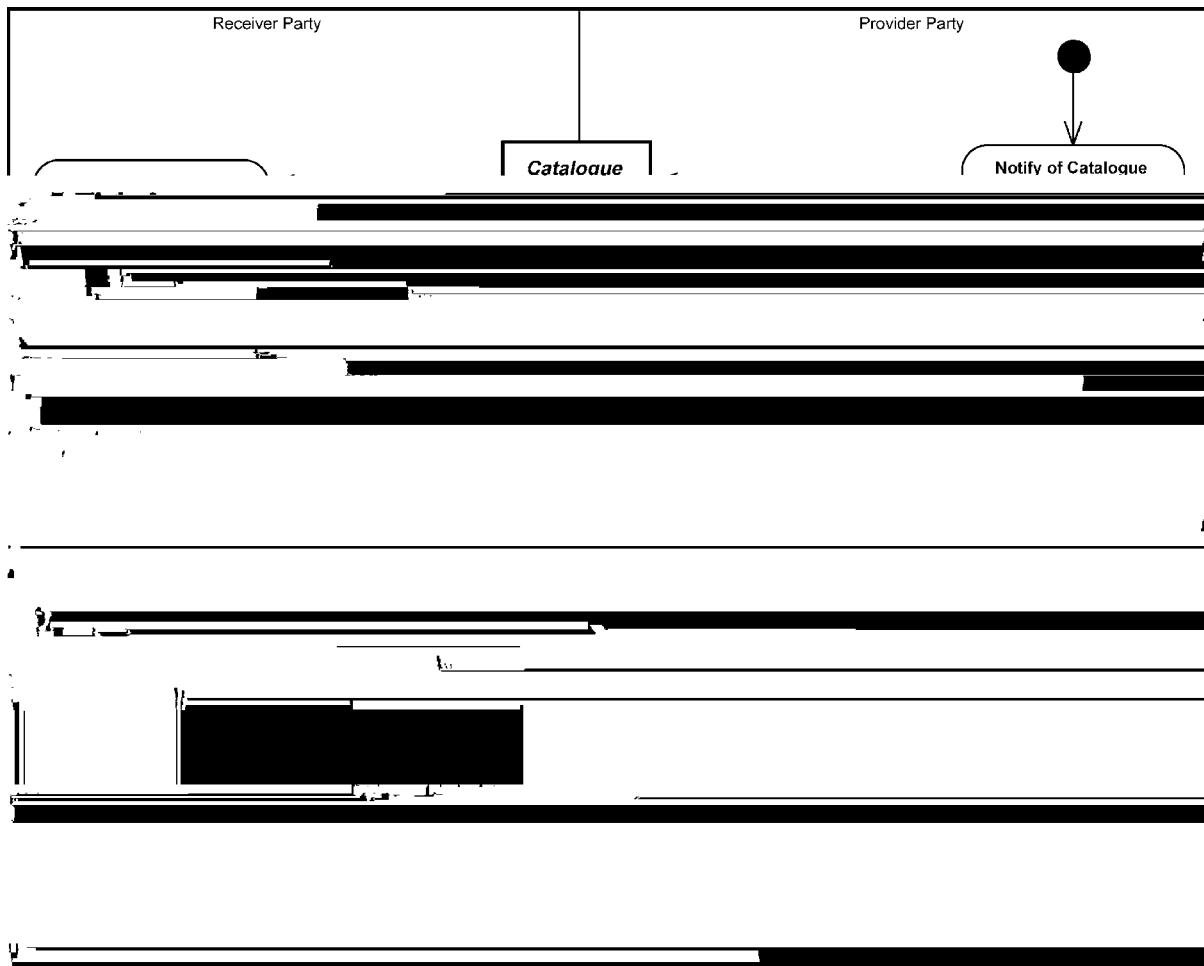




#### 2.4.2.4 Delete Catalogue

Deletion of a Catalogue using [Catalogue Deletion](#) and [Application Response](#) is shown in the following diagram.

*Figure 18. Delete Catalogue Process*







## 2.6 Ordering

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

*Figure 21. Ordering Process*

267.



## **2.7 Fulfilment**

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

2. Organization of the delivery set of items by Despatch Line, annotated by the Transport Handling

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

### **2.7.3 Fulfilment Cancellation Business Rules**

In real life, the sender of a Despatch Advice or Receipt Advice sometimes needs to cancel the document after it has been sent. The Fulfilment Cancellation document is provided for this purpose.

For example, a Despatch Advice may later be cancelled by the Supplier when a problem with shipment prevents the delivery of goods, or the goods to be shipped are not available, or the order is cancelled; in these cases, the customer cancels receipt and adjusts the order accordingly (see Figure 22, "Fulfilment with Despatch Advice Process").

Similarly, a Receipt Advice may later be cancelled by the customer (see Figure 23, "Fulfilment with Receipt Advice Process") if the customer discovers an error in ordering (failure to follow formal contractual obligations, incorrect product identification, etc.) or a of a Despatch Advice deliver



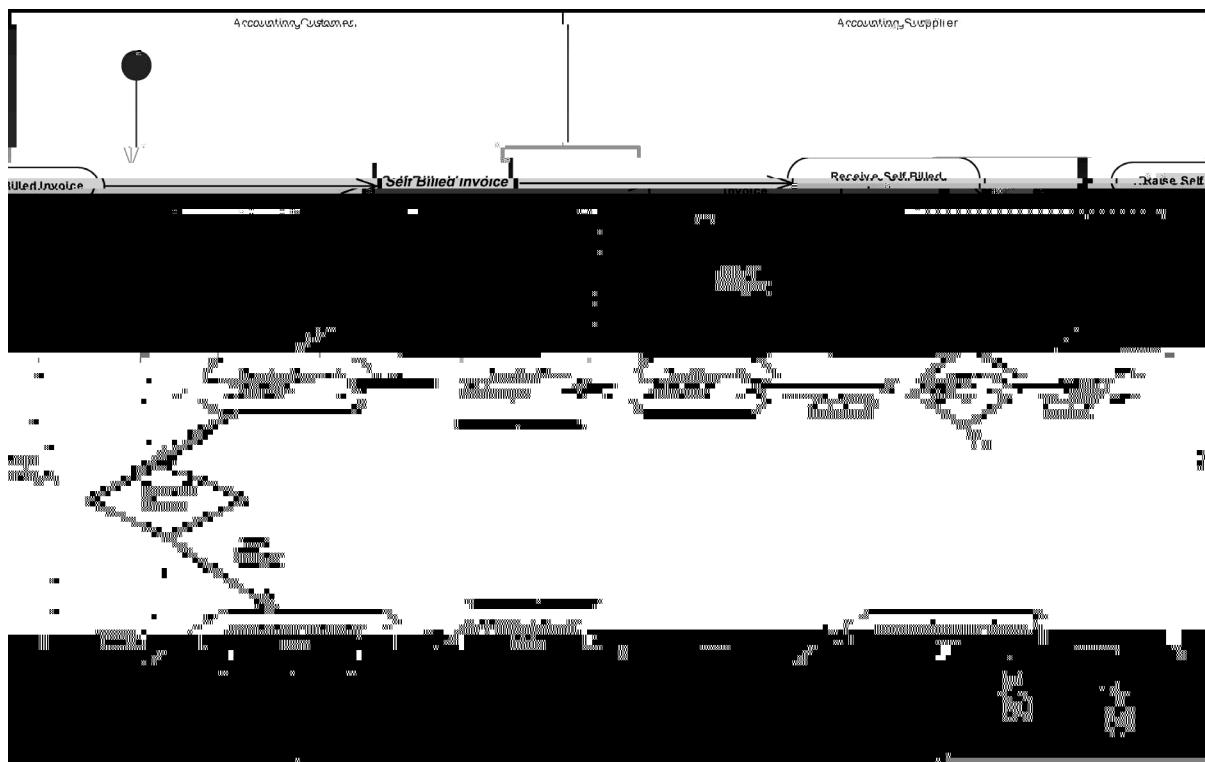




### 2.8.3.2 Self Billing Using Self Billed Credit Notes

Self Billing using [Self Billed Credit Note](#) is shown in the following diagram.

Figure 27. Self Billing with Self Billed Credit Note Process



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

## **2.8.4 Reminder for Payment**

A [Reminder](#) may be used to notify the Customer of accounts due to be paid.

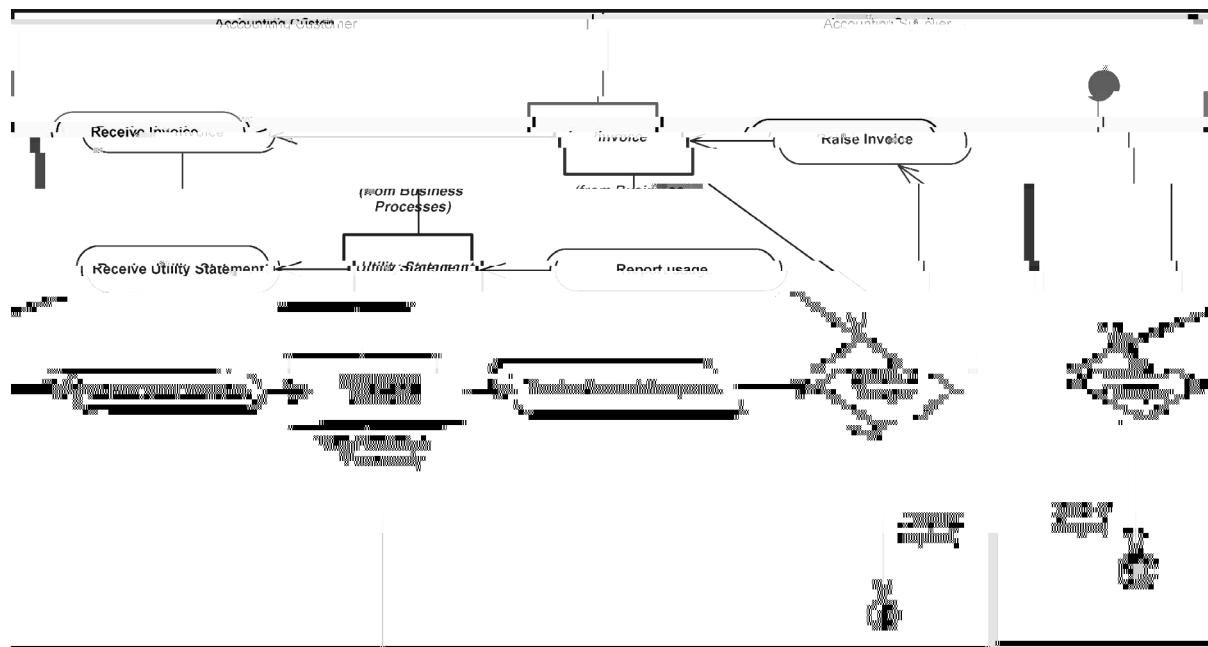
*Figure 28. Reminder for Payment Process*

## 2.10 Utility Billing

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

The [Utility Statement](#) supplements an [Invoice](#) with information about consumption of the utility's services. An invoice may refer to one or more utility statements, and a utility statement may refer to one or more invoices.

