

Norsk EDIPRO is today named  
**NorStella** (Foundation for E-Business  
and Trade Procedures)

# transportXML

A Norsk EDIPRO project

## Collaboration models and XML messages for the transport industry

Version 3.0

## Preface

This document presents a recommendation (hereafter referred to as *the Recommendation*) for conducting electronic collaboration within *the transport business domain* based on XML.

XML (eXtensible Markup Language) is a general data description language with widespread usage, in particular where the applications involved are based on internet technology.

The Recommendation has been produced through *the transportXML project* conducted by Norsk EDIPRO (note: Norsk EDIPRO is named NorStella after the 1st of January 2003), in which a significant number of Norwegian transport related companies have participated. The project was started on the 15<sup>th</sup> of May 2001 and was completed in October 2002.

The Recommendation covers *two functional areas* within the transport domain: **Transport Job** and **Track&Trace**. These areas are both broadly defined, and the solutions presented are consequently considered to satisfy requirements for many scenarios related to transport.

The first openly available version of transportXML – version 2.0 – was presented and distributed in April 2002. This initial version mainly covered the Transport Job functional area.

We now present version 3.0 of the specification, which in addition to Transport Job also includes the Track&Trace functional area. We have further made some important modifications and additions in the model and the message type for the Transport Job area.

## Background.

The transport industry is a business through which and within which large volumes of information are channeled involving many different parties. The reason for this is manifold: First the companies while performing their business have to fulfill a lot of documentation and reporting requirements from authorities in various countries – customs declarations, transit regulations, dangerous goods documentation to mention only a few examples. Further a transport chain – in particular an international transport chain - involves many parties, and it is vital that the relevant information flows between these parties in a correct way and at the right time.

Therefore it was only natural that the transport industry was among the first ones to employ automatic information exchange, and the industry can probably display some of the most successful and comprehensive EDI implementations ever achieved. Transport companies have through the 1990s shown a great will to cooperate for the standardization of solutions, both as regards EDIFACT message implementation guidelines and related solutions like bar coding standards.

During the last years of the preceding decade and the first years of the present one we have witnessed a steadily increasing use of internet related technology, in which web based services work together with EDI solutions. In the transport industry this development may be exemplified by applications for the tracing of goods - "track & trace". We feel the time is ripe for agreeing on a common and harmonized way to employ this technology: language, message structures, availability and tools for implementation. This naturally leads us to XML – the eXtensible Markup Language.

Oslo, the 10th of October 2002

Arild Nybakk

Mariann Sundvor

## Contents

<b>1</b>	<b>Introduction.....</b>	<b>6</b>
1.1	The project and the way in which we have been working.....	6
1.2	Working group. ....	6
1.3	Functional areas. ....	7
1.4	Concrete deliveries – "model" og "message".....	8
1.5	Collaboration scenarios (usages). ....	9
1.5.1	Transport Job.....	9
1.5.2	Track&Trace.....	14
1.6	Basic design principles and rules for the model / message types. ....	15
1.6.1	Wide application area per message type: Flexible message structure - profiles.....	15
1.6.2	Representing generalizations/specializations in model and message type.....	16
1.6.3	Data types in model og message type.....	17
1.6.4	Attributes as opposed to elements in an XML message type.....	18
1.6.5	Representing core component types in XML messages. ....	18
1.6.6	Naming convention: Camel case. ....	19
1.6.7	Basic objects and structures – the structure of message model and message type.....	19
<b>2</b>	<b>Transport domain: Domain model and collaboration models.....</b>	<b>21</b>
2.1	Domain model: Information model described using class diagrams.....	22
2.1.1	Class diagram - part: Party, goods classes (Consignment/Goodsltem/Package) and the TransportJob class.....	22
2.1.2	Class diagram - part: TransportLeg.....	23
2.1.3	Class diagram - part: DangerousGoods.....	24
2.1.4	Class diagram - part: Freight – Service – DateAndTimes.....	25
2.1.5	Class diagram - part: CustomsInformation.....	26
2.1.6	Class diagram - part: TrackAndTrace.....	27
2.2	Collaboration models – process models described using activity diagrams.....	28
2.2.1	Collaborations in the Transport Job functional area.....	29
2.2.2	Collaborations in the Track&Trace functional area.....	41
2.3	Domain model – semantic descriptions of classes and attributes.....	44
2.3.1	Generic attribute names in XML message instances:.....	84
2.3.2	Attributes in the root element of an XML message instance:.....	84
2.4	Transport Job – Domain modell: Semantic descriptions of relationships.....	86
2.5	Track & Trace – Domain model: Semantic descriptions of relationships.....	99
2.6	Domain model: Code lists.....	101
	.....	103
<b>3</b>	<b>Message model - TransportJob.....</b>	<b>106</b>
3.1	Message model for TransportJob.....	106
3.1.1	Message model TransportJob – High level view of BasicObject – Structure.....	106
3.1.2	Message model TransportJob - BasicObject.....	107
3.1.4	Message model TransportJob - Structure part 1: Consignment.....	108
	Message model TransportJob - Structure part 2: Transport, Equipment, DangerousGoods.....	109
	Message model TransportJob - Structure part 3: TermsOfDelivery, Party (1).....	110
3.1.6	Message model TransportJob - Structure part 4: Party (2).....	111
3.1.7	Message model TransportJob - Structure part 5: Party (3).....	112
3.1.8	Message model TransportJob - Structure part 6: Offer, Service, CustomsInformation.....	113
3.1.9	Message model TransportJob - Structure part 7: DateAndTimes (1).....	114
3.1.10	Message model TransportJob - Structure part 8: DateAndTimes (2).....	115

3.2	<i>Message model TransportJob – Profile: DomesticNO</i> .....	116
3.2.1	<i>Profile DomesticNO – BasicObject</i> .....	116
3.2.2	<i>Profile DomesticNO – Structure part 1: Consignment</i> .....	117
3.2.3	<i>Profile DomesticNO - Structure part 2: Transport, Equipment, DangerousGoods</i> ..	118
3.2.4	<i>Profile DomesticNO - Structure part 3: TermsOfDelivery, Party(1)</i> .....	119
3.2.5	<i>Profile DomesticNO - Structure part 4: Party (2)</i> .....	120
	<i>Profile DomesticNO - Structure part 5: Party (3)</i> .....	121
3.2.7	<i>Profile DomesticNO - Structure part 6: Service</i> .....	122
3.2.8	<i>Profile DomesticNO - Structure part 7: DateAndTimes (1)</i> .....	123
3.2.9	<i>Profile DomesticNO - Structure part 8: DateAndTimes (2)</i> .....	124
3.2.10	<i>The DomesticNOMin profile: A recommendation for the use of a minimal set of mandatory information objects for domestic transport</i> .....	125
3.3	<i>Message model TransportJob – Profile: TransportPortal</i> .....	125
3.3.1	<i>Profile TransportPortal - BasicObject</i> .....	126
3.3.2	<i>Profile TransportPortal - Structure part 1: Offer</i> .....	127
3.3.3	<i>Profile TransportPortal – Structure part 2: Party</i> .....	128
3.3.4	<i>Profile TransportPortal – Structure part 3: DateAndTimes</i> .....	129
<b>4</b>	<b><i>Message model - TrackAndTrace</i>.....</b>	<b>130</b>
4.1	<i>Message model TrackAndTrace : BasicObject</i> .....	130
4.2	<i>Message model TrackAndTrace – Structure part 1: Event</i> .....	131
4.3	<i>Message model TrackAndTrace – Structure part 2: Party</i> .....	132
4.4	<i>Message model TrackAndTrace – Structure part 3: Consignment</i> .....	133
<b>5</b>	<b><i>XML messages</i>.....</b>	<b>134</b>
5.1	<i>Comprehensive sequential run-through with example: TransportJob</i> .....	134
5.2	<i>Comprehensive sequential run-through with example: TrackAndTrace</i> .....	156

# 1 Introduction

## 1.1 The project and the way in which we have been working

The transportXML project - initiated in May 2001 and completed in October 2002 – presents in this document:

- A. *A model for a subset of business areas within transport & logistics being independent of concrete message definitions or message syntax. The model is made using UML (Universal Modelling Language). The subset is identified as the functional areas **Transport Job** and **Track&Trace**.*
- B. *Message models equally expressed in UML for the message types which are necessary to support/implement ecommerce functions within the functional areas listed in paragraph A.*
- C. *Concrete XML message types derived from the UML message models referred to in paragraph B. These message types are defined using XML schemas..*

Through its project *Infrastructure for electronic commerce* Norsk EDIPRO has developed an internationally based description technique for ecommerce solutions. The description of models and solutions in transportXML broadly follows this description technique.

The infrastructure project of Norsk EDIPRO has further presented recommendations as to how descriptions of ecommerce solutions may be made available in open Registries/Repositories. As of today few repositories exist supporting open access to such descriptions. In the transportXML project there has been, however, a clear target to make our model and message descriptions available in Registries/Repositories whenever and wherever appropriate.