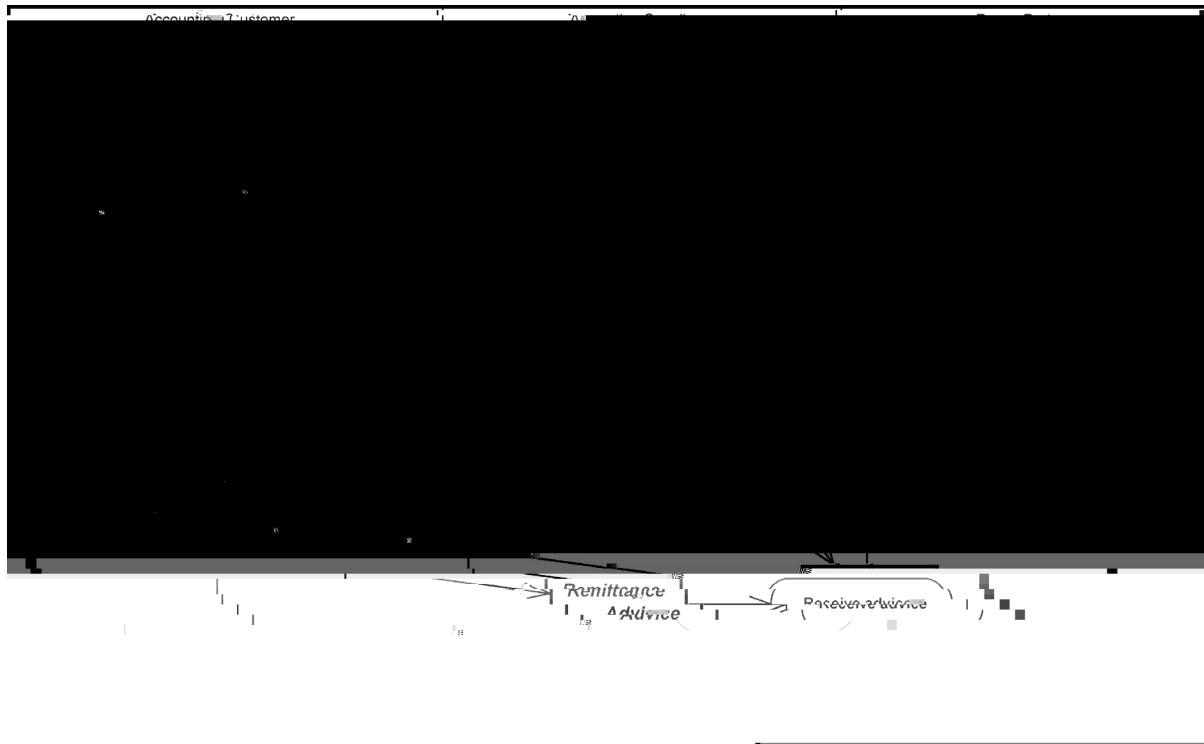
*Figure 30. Utility Billing Process*



## 2.11 Payment Notification

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

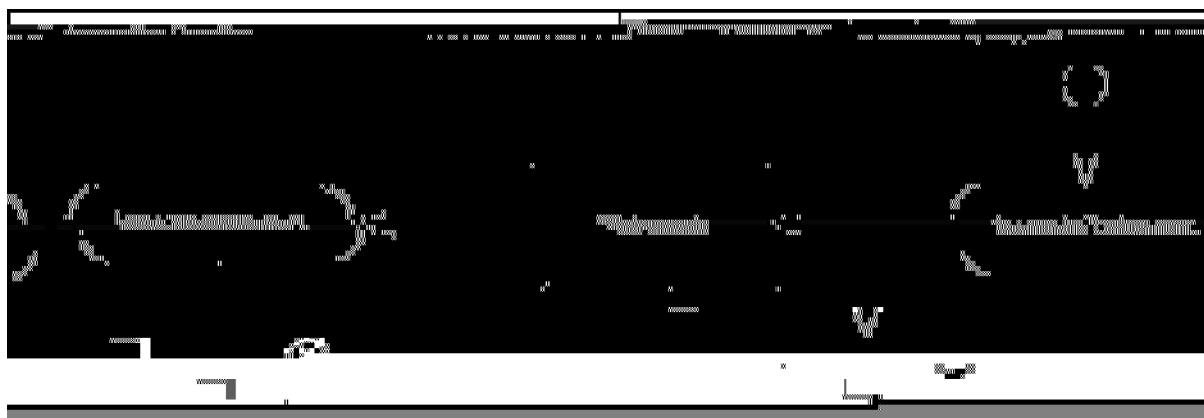
*Figure 31. Payment Notification Process*



## 2.12 Report State of Accounts

A [Statement](#) of account may be used to notify the Accounting Customer of the status of the billing.

*Figure 32. Statement Process*

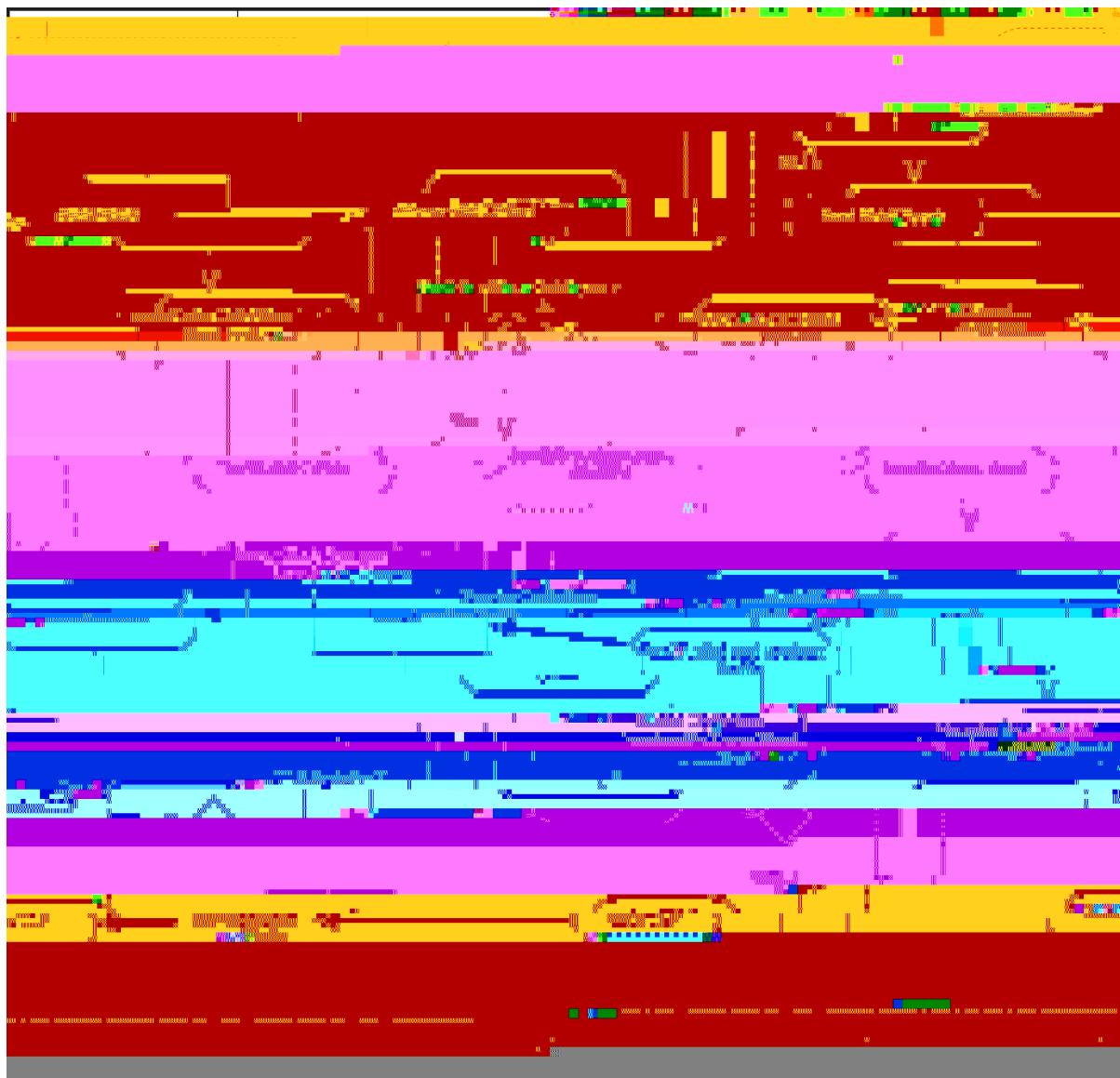




- Document additional information that can be used in the event analysis.



Figure 35. Create Joint Business Plan



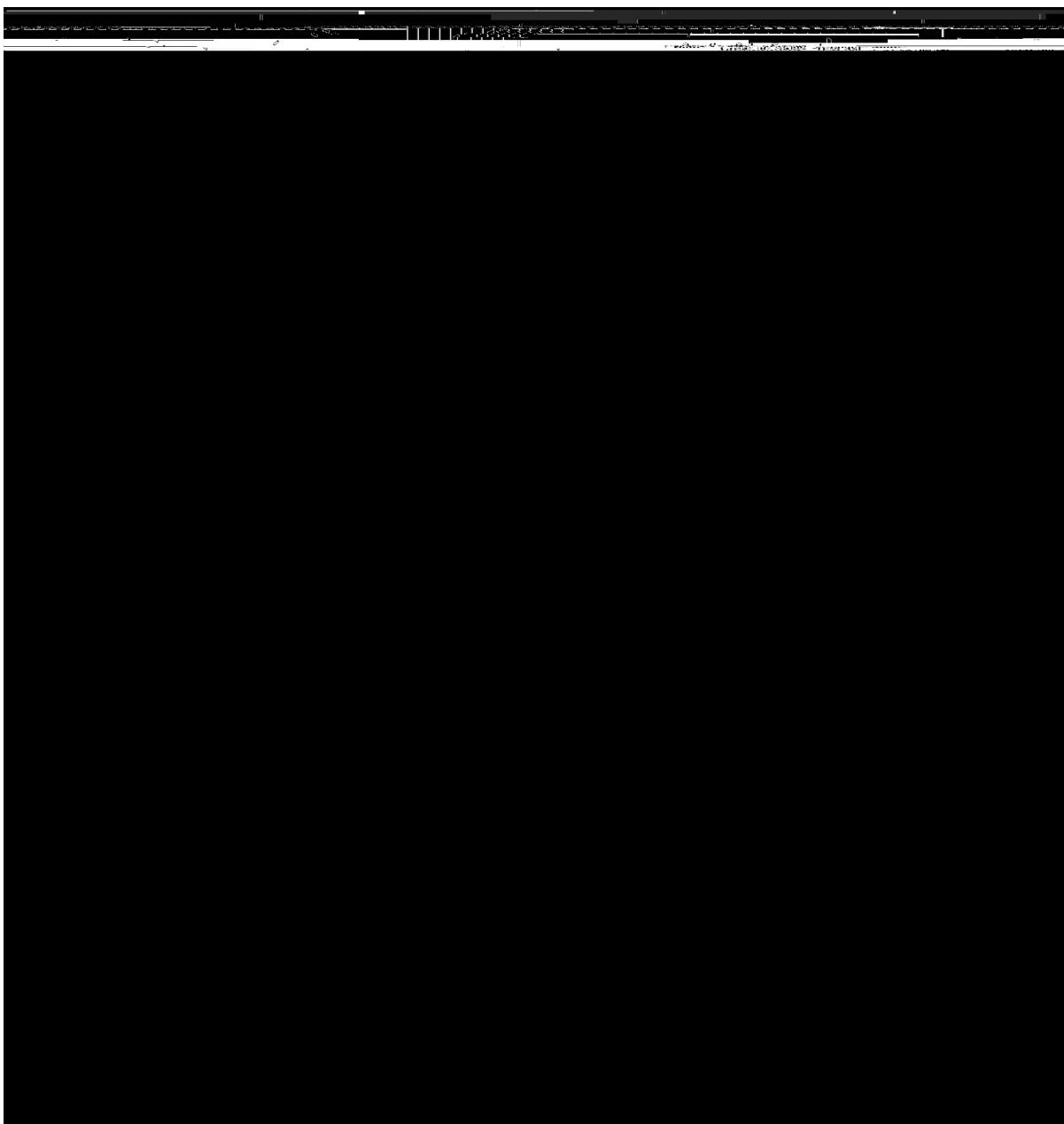
## 2.13.2 Sales Forecast Generation and Exception Handling

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

In CPFR Step 3, the Sales [Forecast](#) is generated. Following Option A, Conventional Order Management, from the CPFR implementation scenario (Table 3), the responsible partner for the generation of Sales Forecast is the Seller. Having Event Calendar information and the Delivery Plan already in their system, there are two more kinds of information that the Seller needs for an effective Sales Forecast: POS Data and DC Data. As shown in [Figure 36, “CPFR Steps 3, 4, and 5”](#) and [Figure 37, “Create Sales Forecast”](#), both of these pieces of information are sent within a [Product Activity](#) message.



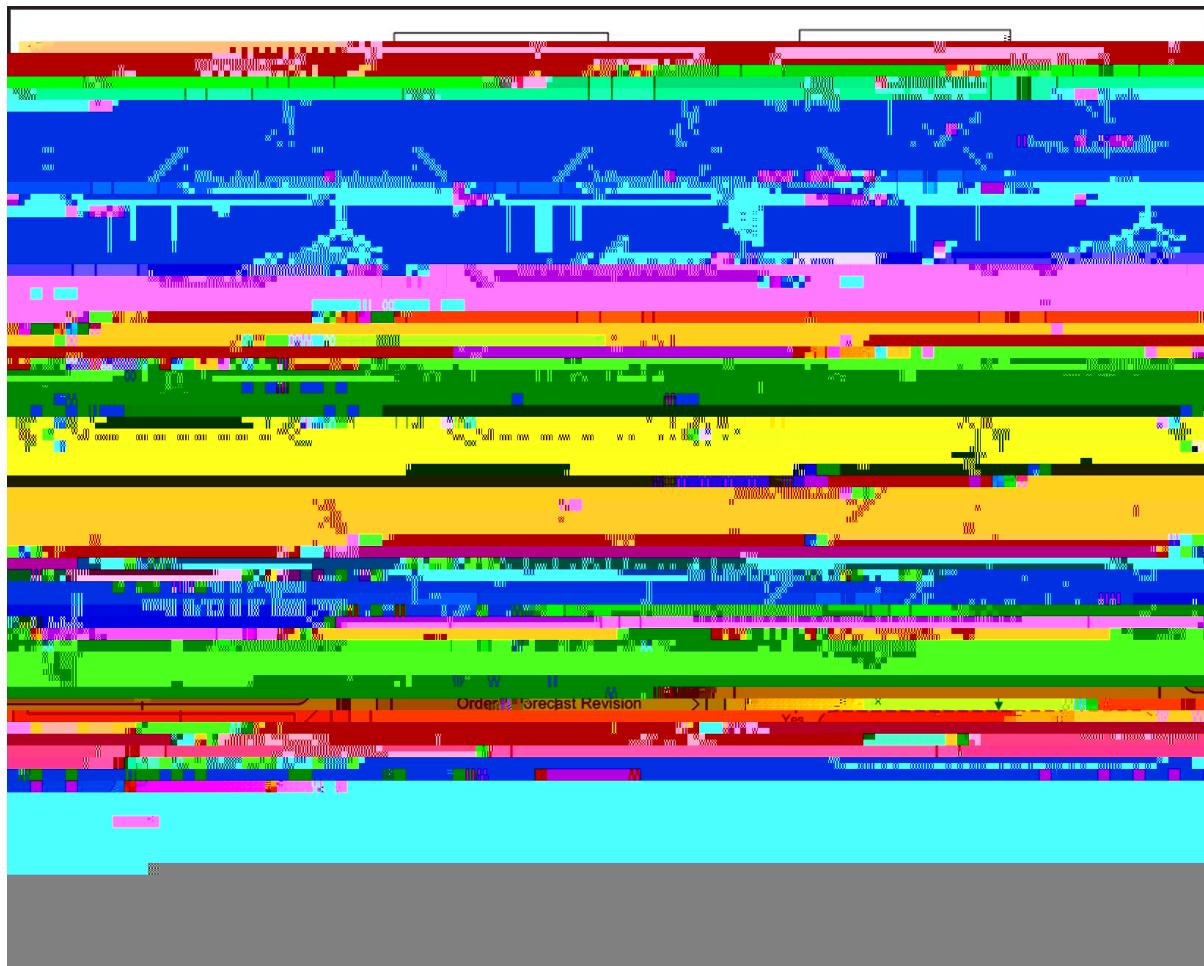
*Figure 37. Create Sales Forecast*



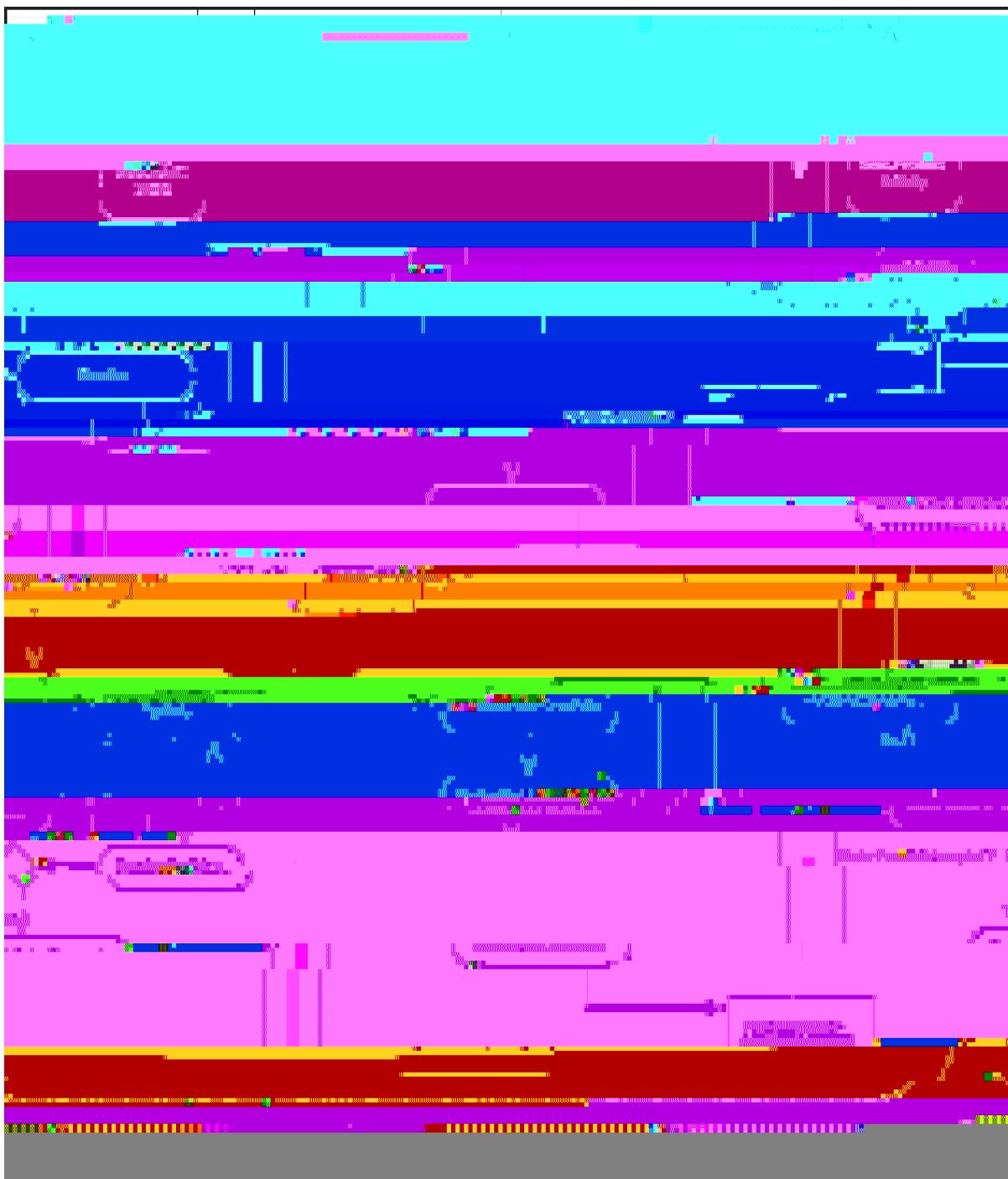
In many cases some time may elapse between Sales Forecast Generation and Order Generation. During this period, both sides observe changes to the conditions. If one of the partners detects an exception



Figure 394 CPFR Steps 6, 7, 8 and 9



After the Seller creates the Order Forecast using the obtained data, it sends the forecast to the Buyer. The Buyer checks the order forecast and sends back a revision document which includes update requests if necessary. The exchange of Order Forecast Revisions continues until there are no further update requests and the Order Forecast is agreed by both sides<sup>4</sup>. Document types used in this process are [Retail Event](#), [Product Activity](#), [Forecast](#), and [Forecast Revision](#).