## 1.2 Working group.

The Recommendation has been worked out by a working group consisting of the following members:

Company	Participant
Danzas ASG	Ole-Kristian Smaadahl
DFDS Tollpost-Globe	Håkon Stokke Sæther
Entra Data/Take Cargo	Kurt Arve Veum
Systema	Svein Terje Berg
Linjegods	Svein Vikhamar

<i>NorCargo</i>	<i>Atle Thorstensen</i>
<i>Frans Maas</i>	<i>Øyvind Bondkall</i>
<i>ErgoSolutions</i>	<i>Inger Lise Berglund</i>
<i>Norsk EDIPRO</i>	<i>Mariann Sundvor</i>
<i>Skandinavisk Transport System</i>	<i>Arild Nybakk</i>
<i>KSD Software Norway</i>	<i>Jon-Arild Ludvigsen</i>
<i>Posten Logistikk</i>	<i>Dag Sørensen</i>
<i>Gemsys</i>	<i>Morten Hanssen</i>

Arild Nybakk has been the professional chair of the group, whereas Mariann Sundvor has chaired all administrative functions.

The group has met once each month all through the project period, and additionally each participant has performed dedicated tasks in the period between meetings.

### 1.3 Functional areas.

The Recommendation covers two functional areas:

- **Transport Job.** This area includes all model and message applications in which there is a need to convey all (relevant) information related to a goods transport. Such conveyance is envisaged to take place:
  - before a transport is performed,
  - during the carrying out of a transport,
  - upon completion of a transport,
  - any time after the completion of a transport, f.ex. as part of a statistics function.
- **Track&Trace.** This functional area includes solutions for requests, responses and reports on administrative and operational events, including exception reports, related to a goods transport artifact (means of transport, consignment, package, etc). Events may be reported – possibly through several links in the transport chain – independent of any preceding requests. One of the event reports is the *Proof of Delivery*.

In summary **Transport Job solutions** will be used when the collaboration requires that relevant information (and often complete information) *about a goods transport or goods transport artifacts* is conveyed. **Track&Trace**

**solutions** is to be used when the collaboration requires that status or state information be given for the goods transport artifact.

## 1.4 Concrete deliveries – "model" og "message".

The Recommendation contains the following specifications:

- **Collaboration models/domain model.** The Recommendation contains a message independent information model for the transport business domain (we call such an information model a **domain model**) and a set of **collaboration models** employing/containing (subsets of) this information.

The collaboration models describe the processes which are relevant within the selected functional areas, and the domain model includes as a minimum *all information* employed by the defined collaborations.

The model is presented using the modelling language Unified Modelling Language (UML) through:

- *A class diagram:* For practical reasons the class diagram is divided into several parts (f.ex. Party part, TransportLeg part, etc.), but it is nonetheless to be considered as one unified description of the information model of the transport domain.
  - *Semantic descriptions:* A detailed description of the meaning content of each component of the model (classes, attributes and relationships).
  - *Code lists* for attributes where the data content is coded.
  - *Activity diagrams:* A set of activity diagrams describing the *processes* or *collaborations*. Each activity diagram contains the process part of *one* collaboration model.
- **Message models.** For each functional area the Recommendation specifies one message model – equally described in UML – based on the overall domain model. A message model reflects the information to be exchanged between the parties and the structure of this information.
  - **XML message types (XSD schemas and example messages).** For each message model the Recommendation contains an XML message type generated by a "conversion" of the message model. The message types are described using *XSD schemas*. In order to enhance the understanding of the message types the documentation further includes *one complete all encompassing example* of each message type.

We emphasize that *the two primary and important deliveries of the project are the collaboration/domain model and the XML message types*. The message model is a tool – a "missing link" – created in order to ease the generation of XML message types based on the collaboration/domain model. The message model may further be helpful for the understanding of the message types.



We further emphasize that the model is not only meant to *describe* the information content and its structural relationships, but also to *define the meaning of each model element*. The model conveys a *semantic definition of the XML message types*. The usage and meaning content of every construction in the XML message types are defined by the model (let us say that the description of the XML message type in the XSD schema together with the semantic definitions in the model are intended to convey information corresponding to what EDIFACT solutions aim to deliver in implementation guidelines).

## **1.5 Collaboration scenarios (usages).**

The Recommendation identifies a set of **collaborations** or **scenarios** (usages) for the transportXML solutions. Each collaboration is described in a separate **collaboration model**. The *process part* of such a collaboration model is documented in an activity diagram, whereas *the information employed* by the collaboration is part of the total information model (the domain model) we describe. Below we give short textual descriptions of each collaboration.

### **1.5.1 Transport Job.**

- *Collaboration 1 – Domestic transport request (Request):*
  - OrderingParty sends a request to TransportCompany/Forwarder regarding the carrying out of a particular kind of transport service, possibly to be conducted at a particular time.
  - TransportCompany evaluates the request, in particular whether the requested transport job type may be undertaken.
  - If TransportCompany is unable or unwilling to undertake a transport job of such a kind, he returns a negative message to OrderingParty giving the reasons for the decline.
  - If TransportCompany may undertake a transport job of such a kind, he calculates answers to questions posed in the received message (f.ex. freight costs) and returns a positive message to OrderingParty.

- *Collaboration 2 – Domestic transport reservation (Booking)*
  - OrderingParty books a transport service for a particular goods artifact (or goods artifacts) at a specified time to TransportCompany/Forwarder.
  - TransportComapny evaluates the booking terms.
  - If TransportCompany is unable or unwilling to undertake the requested transport job, he returns a negative message to OrderingParty giving the reasons for the decline
  - Alternatively TransportCompany returns to OrderingParty a message telling that he can undertake the transport job, but according to different terms. In this case OrderingParty will evaluate the changed terms – if he finds them acceptable, he confirms the booking in a renewed booking message to TransportCompany. If the changed terms are unacceptable, he must solve his transport needs in another way, in which case no further message exchange needs to take place.
  - If TransportCompany can undertake the booked transport according to the booking terms, he returns a confirmation message to OrderingParty.
  
- *Collaboration 3 – Domestic transport job instruction (Instruction)*
  - This is a message relating to goods which are about to be or which has already been loaded on a pick-up means of transport. DespatchParty/Consignor furnishes the packages involved with bar coded labels and loads the cargo. OrderingParty sends a transport job message to TransportCompany – the message contains information about the goods being loaded.
  - TransportCompnay unloads (and possibly scans) the goods.
  - TransportCompany receives the transport job message, controls entered measurements against the performed scanning results and possibly calculates freight charges.
  - TransportCompany issues a confirmation message (receipt) to OrderingParty and possibly an arrival notice to Consignee.

- *Collaboration 4 – Domestic transport advice (Advice)*
  - Goods arrive or are underway to TransportCompany/Forwarder.
  - If the goods are to be relayed to another TransportCompany/Forwarder, the first TransportCompany may send an advice about the on-road goods to the next transporter in the chain.
  - If the goods are to be customs cleared by Forwarder, such customs clearance takes place, upon which an arrival notice may be sent by Forwarder to Consignee.
  - If the goods are to be customs cleared by Consignee, Forwarder sends an arrival notice to Consignee.
  - If the goods are to be customs cleared by another forwarder, Forwarder(1) sends an arrival notice to Forwarder(2). The latter party performs the customs clearance and may subsequently send an arrival notice to Consignee.
  
- *Collaboration 5 – International import transport (ImportManifest/Mainfest) – international transport as modelled from the receiving country's point of view*
  - DespatchingAgent sends a manifest message to Forwarder in receiving country (Norway).
  - ReceivingAgent unloads the goods – if the cargo displays discrepancies as compared to the manifest or has been damaged, an exception report is sent back to DespatchingAgent.
  - If the cargo does not display any discrepancies as compared to the manifest information, an unloading report may be returned to DespatchingAgent.
  - Thereupon each individual consignment contained in the manifested cargo load is processed as domestic collaborations as described in collaboration 4 (and/or possibly 3).

- *Collaboration 6 – International export transport (Exportt/Mainfest) – international transport as modelled from the despatching country's point of view*
  - Exporter sends a transport job message to Forwarder regarding the carrying out of a particular international transport service to be conducted at a particular time.
  - Forwarder evaluates the transport terms.
  - If Forwarder is unable or unwilling to undertake the transport job, he returns a negative message to Exporter giving the reasons for the decline.
  - Alternatively TransportCompany returns to Exporter a message telling that he can undertake the transport job, but according to different terms. In this case Exporter will evaluate the changed terms – if he finds them acceptable, he confirms the booking in a renewed booking message to Forwarder. If the changed terms are unacceptable, he must solve his transport needs in another way, in which case no further message exchange needs to take place
  - If Forwarder can undertake the booked transport according to the booking terms, he returns a confirmation message to Exporter.
  - Forwarder processes potential pick-up of the cargo to be exported (cfr. collaboration 3).
  - Forwarder performs potentially outbound export clearance and transit handling.
  - Forwarder consolidates and loads the outbound goods and sends thereupon a manifest message to ReceivingAgent in the destination country.
  - ReceivingAgent receives the manifest, unloads the consignment in question (normally together with other consignments in the consolidation) and sends an unloading report and/or possibly an exception report – if discrepancies or damage are involved – back to DespatchingAgent (Forwarder).