After the Order Forecast is frozen, the process continues with the exception detection activity (CPFR Step 7). The exception detection process that follows Order Forecast is similar to process described earlier for exception detection following Sales Forecast (see [Section 2.13.2, "Sales Forecast Generation and Exception Handling"](#)). The only difference between the Order Forecast and Sales Forecast exceptions is the content of the exceptions.

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

*Figure 41. Identifying and Resolving Exceptions for Order Forecast*

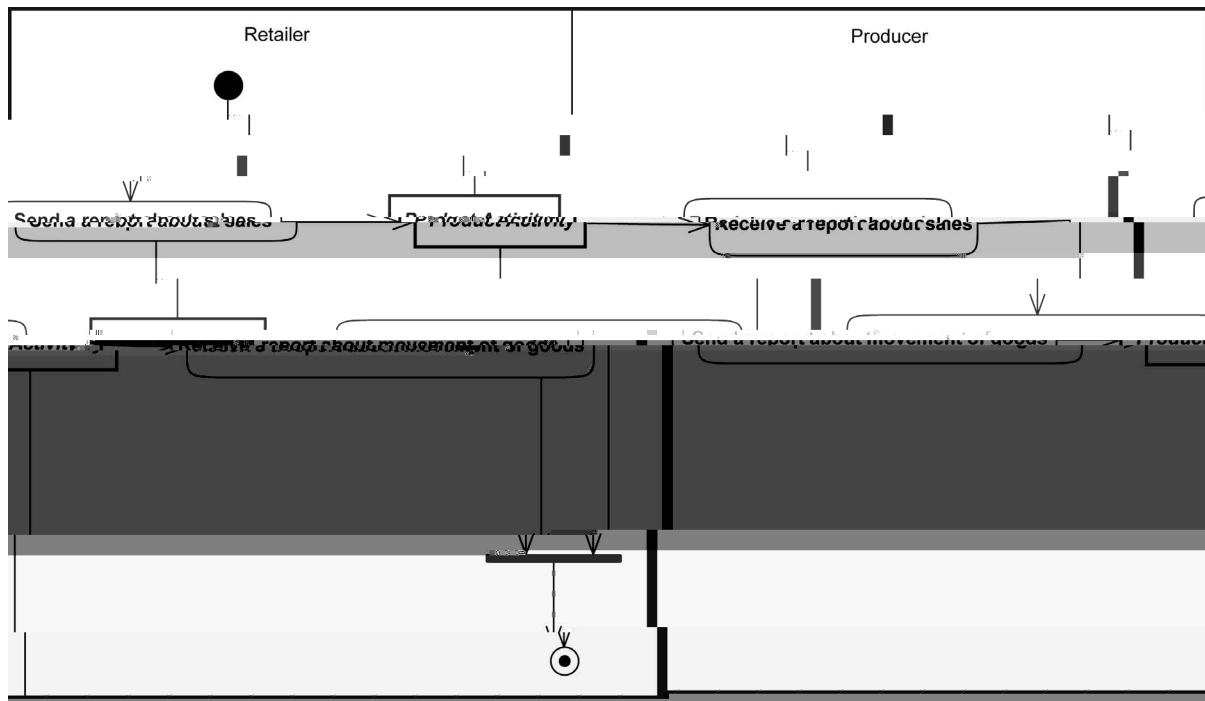
From the technical point of view, the exception monitoring and its resolution are exactly same as in the case of Order Forecast Exception Handling and Sales Forecast Exception Handling. The difference is in the content of the exceptions. The actual events and orders are compared to the Forecasted Sales and Forecasted sames. When there is a situation violating the normal exception criteria, one of the sides might generate an exception notification. Besides comparison of forecasts, other information gathered

#### **2.14.1.1 Initial Stocking of the Area by Producer**

### 2.14.1.2 Report of Sales and Inventory Movement

The sales and inventory movement information is transferred from the retailer to the producer using [Product Activity](#).

*Figure 44. Report of Sales and Inventory Movement*



### **2.14.1.3 Permanent Replenishment**

Based on sales and inventory movement, the producer periodically makes new delivery of goods accompanied by a [Despatch Advice](#). If the delivery contains an item not previously stocked, an updated [Catalogue](#)

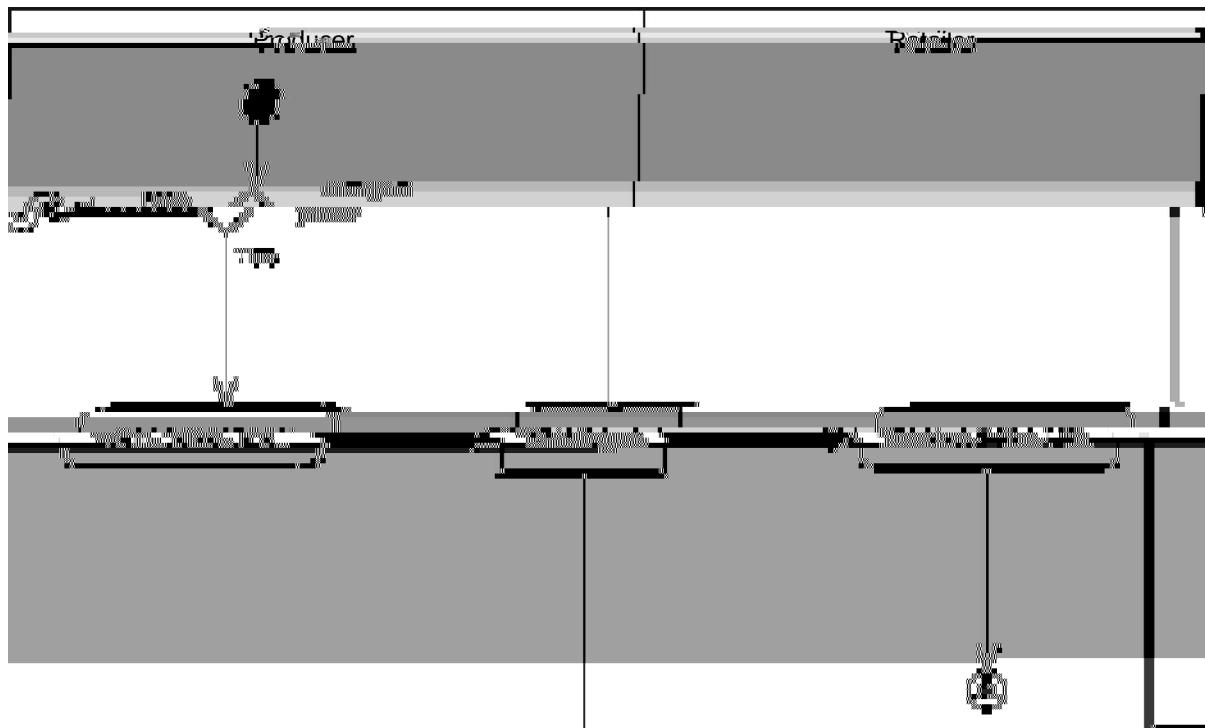




### 2.14.1.6 Price Adjustments

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

*Figure 48. Price Adjustments*







The replenishment process uses the same documents in the same order as the Initial Stocking process, so the duplicate diagram is omitted here; see [Figure 50, “Initial Stocking of the Area by Retailer”](#). It must be remembered, however, that the two processes are taking place at different points in time, so their pre and post conditions will be different.

## **2.14.2.6 Synchronizing of Stock Information**

Information about the actual stock is synchronised periodically (for example, every one to three months) using [Inventory Report](#). This is combined at least once a month.