- *Collaboration 7 – Complete transport information (Information).*

A participant in the transport chain sends – either on request or without having received a request – complete transport information about a concrete goods artifact / several concrete goods artifacts to another participant in the transport chain. This message may f.ex. convey statistics information given a considerable time after the carrying out of the transport.

- *Collaboration 8 – Quote through a transport portal (PortalSpotQuote)*
- *Collaboration 9 – Auction at a transport portal (PortalAuction)*

The collaborations PortalSpotQuote and PortalAuction both implement the ordering of transport jobs through an electronic market place (portal) based on quotes given by interested transport service providers. The difference between the two collaborations is:

- Using PortalSpotQuote the transport job is unveiled for quoting to a group of selected transport service providers. Agreed-upon business rules for the evaluation by the market place of received quotes will normally apply – the market place will further often represent and store permanent agreements between transport service buyer and transport service provider. The market place is therefore able to auto evaluate a given quote. In the PortalSpotQuote collaboration one quoting company has no knowledge of other companies' quotes.
  - Using PortalAuction the transport job is visible to anyone having access to the portal, and anyone may deliver quotes for the job. No agreed-upon business rules for the processing of given quotes apply – no auto evaluation on the part of the market place is therefore possible. In the PortalAuction collaboration all participants will at any time see and know the best – the lowest - of the quotes given so far in the auction process.
  - In the PortalAuction collaboration details about a transport job may be sent by the market place to a potential quoter when the latter expresses interest in the job (and before he quotes). This option is not relevant in PortalSpotQuote, since in the latter case the complete job data will be made available for the selected group of transport service providers when the invitation for the quote is issued (RequestForQuoteValidated).
  - It is presupposed that the status or status changes of the processes of both collaborations are available to the participating parties.
- *Collaboration 10 – Transport job instruction through a transport portal (PortalInstruction)*

- The collaboration PortalInstruction is a variant of the "ordinary" domestic transport job instruction collaboration (no. 3 above) – the difference being that the instruction is channeled through the electronic market place.
- *Collaboration 11 - Transport reservation through a transport portal (PortalBooking)*
  - The collaboration PortalBooking is a variant of the "ordinary" domestic transport reservation collaboration (no. 2 above) – the difference being that the booking is channeled through the electronic market place.

### **1.5.2 Track&Trace.**

- *Collaboration 12 - Status report (ReportStatus)*
  - An event or a (new) state is caught in the sender's system.
  - If the criteria for the reporting of the caught event or state change are fulfilled, a status message is generated by the sender, otherwise nothing happens. Criteria may f.ex. be that the caught event is related to a party who is "subscribing to" status reports from the party catching the event.
  - Status report is sent and subsequently received by the receiver.
- *Collaboration 13 - Status request with response (RequestCurrentStatus)*
  - Potential status report recipient sends a request message to the status report sender requesting the status of a consignment or package based on a set of input criteria.
  - The status report sender checks and collects the requested status.
  - If it is possible and/or relevant to send a status report, such a report is generated and sent to the party having requested the status report.
- *Collaboration 14 - Status report subscription (NotifyStatus)*
  - Potential status report message receiver defines a set of criteria intended to trigger the sending of status messages from an identified (potential) status report sender to himself.
  - The criteria are sent the (potential) status report sender as a subscription request.
  - The status report sender controls the criteria and stores them in his own status report system.

- Status reports will hereafter be generated and sent the potential status report receiver based on the ReportStatus collaboration (no. 12 above).

## 1.6 Basic design principles and rules for the model / message types.

For those who are to implement solutions based on the transportXML specifications, it may be useful to know the basic design principles upon which the models and message types are built.

Note! This section contains technical data and is primarily written for those who need a thorough knowledge of the models and message type definitions of transportXML !

### 1.6.1 Wide application area per message type: Flexible message structure - profiles.

The Recommendation specifies ***one message model per primary functional area*** (Transport Job and Track&Trace), and thereby one message type per functional area. This means that we have *one message type to convey complete information* about a goods artifact/goods artifacts and *one message type to convey state reports* relating to a goods artifact/goods artifacts.

For each of the message models subsets can be identified to create **profiles**: domestic profile, international transport profile, etc. This will be subsets of the information content of the respective collaboration model. Within closed user groups – potentially by a specific party – further subsets may be identified, either based on already existing profiles or based on the message model itself. The transportXML project considers such a flexibility as vital in order to meet the needs for business variations within the transport industry. We hold it as a requirement that such profiles are defined as real subsets of the message model and that all constructs in a “profile message” consequently can be related back to the collaboration/domain model. Given this requirement the profile concept will not prevent the intended harmonization (or standardization) of XML based solutions.

As of October 2002 three profiles have been defined for TransportJob:

- **DomesticNO**: Domestic transport within Norway.
- **DomesticNOMin**: Subset av DomesticNO – the least common information set which all Norwegian domestic transport providers using transportXML commit themselves to support. It means that if the information of this minimal set is given by an ordering party, the sender shall be guaranteed that the transport company will be able to carry out the job.
- **TransportPortal**: Profile for transport job handling through a market place/portal.

In addition to (potentially) belong to a **profile**, a message will have a well defined **message function** within a defined **collaboration**. Example (the function BookingDecline in the Booking collaboration):

```
<TransportJob
  version="3.0"
  definedBy="Norsk EDIPRO"
  profile="DomesticNO"
  domain="transportXML"
  collaboration="Booking"
  messageFunction="BookingDecline"
  messageId="1004">
```

This principle means a.o. that no model or message constructs are considered as mandatory in the transportXML specification itself. In **profiles** we may, however, specify the subset (of classes and attributes) which *we recommend to be present in an instance of a message* in order to guarantee that a collaboration is carried out as intended whichever collaboration parties – provided they support the respective profile - are involved.

### 1.6.2 Representing generalizations/specializations in model and message type.

The domain model contains many generalization/specialization structures (superclass/subclass). When defining such structures the following two principles have been applied:

In the model subclasses have been explicitly modelled with associations normally linked to each separate subclass (the alternative would have been to have several alternative associations linked to the superclass thereby incorporating the semantics of the subclass into the semantics of the association), cfr. f.ex. the model part containing the Party superclass.

In the XML message type a subclass is represented by an element whose name is the name of the superclass and carrying an attribute *subClass* with a value = the subclass name. Example:

```
<Address subClass="PhysicalAddress">
  <PostalCode>0575</PostalCode>
  <City>OSLO</City>
  <CountryCode>NO</CountryCode>
</Address>
```

This technique enables us to “select” superclasses which may have several *roles* in a concrete message instance and only represent such a class in one place in the message.



### 1.6.3 Data types in model og message type.

Within the international ecommerce standardization body UN/CEFACT a recommendation for the use of **core component types** has been specified. Such core component types are meant to be data and object types for all information objects to be transferred as part of an ecommerce message. The core component type recommendation is part of a broader specification of **core components** ("context free" basic information objects in which the data elements are typed in accordance with core component types) and **business information entities** ("context dependent" information objects built upon core components).

Norsk EDIPRO has in its specifications from the *Infrastructure for Electronic Commerce* project recommended that core component types should be used as data types in all electronic collaborations. transportXML has chose to follow this recommendation.

In version 1.8 of the Core Component specification 11 CC types are defined: Amount, Code, DateTime, Graphic, Identifier, Indicator, Measure, Numeric, Picture, Quantity and Text. Each of these types contains a *basic data element* named **Content** representing *the content value* of the element/object for which the type is applied. In addition a CC type has a number of *auxiliary attributes* – typical examples are CurrencyIdentificationCode for Amount, UnitCode for Quantity and LanguageCode for Text.

In transportXML we have chosen to use a limited selection of the core component types. We have further chosen to restrict the use of auxiliary attributes according to what we have found appropriate. We do emphasize, however, that the selection we have made represents a real subset of the CC specification and thereby real core component types.

Our use of core component types is summarized in the following table:

Core component type	Applied auxiliary attributes	Comment
Text	None	The CC specification uses LanguageCode – we consider this attribute unnecessary in transportXML.
Numeric	None	The CC specification introduces the FormatText attribute in order to specify a derived numeric type (for example integer or decimal).
Quantity	unitCode	The CC specification introduces additionally the attributes UnitCodeListIdentifier and UnitCodeListAgencyIdentifier (cfr. 1131/3055 in EDIFACT).
Measure	unitCode	

Amount	currencyIdentificationCode	
Indicator		A logical true/false attribute. The CC specification specifies the use of the auxiliary attribute FormatText in order to convey how the logical value is represented. We see no need to explicitly specify this in a message instance. In transportXML we use the values <b>YES</b> and <b>NO</b> .
Code	codeListIdentifier codeListAgencyIdentifier name	The name attribute may be used to give a textual description in case no code value is applicable. The CC specification additionally introduces the auxiliary attributes CodeListVersionIdentifier and LanguageCode.