ventorereepors

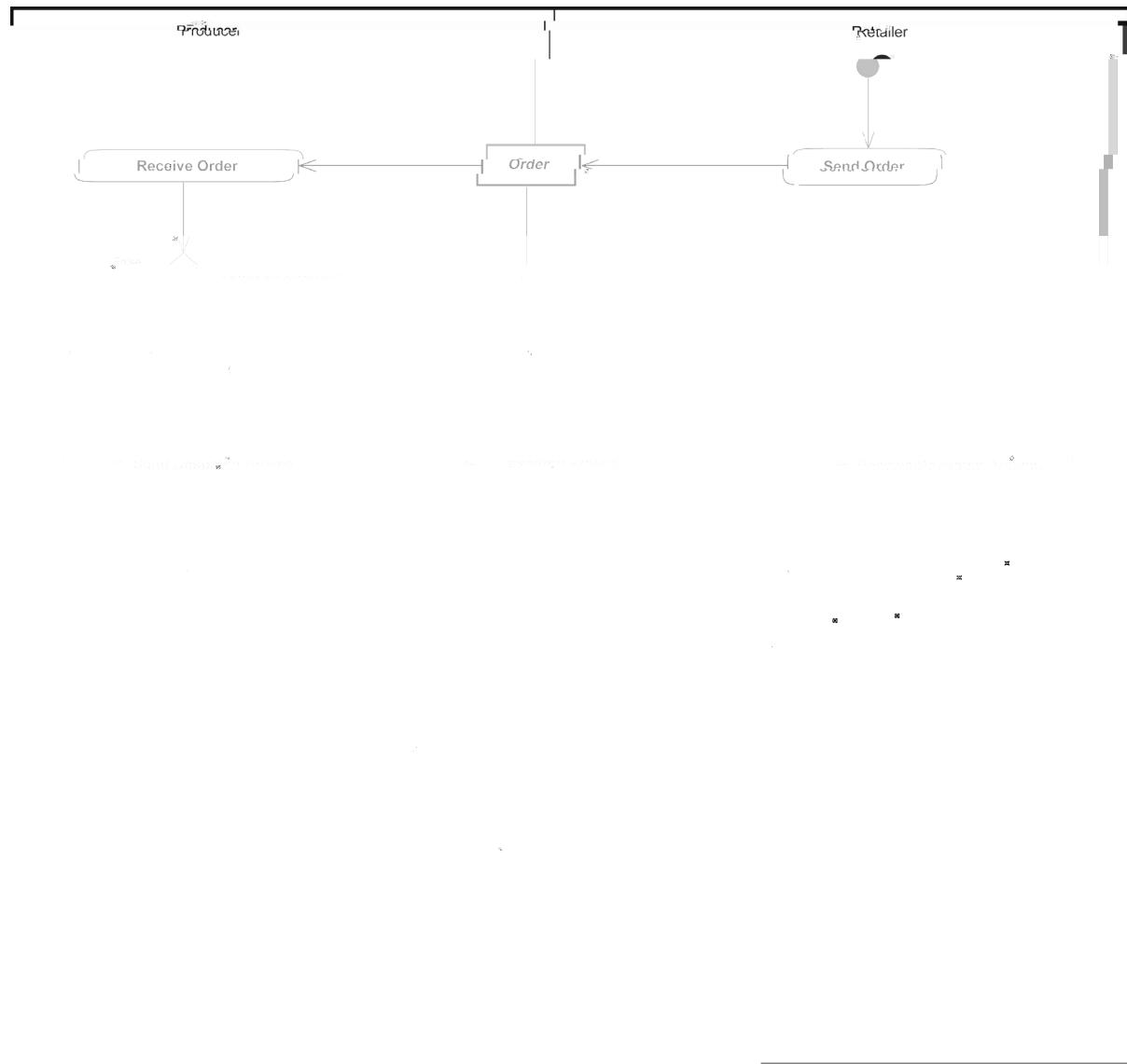
lo

ducer periodically sends information about the availability of items so that the retailer can form the best

### 2.14.3.3 Initial Stocking of the Area by Producer and Retailer

At the beginning of the business cooperation—or perhaps at the beginning of a season, if seasonal NOS (never out of stock) products are the focus—the retailer orders its base stock and the products are delivered. Note that the producer is allowed to propose changes to the order (compare this figure with Figure 50, “Initial Stocking of the Area by Retailer”). Document types used in this process include Order, Order Change, Despatch Advice, and Receipt Advice.

Figure 57. Initial Stocking of the Area by Producer and Retailer



### 2.14.3.4 Periodic Replenishment

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

The replenishment process uses the same documents in the same order as the Initial Stocking process, so the duplicate diagram is omitted here; see [Section 2.14.3.3, “Initial Stocking of the Area by Producer and Retailer”](#).

# Sales and inventory movement

The process for sales and  
Inventory Movement

2.14.3.5 Report of Sales and Inventory Movement

An invoice is issued to the customer.

The process for change order (see Figure 4).





## 2.15.4 Waybill

A [Waybill](#) is a transport document issued by the party who undertakes to provide transportation services, or undertakes to arrange for their provision, to the party who gives instructions for the transportation services (shipper, consignor, etc.). It states the instructions for the beneficiary and may contain 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 (endorsement). It may be issued as a cargo receipt and is not required to be surrendered at the destination in order to pick up the cargo. This may simplify the documentation procedures between a Transport Service Buyer and a Transport Service Provider, but using this document in combination with international payments (e.g., documentary credits) is not advisable.

A freight forwarder may decide to issue a waybill to communicate consignment, transport, and conveyance information to third parties, be they shippers, subcontractors, transport operators, or authorities.

## 2.16 Freight Status Reporting

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

A [Transportation Status](#) document is provided either through a [Transportation Status Request](#)



## **2.18 Intermodal Freight Management**

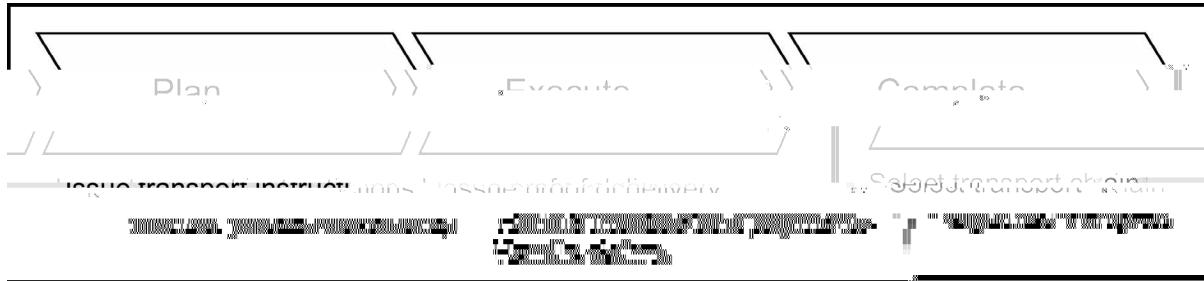
Intermodal transport implies the use of a combination of transport modes. Any support for the management of such chains has to support the modal change of cargo flows from one mode to another in order to create seamless sequences of transportation legs. Quite oftisrgo pndn leg aref caoryroad, b t sefshipping,failandnwsil beinge usd haspndn legs.

I  
t

transport infrastructure from Transportation Network Managers using the [Transport Progress Status](#) document.

- *Completion:* This stage facilitates the issuing of proofs of delivery, claims, and invoices between Transport Service Providers and Transport Users.

*Figure 61. The Generic Freight Management Process*



These three stages are detailed in the following diagram, which shows the part played in the Intermodal Freight Management process by the UBL document types [Transport Service Description](#), [Transport Service Description Request](#), [Transport Execution Plan](#), [Transport Execution Plan30\(vice Descr\)-15\(option Request\)JTJ](#), [Transport Progress Status](#), [Transport Progress Status](#).

*Figure 62. The Intermodal Freight Management Process*



## **2.18.1 Announcing Intermodal Transport Services**

The **Transport Service Description** is used to publish information about a transport service. A **Transport Service Description Request** is used to request such information. A transport service can be the physical transport of cargo between an origin and a destination, and it can also refer to other transport-related services such as terminal services, warehousing services, handling services, or document handling services such as ter





## **2.18.4 Reporting Transport Means Progress Status**

The

## 2.19 Party Roles

In the UBL supply chain processes, two main actors, Customer and Supplier, represent the key organizations or people involved in the processes. Each of these actors may play various roles. Some processes may also involve supplementary roles that may be provided by different parties.

The actual role undertaken is dependent on the context of use. For example, the Despatch Party and Delivery Party as applied to the Procurement process may differ in the Transportation process. In the Transportation Process, two of the main roles are the Transport User and the Transport Service Provider. The Transport User is the role responsible for purchasing a transport service, while the Transport Service Provider is the role responsible for selling and executing a transport service. Both the Customer and the Supplier may be responsible for purchasing and following up the transport of goods, hence both these actors may undertake the Transport User role. In other words, the role of a specific actor depends on the specific circumstances.

The following table contains a description of the typical roles for the actor known as Party. Note that

*Table 1. Party Roles*

Rty0hat haO1t0.253icipates in pre-order Td[(TheTj0 gion)Tj-- that hao

Actor	Role	Description	Example	Synonyms	Sends	Receives
			The people0hat haO1t0.253icipates in pre-order Td[(TheTj0 gion)Tj-- that hao	tion		



---

---

---

---

---

---

---

---





---

---

---

---

Submitter role	Seller
Receiver role	Contracting Party
	<a href="#">xsd/main/doc/UBL-CatalogueItemSpecificationUpdate-2.1.xsd</a>

Receiver role	Issuer, Importer
Normative schema	<a href="#">xsd/maindoc/UBL-CertificateOfOrigin-2.1.xsd</a>
Runtime schema	<a href="#">xsdrt/maindoc/UBL-CertificateOfOrigin-2.1.xsd</a>
RELAX NG schema	<a href="#">rnc/versions/UBL-CertificateOfOrigin-2.1.rnc</a>
Document model (ODF)	<a href="#">mod/maindoc/UBL-CertificateOfOrigin-2.1.ods</a>
Document model (Excel)	<a href="#">mod/maindoc/UBL-CertificateOfOrigin-2.1.xls</a>
Document model (UML)	<a href="#">uml/UBL-CertificateOfOrigin-2.1.html</a>
Summary report	<a href="#">mod/summary/reports/UBL-CertificateOfOrigin-2.1.html</a>

### 3.1.12 Contract Award Notice

Description: A document published by a Contracting Party to announce the awarding of a procurement project.









---





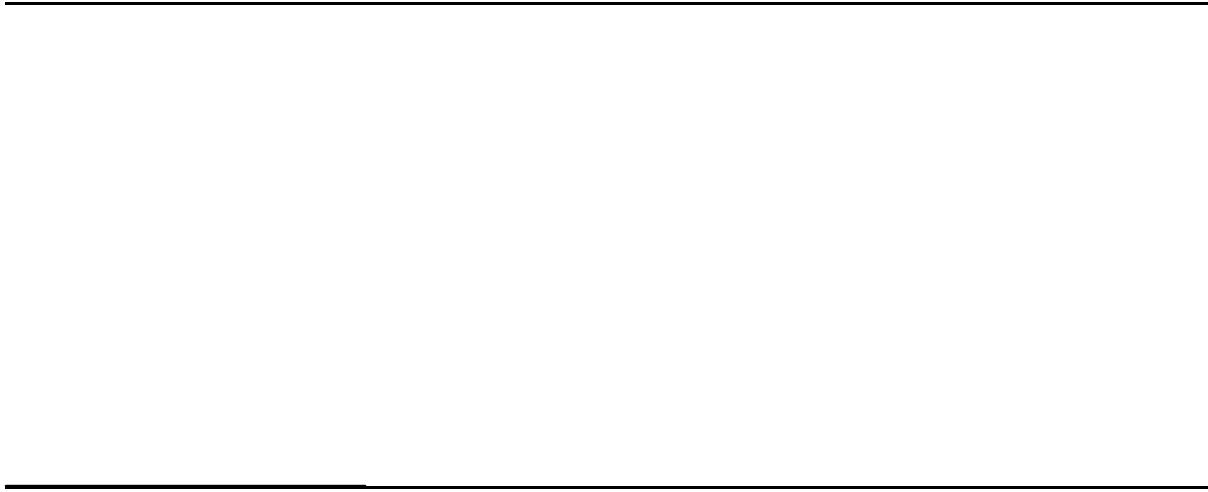
### 3.1.34 Order Change

Description: A document used to specify changes to an existing [Order](#).