The length of a model attribute or XML data value is not restricted in transportXML.

All the auxiliary attributes are considered optional in transportXML. The Measure type may f.ex. be used without the inclusion of the unitCode attribute as long as the measurement unit is considered known.

#### 1.6.4 Attributes as opposed to elements in an XML message type.

transportXML follows the following recommendation of Norsk EDIPRO: Model attributes are to be translated into XML attributes if they are considered as meta data and into XML elements in all other cases. In particular transportXML uses XML attributes in order to represent *subclass identifiers* (the XML attribute **subClass**, cfr. above), *measurement units*, *identifiers* and *internal references* (to identifiers in other places in the same message).

#### 1.6.5 Representing core component types in XML messages.

The Norsk EDIPRO recommendation regarding the use of elements as opposed to attributes in XML messages has a very precise meaning when it comes to the XML representation of the core component types: the content value (Content) becomes the element content in XML whereas the auxiliary attributes (f.ex. unitCode) becomes attributes in XML. In accordance with the naming conventions described in section 1.6.6 below the attribute name will be written using lower camel case.

An amount in transportXML may therefore look like this:

```
<CodAmount currencyIdentificationCode="EUR">
  34560,50
</CodAmount>
```

Each of the core component types will be defined as a separate type in the XSD schemas for the transportXML message types. These definitions will all follow the same "template" – as an example Amount is defined in the following way:

```
<xsd:complexType name="Amount">
  <xsd:simpleContent>
    <xsd:extension base="xsd:float">
      <xsd:attribute name="currencyIdentificationCode" type="xsd:string"/>
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>
```

#### 1.6.6 Naming convention: Camel case.

In accordance with the recommendations of Norsk EDIPRO and ebXML transportXML uses a naming convention in which **UpperCamelCase** is used for XML elements and **lowerCamelCase** is used for XML attributes.

When choosing (English) terms for the names of model classes and attributes and consequently XML elements and attributes we have as much as possible used names employed in corresponding international works and projects – we have in particular tried to use terms from the Swedish Pharos project where they have been relevant.

#### 1.6.7 Basic objects and structures – the structure of message model and message type.

In transportXML we apply a message model structure – and thereby a message type structure – in which the objects which may be used in several locations in a message instance, f.ex. because the concrete object instance may play several roles or because it is related to several "units" (or other objects) in the message, are "extracted" (or factorized) and represented as **basic objects** (we introduce a BasicObject class as a containment class for such objects – this becomes a BasicObject element in XML). As such a basic object an object's full representation will exist at only one place in the message instance. In a separate Structure part of the model (and the message type) we define all **the structures** in which the basic objects are involved – in this part these objects are represented solely through references. This technique has many advantages:

- The information content of an object is represented only once. In a TransportJob message a party may play several roles, a product may relate to many consignments, etc.
- It increases the potential for flexible message structures, since the structure combinations in the Structure part of the message may be exactly as many as and exactly in line with the concrete need of the transaction at hand.

- We believe that we in this way will catch the semantic information of the domain model more fully. Normally one risks to lose semantic information when transforming a domain model to a message model/message type, primarily because the domain model is a network of information with semantics possibly linked to complex relationships, whereas the message model/message type represents a hierarchy. By placing the relationship structures in a separate Structure part the resulting message type will more faithfully reflect the semantics of the collaboration/domain model.
- We may more easily exchange components in the message model/message types. We expect that internationally standardized UML and XML based core components and business information entities will emerge which may be usable for transport collaboration. Such a component might f.ex. represent the transport product concept. In a subsequent version of transportXML we could then replace our own Service class by the internationally standardized component just by pulling out one basic object and putting in another one – one place in the message only – and leave the structure of the message model and the message type unchanged.

## 2 Transport domain: Domain model and collaboration models

This chapter presents a "platform independent" model for the transport domain including the collaborations which have been defined for this domain by the transportXML project.

**The domain model** describes all the information which is relevant for the defined collaborations. The domain model is described by means of an information model consisting of a set of class diagrams together with semantic descriptions of all classes, attributes and relationships. The class diagrams are displayed in chapter 2.1. In order to reduce complexity of presentation the attributes are not shown in these diagrams – they are instead documented as part of the "Semantic descriptions of classes and attributes" in chapter 2.3. The description of the relationships of the domain model is given in chapters 2.4 and 2.5.

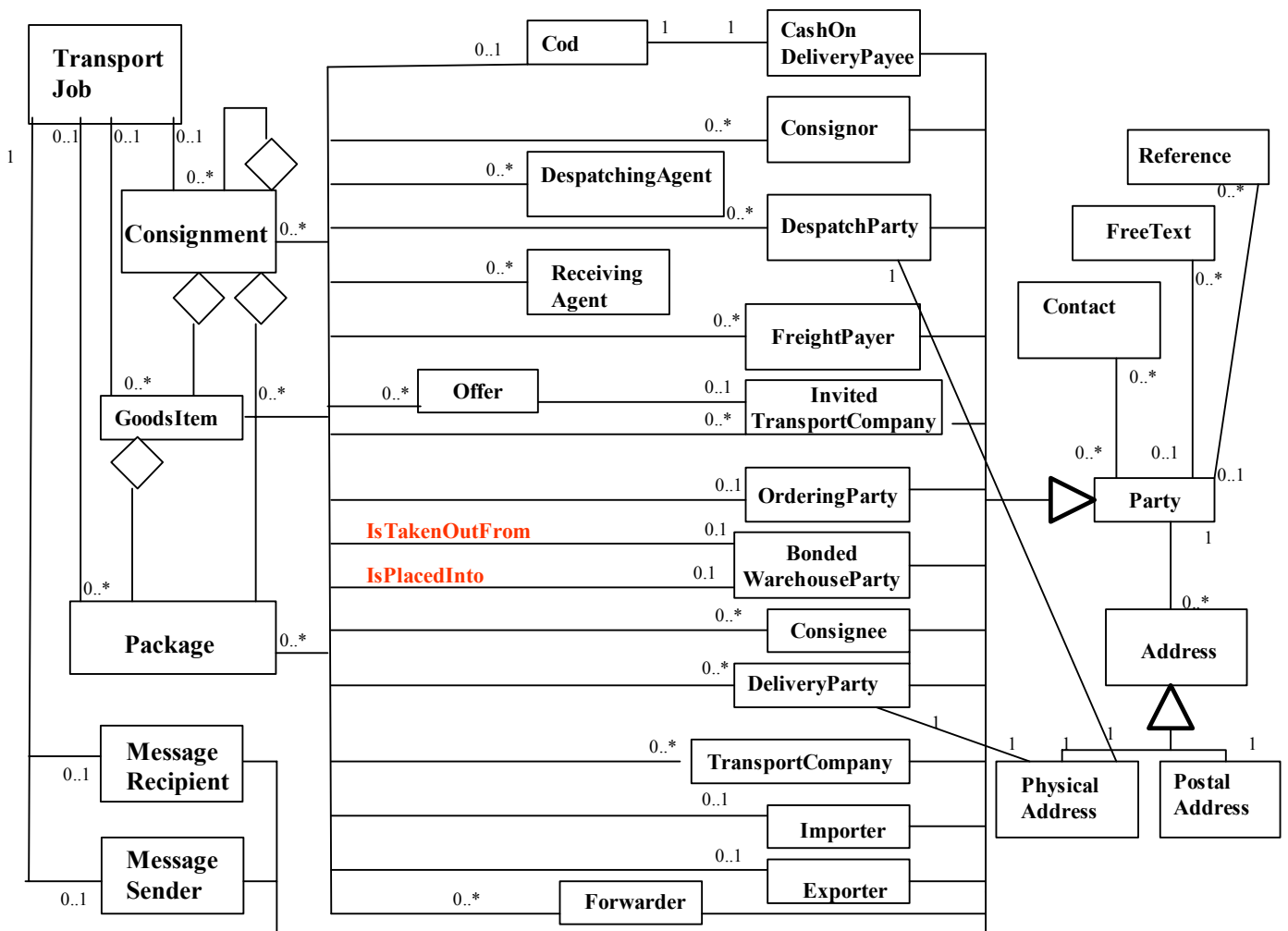
The collaborations are described through **a set of collaboration models** in chapter 2.2. Each collaboration is described by means of a separate *activity diagram* (Note that in line with Norsk EDIPRO's recommended *description technique* the collaboration model consists of this activity diagram plus the subset of the domain model defining the information used by the collaboration in question).

The descriptions of chapter 2 are platform independent in the sense that they do not prescribe any syntax or structures in the documents – f.ex. the message types – which are to realize the collaborations being described. The structures of the message types defined by transportXML – *TransportJob* and *TrackAndTrace* – are defined through **the message models** which are documented in chapter 3.

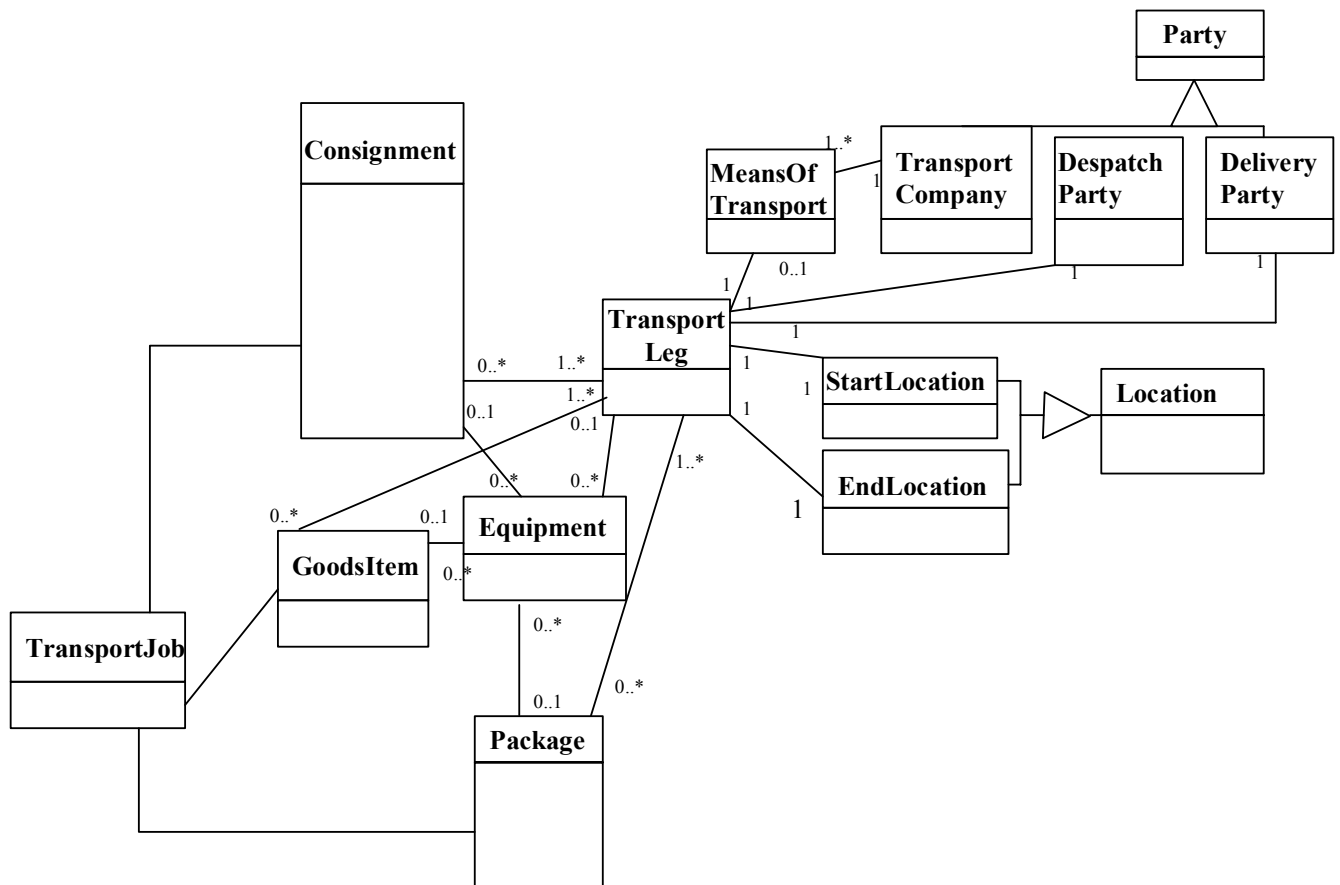
## 2.1 Domain model: Information model described using class diagrams

Note: The "goods" classes Consignment, GoodsItem and Package, as well as their mutual relationships and their relationships to the TransportJob class, are only fully documented in section 2.1.1.

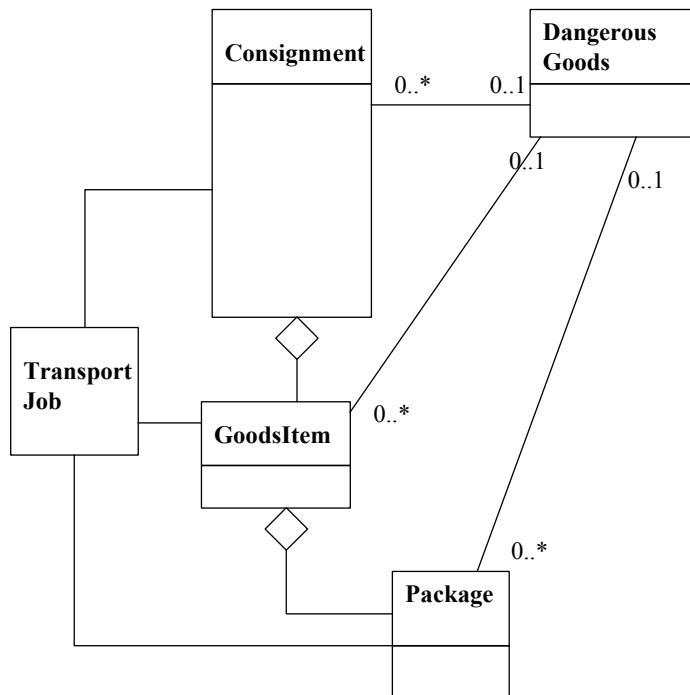
### 2.1.1 Class diagram - part: Party, goods classes (Consignment/GoodsItem/Package) and the TransportJob class