Processes involved	Ordering, Fulfilment

Summary report	<a href="#">mod/summary/reports/UBL-OrderResponseSimple-2.1.html</a>
UBL 2.0 example instance	<a href="#">xml/UBL-OrderResponseSimple-2.0-Example.xml</a>
UBL 2.1 example instance	<a href="#">xml/UBL-OrderResponseSimple-2.1-Example.xml</a>





### **3.1.45 Retail Event**

Document model (Excel)	<a href="#">mod/maindoc/UBL-SelfBilledInvoice-2.1.xls</a>
Document model (UML)	<a href="#">uml/UBL-SelfBilledInvoice-2.1.html</a>
Summary report	<a href="#">mod/summary/reports/UBL-SelfBilledInvoice-2.1.html</a>

### **3.1.48 Statement**

Description: A document used to report the status of orders, billing, and payment. This document is a statement of account, not a summary invoice.





Processes involved	<a href="#">Intermodal Freight Management</a>
Submitter role	Transport User
Receiver role	Transport Service Provider
Normative schema	<a href="#">xsd/main/doc/UBL-TransportExecutionPlanRequest-2.1.xsd</a>
Runtime schema	<a href="#">xsdrt/main/doc/UBL-TransportExecutionPlanRequest-2.1.xsd</a>
RELAX NG schema	<a href="#">rnc/versions/UBL-TransportExecutionPlanRequest-2.1.rnc</a>
Document model (ODF)	<a href="#">mod/main/doc/UBL-TransportExecutionPlanRequest-2.1.ods</a>
Document model (Excel)	<a href="#">mod/main/doc/UBL-TransportExecutionPlanRequest-2.1.xls</a>
Document model (UML)	<a href="#">uml/UBL-TransportExecutionPlanRequest-2.1.html</a>
Summary report	<a href="#">mod/summary/reports/UBL-TransportExecutionPlanRequest-2.1.html</a>
UBL 2.1 example instance	<a href="#">xml/UBL-TransportExecutionPlanRequest-2.1-Example.xml</a>

### 3.1.57 Transport Progress Status

### **3.1.59 Transport Service Description**

Description: A document sent by a transport service provider to announce the availability of a transport service.


---

---

## **CommonAggregateComponents**

[xsd/common/UBL-CommonAggregateComponents-2.1.xsd](#)

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

For a discussion of the terms Basic Business Information Entity and Aggregate Business Information Entity, see [Section C.3, “Business Information Entities”](#).

### **3.2.2 Reusable Data Type Schemas**

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

[xsd/common/CCTS\\_CCT\\_SchemaModule-2.1.xsd](#)

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

#### **UnqualifiedDataTypes**

[xsd/common/UBL-UnqualifiedDataTypes-2.1.xsd](#)

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

#### **QualifiedDataTypes**

[xsd/common/UBL-QualifiedDataTypes-2.1.xsd](#)

[\[CCTS\]](#) permits the definition of Qualified Datatypes as derivations from CCTS-specified Unqualified Datatypes. In UBL 2.1, all data type qualifications are expressed in the [\[CVA\]](#) file cva/UBL-DefaultDTQ-2.1.cva. The UBL-QualifiedDataTypes-2.1.xsd file in the UBL 2.1 release is included among the schema modules imported by the, all dat41 Tc -0.571 Tw - componesi]TJ0 g(.)Tj0.231. the rel318e h Dat(d1.xs[12.ludeype)requirshould not btheBusines

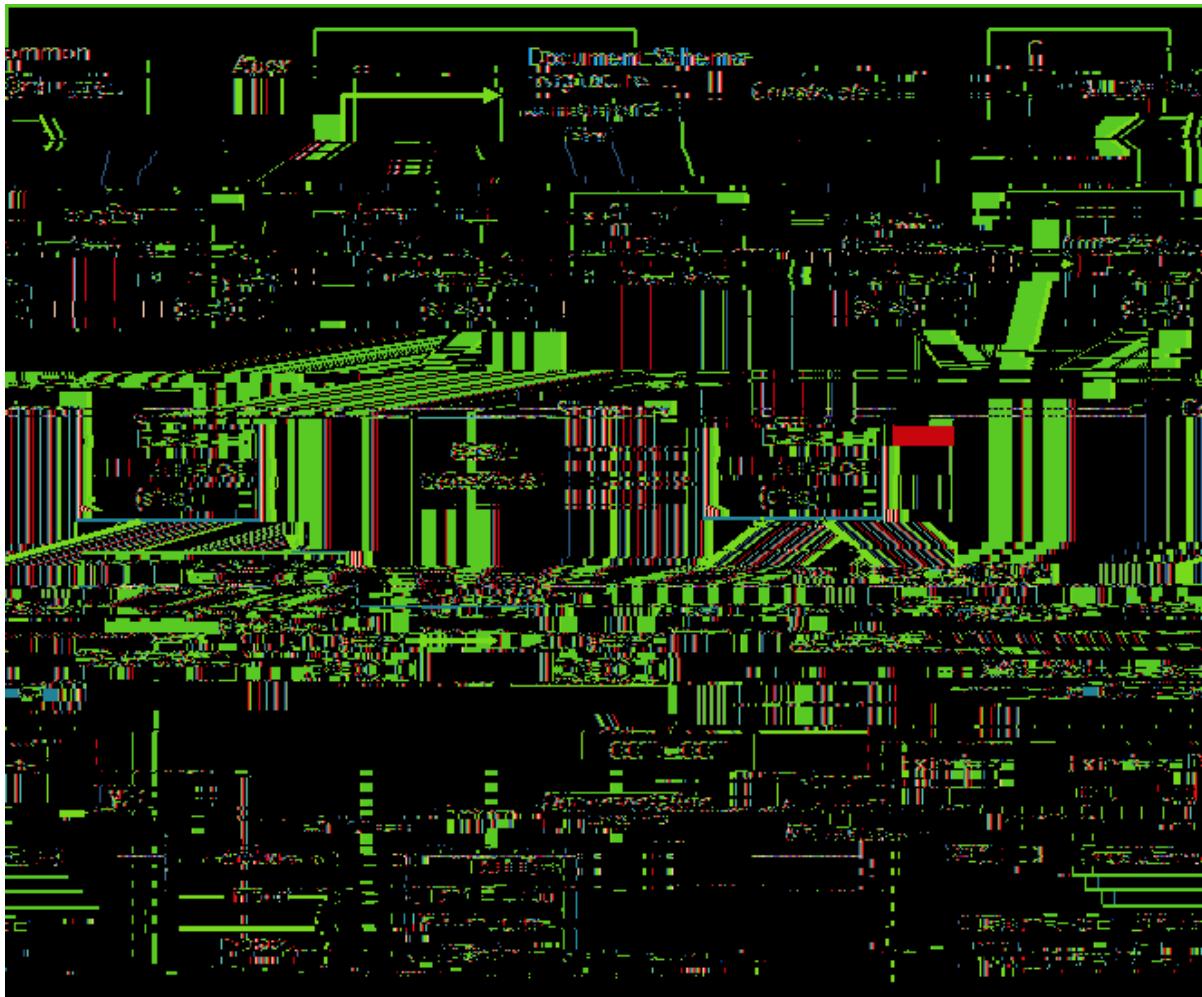




### 3.3 Schema Dependencies

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

*Figure 67. UBL Schema Dependencies*



## Note

The UBL schemas are delivered supporting the UBL standardized extension for digital signatures, defining the content of the extension to be a single element either in or out of the UBL signature extension namespace. As shown on the bottom right in this diagram, a set of UBL schemas supporting a different user-customized extension can be created by replacing the delivered ExtensionContentDataType schema fragment with one also importing the required custom schema fragments that TD[e] the custom content. For more regarding the signature extension, see [Section 5.3, “UBL Extension for Enveloped XML Digital Signatures”](#).

The relationship of the UBL schemas to the UBL data model is illustrated in [Figure C.1, “UBL Data Model Realization”](#).

### **3.4 Extension Validation Methodology**

The UBL extension mechanism allows the addition of other extensions in the same instance. The UBL Digital Signature extension described in [Section 5, “UBL Digital Signatures”](#) is built into the UBL 2.1 distribution and validates transparently.

Users wishing to validate other extensions found in the instance simply revise the UBL-ExtensionContentDataType-2.1.xsd schema fragment. An <xsd:import> directive is added to incorporate the schema constraints of the apex of another extension to be validated in the

---

## 4 Additional Document Constraints

In addition to the UBL 2.1 document constraints formally expressed by the schemas in [Section 3, “UBL 2.1 Schemas”](#), UBL mandates several other rules governing conformant UBL 2.1 instances that cannot

## 4.3 Empty Elements

---

## 5 UBL Digital Signatures

This section provides the  
various digital signature

### 5.1 Introduction (Non-Normative)



- **Enveloped.** The signature applies to the XML content that contains `<ds:Signature>` as an element. Implementations of enveloped signature(s) must take care not to include the signature in the calculation of the signature value.

UBL defines two profiles for signing a UBL document: enveloped and detached.

### 5.1.1.3 XAdES

A compliant implementation of XAdES guarantees wide acceptance in implementing legal regulations, such as European Commission Directive [[1999/93/EC](#)

#### **5.1.1.4 Requirements for Digital Signatures in UBL**

The main requirements to be addressed when choosing a specific signature profile can be divided into the following categories:

- **Legal requirements.** In some countries a digital signature is required on electronic invoices. It can also be compulsory in electronic procurement, especially in a cross-border context, to have a digital signature on the key document exchanged, e.g., a response to a request for tender. Another important legal requirement is long-term document preservation, for a storage period that in general is specific to each country and can span many years. The requirement to guarantee the integrity and authenticity of all fiscally relevant archijen, fnn electroity





## **Note**

The XAdES specification supports an alternative countersignature approach where a



technique without associated UBL artefacts, but an example instance showing detached signatures is included in this package; see [Section 5.4, “Digital Signature Examples”](#).

### 5.3.3 Validation

The `UBL-ExtensionContentDataType-2.1.xsd` module links UBL validation to all needed extensions by importing the apex schema fragment of each extension vocabulary. The distribution version of this module supports IETF/W3C XML digital signatures by declaring that the

### **5.3.4.3 The Signature Apex**

The mandatory `<ext :ExtensionContent>` element contains the UBL signature scaffolding. The apex element of the UBL signature information is `<sig:UBLDocumentSignatures>`.

### **BT0.24 The Signature Information**

Each `<sac:SignatureInformation>` aggregate is used to contain the information related to a single IETF/W3C digital signature. Every signature added to the extension is isolated under a separate `<sac:SignatureInformation>` aggregate element containing the signature and its supporting in-





The [xml/UBL-Invoice-2.0-Detached.xml](#) sample document illustrates the placement of a detached digital signature outside of the UBL file. A `<cac:Signature>` element makes reference to the external signature.