## 2.1.2 Class diagram - part: TransportLeg

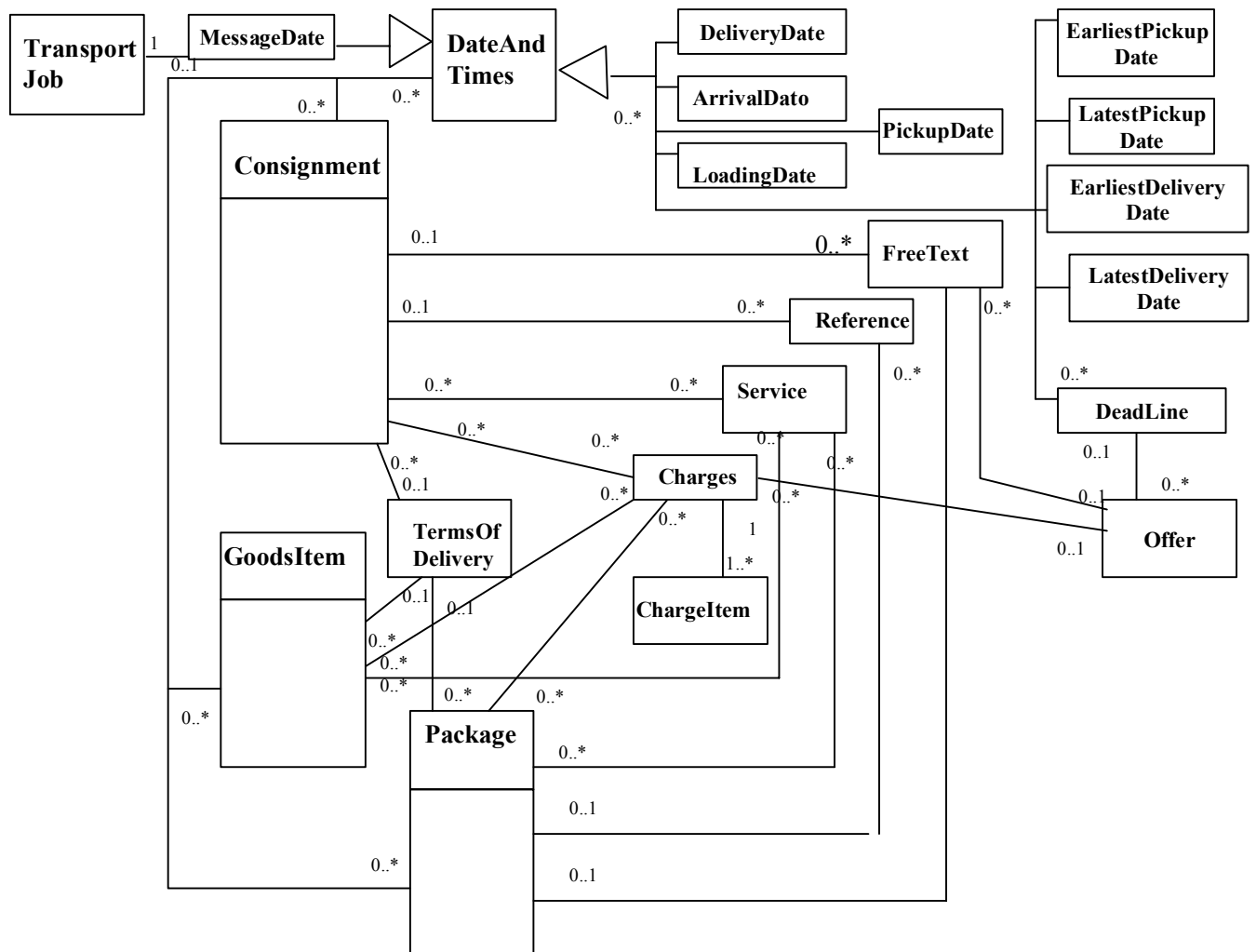


### 2.1.3 Class diagram - part: DangerousGoods

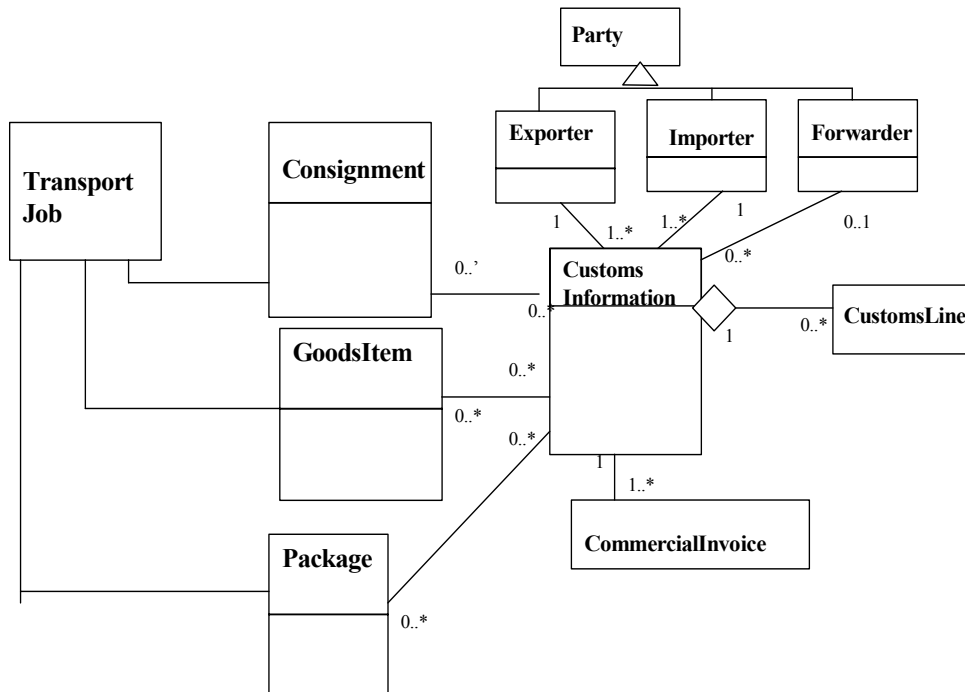




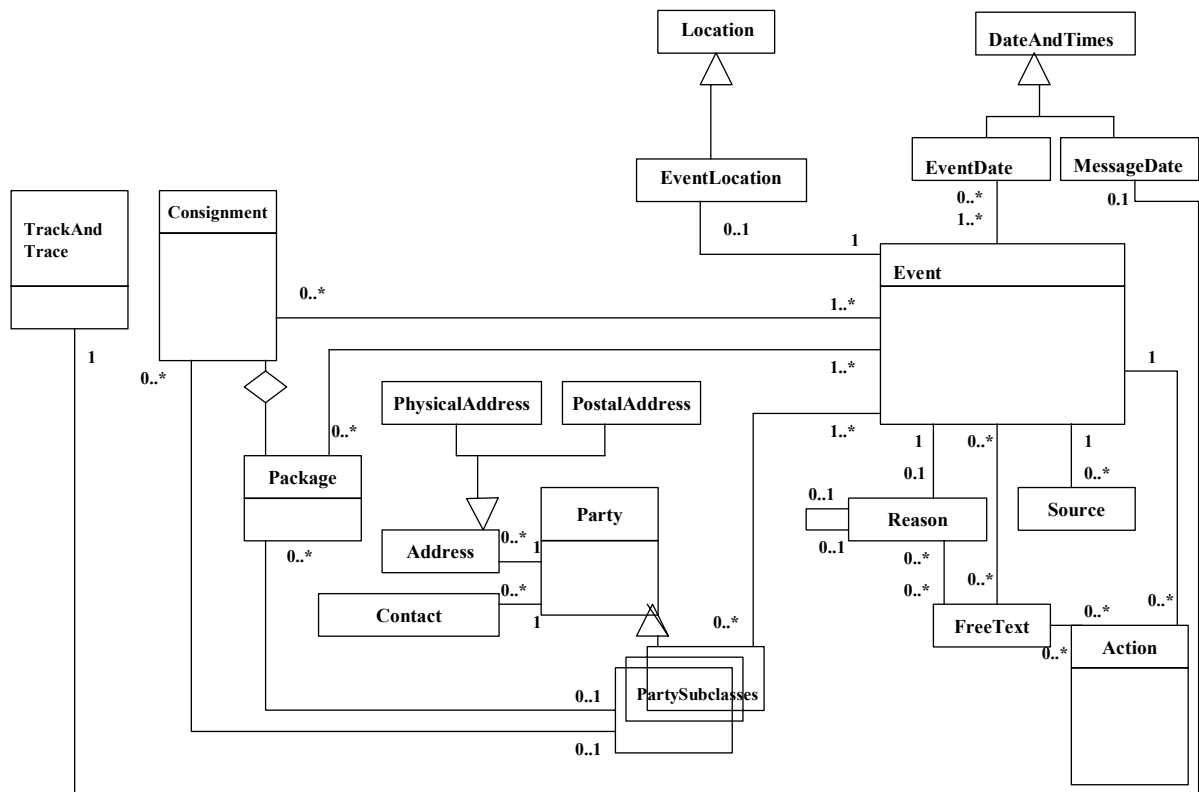
## 2.1.4 Class diagram - part: Freight – Service – DateAndTimes



### 2.1.5 Class diagram - part: CustomsInformation



## 2.1.6 Class diagram - part: TrackAndTrace



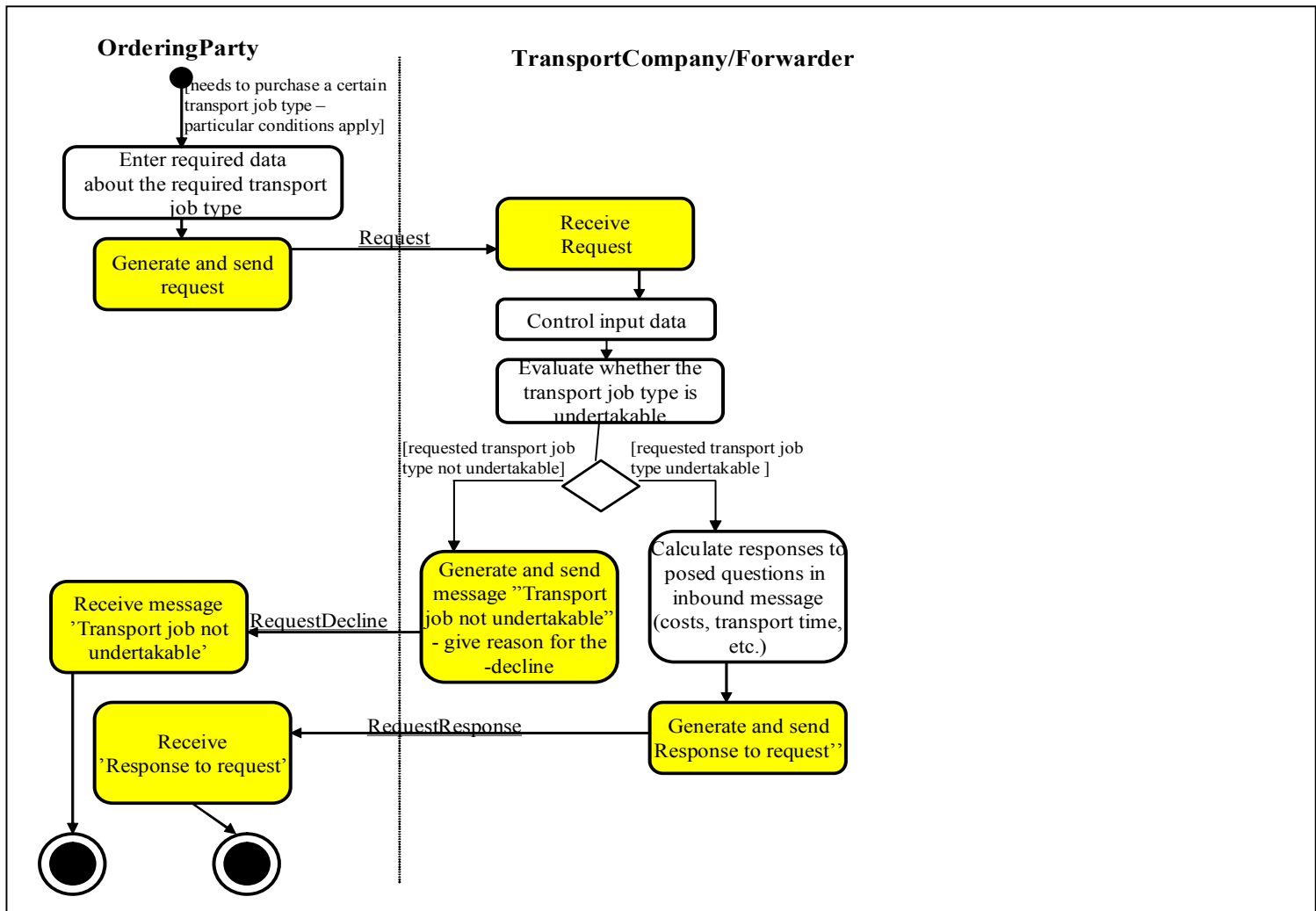
## 2.2 Collaboration models – process models described using activity diagrams

Explanation of the various constructs in the activity diagrams:

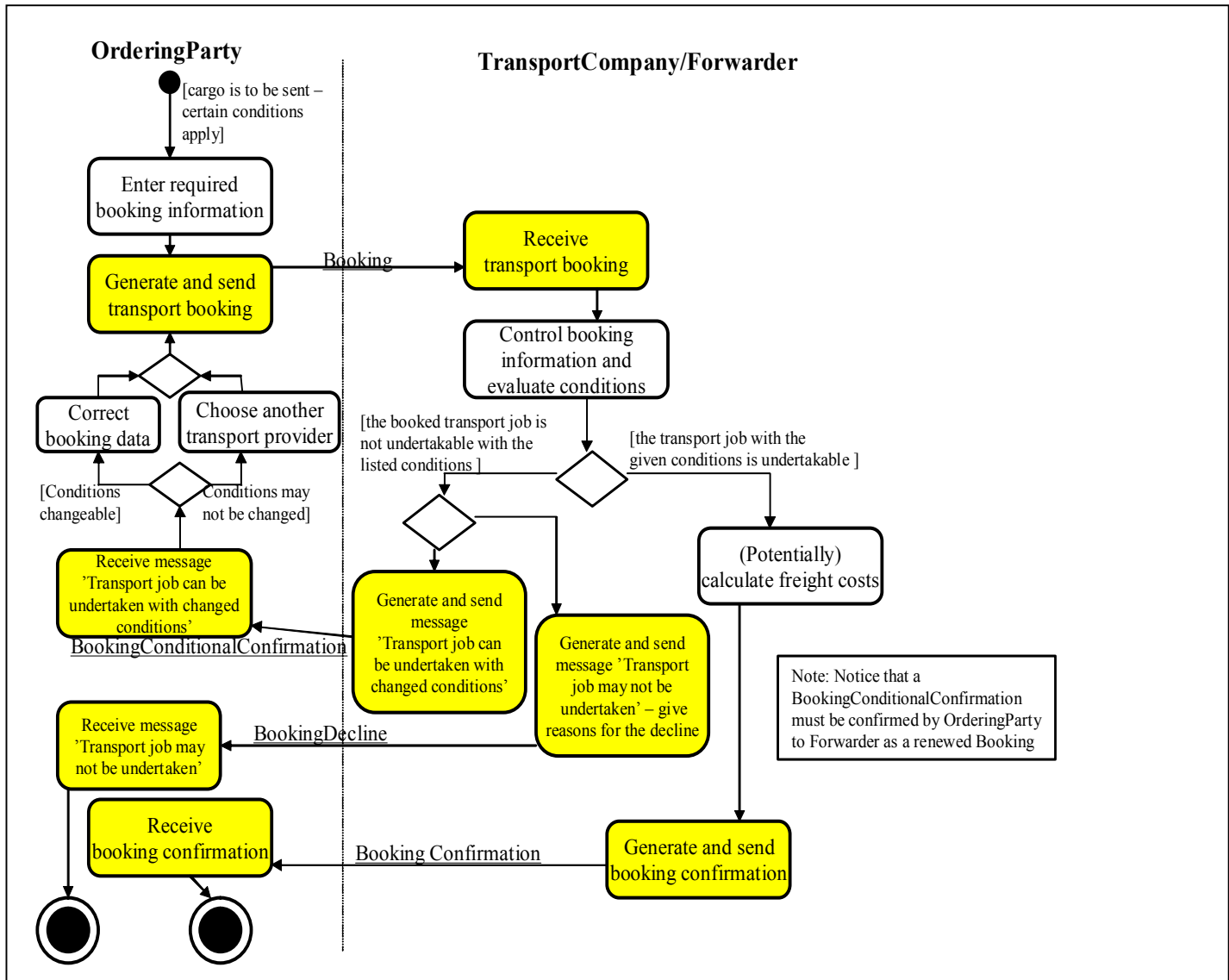
- ❑ **Swimlane:** An activity diagram is divided into two or more swimlanes – these lanes are separated by dotted vertical lines. Each swimlane describes the activities/processes taking place at one *participant* in the collaboration (or one *role*) – this participant/role is named above the swimlane.
- ❑ **Activity:** A set of operations performed by one participant/role. Briefly described in "ovale squares".
- ❑ **Branch:** A point in the process at which the evaluation of a condition leads to the choosing of one of several possible further process pathes. Denoted by a "diamond".
- ❑ **Fork/Join:** Start and stop of two or more parallel processes. Denoted by a solid bar.
- ❑ **Source/Sink:** Start and stop of the collaboration process itself. Only one source may be given; there may, however, exist several sinks. Denoted by circles, the source circle being completely black whereas the sink circle contains a white ring.
- ❑ **Activity coloured yellow:** This denotes an activity which is directly related to the sending or receiving of a TransportJob/TrackAndTrace message to/from another participant in the collaboration. The message instance itself is denoted by *an underlined name* as it flows across the swimlane (it is often called a *flow object*). This underlined name is used in the transportXML messages as the official name of *the message function* and is a mandatory ingredient of the root element of the respective XML message.
- ❑ **Activity coloured green:** This denotes an activity which is directly related to the sending or receiving of a TransportJob/TrackAndTrace message – however, it is an activity further described in another transportXML collaboration and thus does not provide any further details – like the message function identification – in the activity diagram at hand.
- ❑ **Activity coloured blue:** This denotes an activity which is directly related to an electronic collaboration which is defined outside the scope of transportXML (it might, for example, be an electronic customs clearance process).

## 2.2.1 Collaborations in the Transport Job functional area

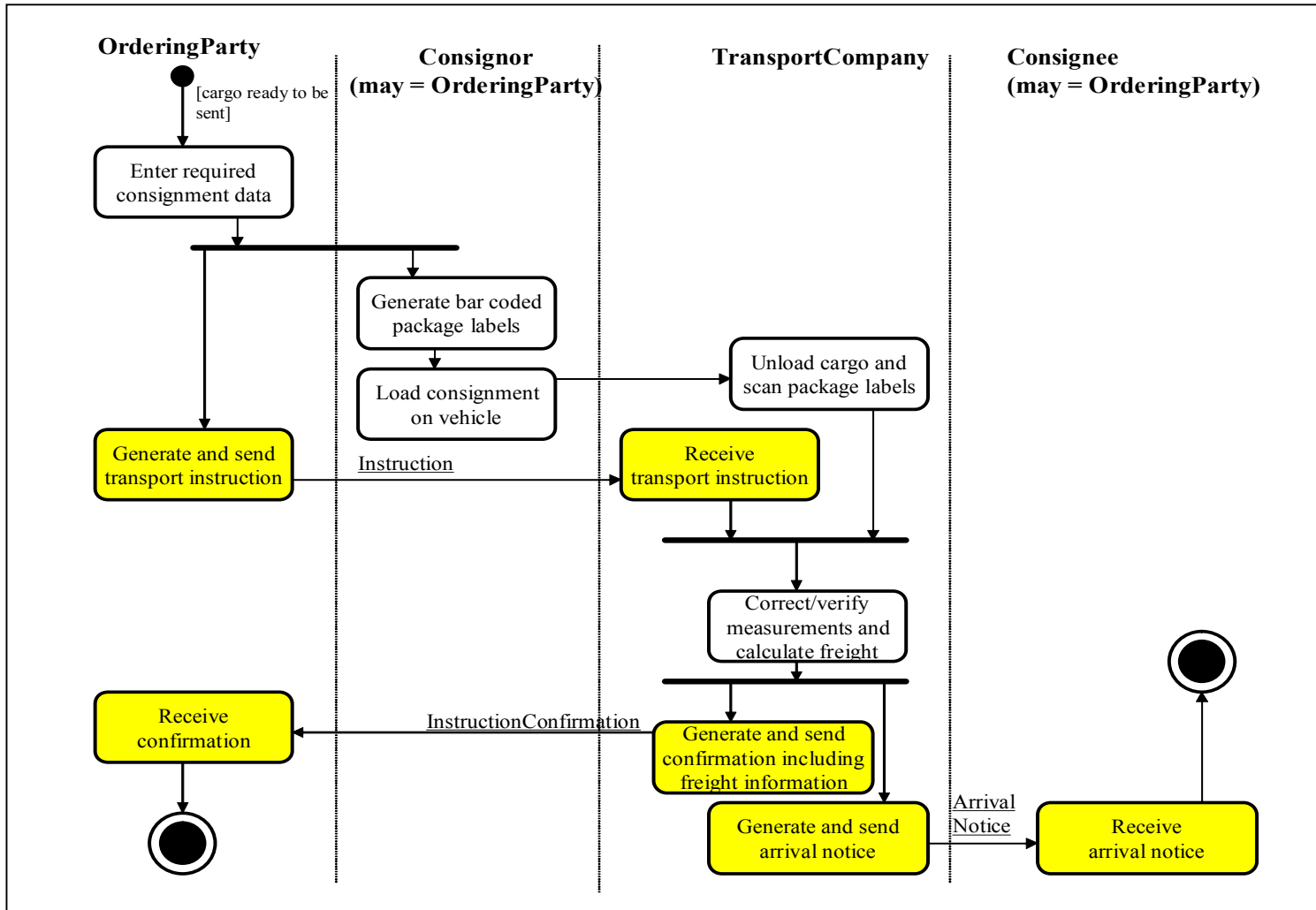
### 2.2.1.1 Collaboration: Domestic transport request (Request)