The [xml/UBL-Invoice-2.0-Detached-Signature.xml](#) instance is an example of a bona fide verifiable digital signature of the [xml/UBL-Invoice-2.0-Detached.xml](#) instance.

---

# **6 Conformance**

## **6.1 Document and Schema Conformance**

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

---

## **Appendix A Release Notes (Non-Normative)**

### **A.1 Availability**



rected in the course of preparing UBL 2.1. These corrections have no effect on processing, validation, or instance generation, but they are listed in [Section B.3.4, “Schema changes from UBL 2.0 XML to UBL 2.1 XML”](#) for completeness in documentation.

# Appendix B Revision

Since its first release as an OASIS Standard in 5004, UBL has experienced one major and now one minor version upgrade.

This appendix provides a description of the evolution of UBL.

## B.1 UBL 1.0

Though apparently limited in scope, the eight document types provided in UBL 1.0 (5004) are applicable to a very large number of real-world use cases and have been widely deployed. These original 1.0 document types, later updated in UBL 2.0 and continued here in 5.1, are [Order](#), [Order Response](#), [Order Response Simple](#), [Order Change](#), [Order Cancellation](#), [Despatch Advice](#), [Receipt Advice](#), and







*Table B.1. Changes to Library Elements, UBL 2.0 to UBL 2.1*





--	--	--

---

--	--	--

---

--	--	--



--	--	--

--	--	--

--	--	--

--	--	--

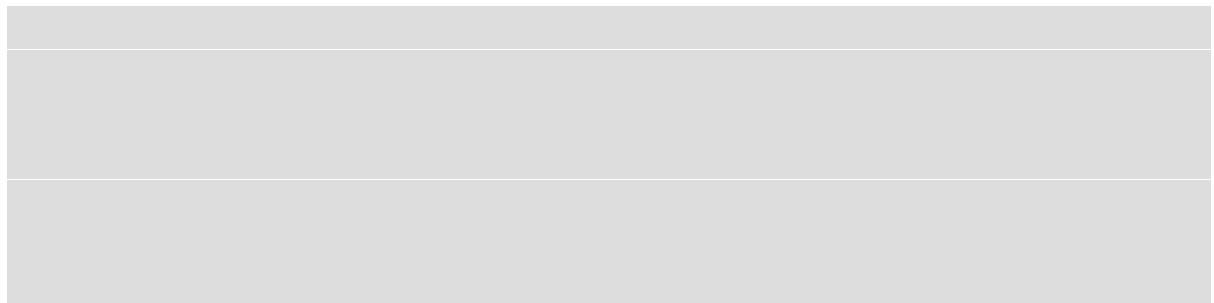
--	--	--

--	--	--

--	--	--

--	--	--









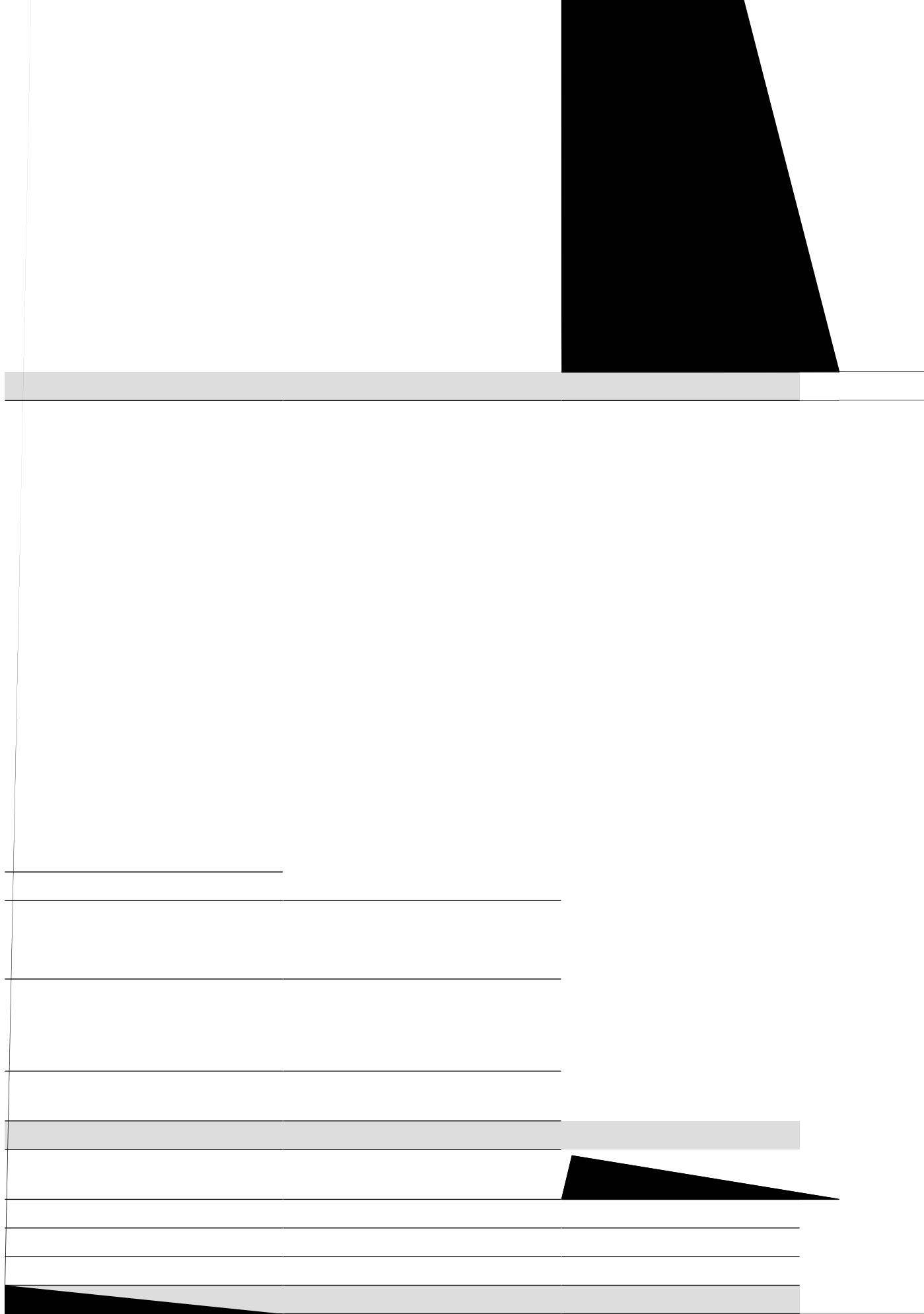
<b>Aggregate BIE</b>	<b>Basic or Association BIE</b>	<b>Changes for UBL 2.1</b>
NotificationRequirement		Added
OnAccountPayment		Added



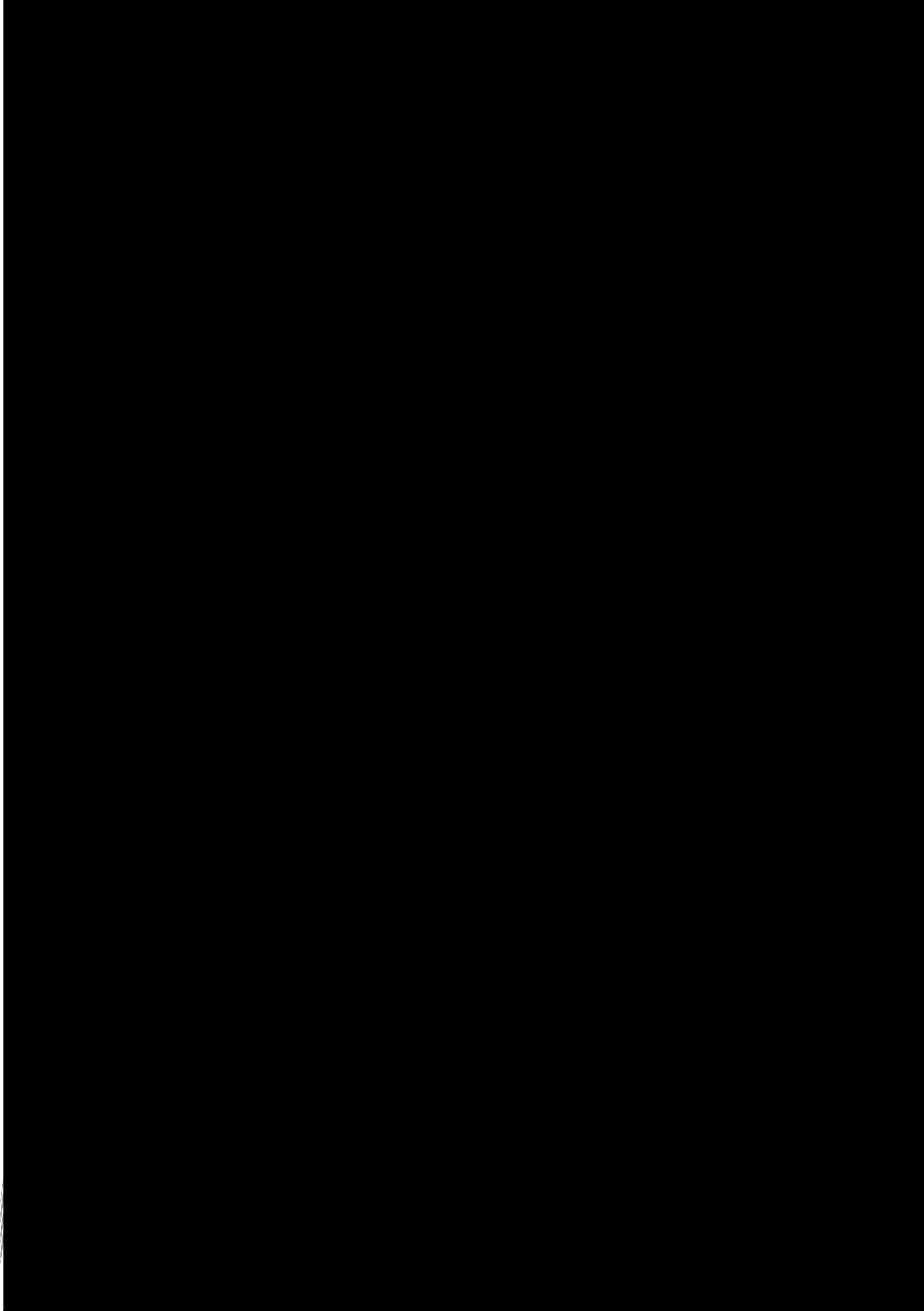




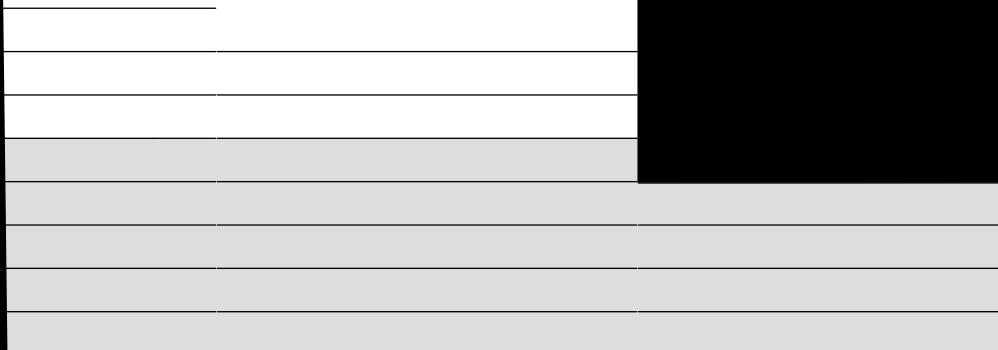










*Table B.2. Changes to Document Elements, UBL 2.0 to UBL 2.1*








<b>Aggregate BIE</b>	<b>Basic or Association BIE</b>	<b>Changes for UBL 2.1</b>
TendererQualificationResponse		Added
TradeItemLocationProfile		Added
TransportExecutionPlan		Added
TransportExecutionPlanRequest		Added
TransportProgressStatus		Added
TransportProgressStatusRequest		Added
TransportServiceDescription		Added
TransportServiceDescriptionRequest		Added
TransportationStatus		
	ProfileExecutionID	Added
	TransportationStatusTypeCode	Added
	TransportExecutionStatusCode	Added
	Consignment	Change cardinality from 1 to 0..n
	TransportEvent	Change cardinality from 1..n to 0..n
	SenderParty	Added
	ReceiverParty	Added
	TransportationStatusRequestDocumentReference	Added
	TransportExecutionPlanDocumentReference	Added
	UpdatedPickupTransportEvent	Added
	UpdatedDeliveryTransportEvent	Added
	StatusLocation	Added
	StatusPeriod	Added
TransportationStatusRequest		Added
UnawardedNotification		Added
UtilityStatement		Added
Waybill		
	ProfileExecutionID	Added



---

## **Appendix C The UBL 2.1 Data Model (Non-Normative)**

Following the principles of the ebXML Core Components Technical Specification [[CCTS](#)], the UBL data model is based on a library of reusable information items known as Business Information Entities (BIEs). Each business document defined by UBL is created by assembling items appropriate to that document type from the UBL BIE library. Further detail regarding BIEs is provided in [Section C.3, “Business Information Entities”](#).

### **C.1 UBL 2.1 Schema Generation**



are members as *elements* of the UBL common aggregate components namespace, denoted in UBL schemas by the `cac:` prefix.

**UBL document ABIEs (Aggregate Business Information Entities)** are the root nodes and top-level structures of UBL documents. Children of document ABIEs may be BBIEs or ASBIEs, never ABIEs. All UBL document ABIEs (and only UBL document ABIEs) are defined within individual namespaces specific to each document as both elements and types.

**UBL library ABIEs** (that is, all ABIEs except document ABIEs) have a structural shape but are not concrete document structures; rather, they are abstract structures or templates for ASBIEs, thus allowing the same structure to be reused in multiple roles. Children of library ABIEs in the data structure can be BBIEs or ASBIEs, never ABIEs. All library ABIEs must be realized as ASBIEs in order to actually exist

One consequence of this approach is that the list of global elements that begins the CAC module contains elements that are in fact never used under those names in UBL 2.1. For example, the element `ActivityDataLine`

[mod/common/UBL-CommonLibrary-2.1.xls](#)

Period will be found at line 1418 and seen to contain seven possible BBIE children, all of them optional; and the ASBIE InvoicePeriod in Invoice therefore has this structure, too. From this one could conclude that instantiations of the Period structure (there are more than 50 of them in UBL 2.1) need not contain any of the seven optional BBIE elements specified at lines 1419–1425, and indeed the cor-



## **C.5 Summary Reports**

While the document model spreadsheets described above function as a basic form of documentation,

---

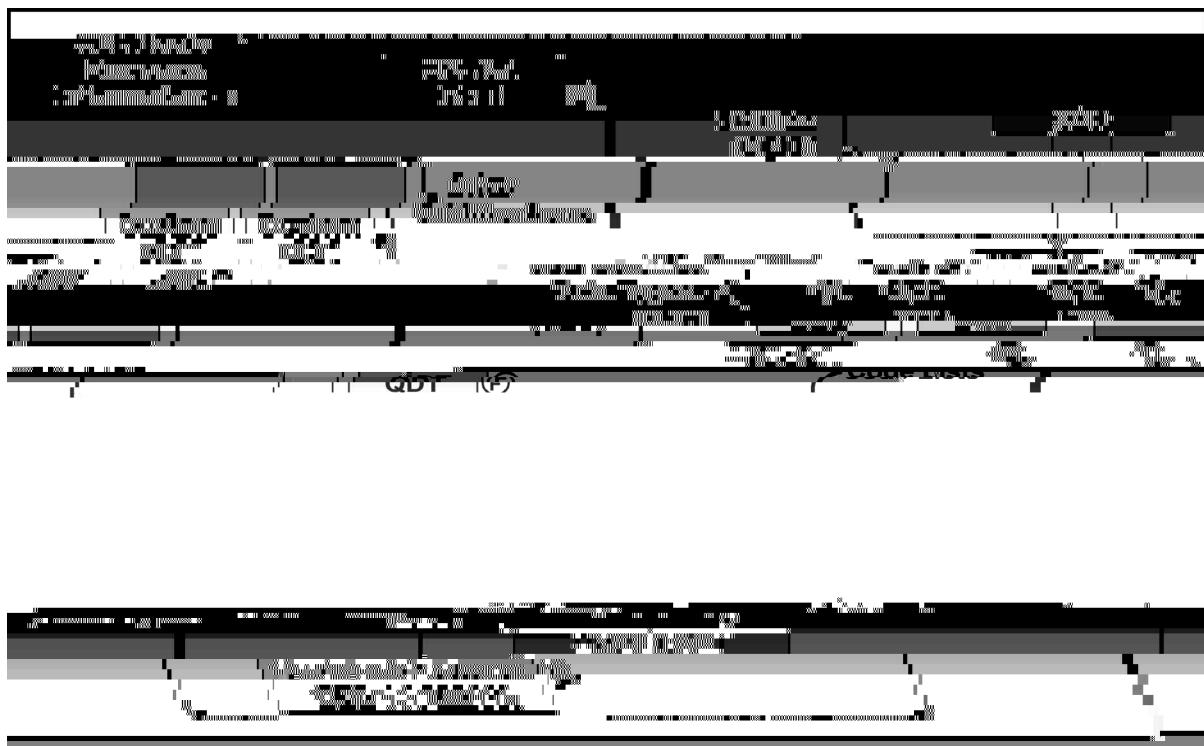
## **Appendix D Data Type Qualifications in UBL (Non-Normative)**

All UBL data types ultimately derive either from the UN/CEFACT Core Components Technical Specific-

*Table D.2. UBL UnqW nBT/T1\_I4595.276 841. Data .2.*

<b>UBL Unqualified</b>	<b>Definition</b>	<b>Attributes</b>

*Figure D.1. Data Type Qualification in UBL 2.1*



In UBL 2.1, the schema library of common basic components (basic information entities or BBIEs, **(A)**

---

## **Appendix E UBL 2.1 Code Lists and Two-phase Validation (Non-Normative)**

### **E.1 Introduction**

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

`test.bat` (Windows)

or

`sh test.sh` (Linux)

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

*Figure E.1. Validation test output*



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

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

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

## E.3 Discussion of the Default Validation Test



## E.4 Customizing the Default XSLT File

The validation framework provided in the val

against the values specified in the

cl/gc/special-purpose/ThresholdValueComparisonCode-2.1.gc  
cl/gc/special-purpose/TimeFrequencyCode-2.1.gc  
cl/gc/special-purpose/TransportationStatusCode-2.1.gc  
cl/gc/special-purpose/CoordinateSystemCode-2.1.gc  
cl/gc/special-purpose/EnvironmentalEmissionCode-2.1.gc  
cl/gc/special-purpose/NotificationTypeCode-2.1.gc  
cl/gc/special-purpose/TransportationStatusTypeCode-2.1.gc  
cl/gc/special-purpose/TransportServiceCode-2.1.gc  
cl/gc/special-purpose/DocumentStatusReasonCode-2.1.gc  
cl/gc/special-purpose/AccountTypeCode-2.1.gc  
cl/gc/special-purpose/AddressTypeCode-2.1.gc  
cl/gc/special-purpose/CargoTypeCode-2.1.gc  
cl/gc/special-purpose/CommodityIdentificationCode-2.1.gc  
cl/gc/special-purpose/DocumentTypeCode-2.1.gc  
cl/gc/special-purpose/FullnessIndicationCode-2.1.gc  
cl/gc/special-purpose/HumanSexesCode-2.1.gc  
cl/gc/special-purpose/HandlingCode-2.1.gc  
cl/gc/special-purpose/HazardousCategoryCode-2.1.gc  
cl/gc/special-purpose/PartyRoleCode-2.1.gc  
cl/gc/special-purpose/ItemClassificationCode-2.1.gc  
cl/gc/special-purpose/LocationTypeCode-2.1.gc  
cl/gc/special-purpose/PackageLevelCode-2.1.gc

```
<Agency>
  <LongName xml:lang="en">OASIS Universal Business Language</LongName>
  <Identifier>UBL</Identifier>
</Agency>
```

```
<Agency>
  <LongName xml:lang="en">International Organization for Standardization</LongName>
  <Identifier Identifier="http://www.unece.org/trade/untdid/d11a/tred/tred3055.htm"
             >5</Identifier>
</Agency>
```

```
<Agency>
  <LongName xml:lang="en">United Nations Economic Commission for Europe</LongName>
  <Identifier Identifier="http://www.unece.org/trade/untdid/d11a/tred/tred3055.htm"
             >6</Identifier>
</Agency>
```

The numeric “5” and “6” agency identifiers are from the code list “Responsible Agency Code” most recently published at

<http://www.unece.org/trade/untdid/d11a/tred/tred3055.htm>

---

## **Appendix F UBL 2.1 Example Document Instances (Non-Normative)**







The FSV addresses the supporting services meeting the mechanistic needs of Open-edi, focusing on