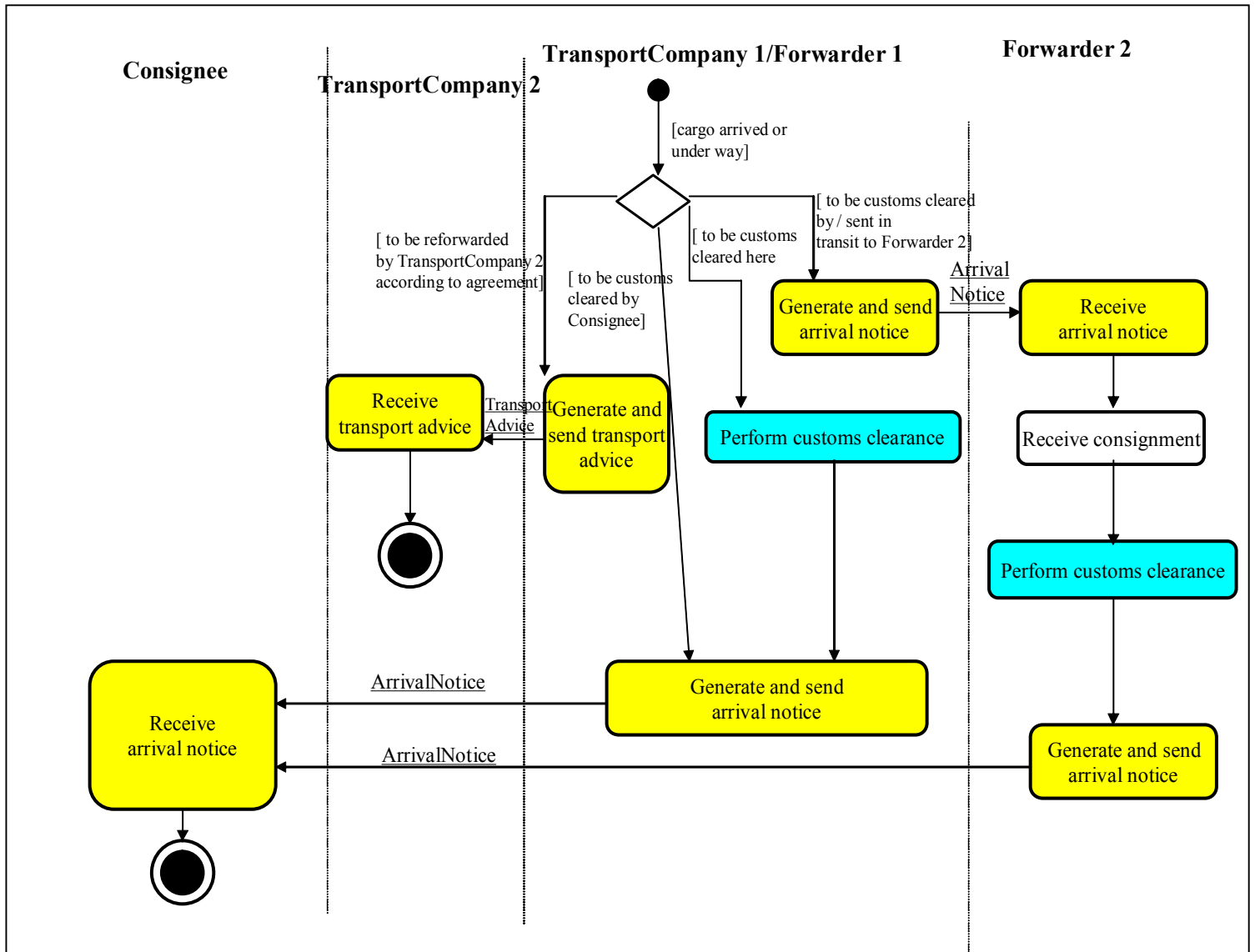
### 2.2.1.2 Collaboration: Domestic transport reservation (Booking)



### 2.2.1.3 Collaboration: Domestic transport job instruction (Instruction)

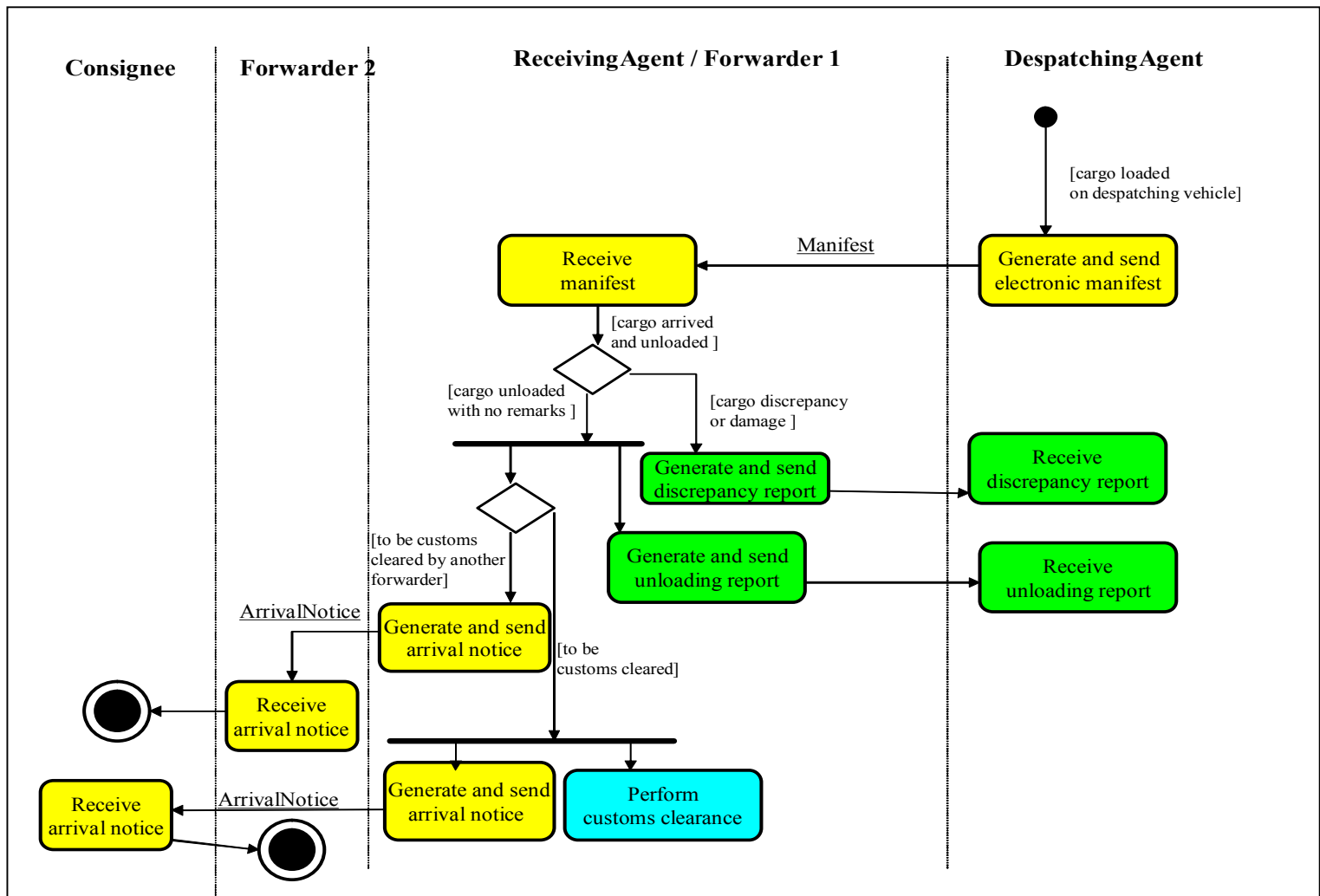


#### 2.2.1.4 Collaboration: Domestic transport advice (Advice)

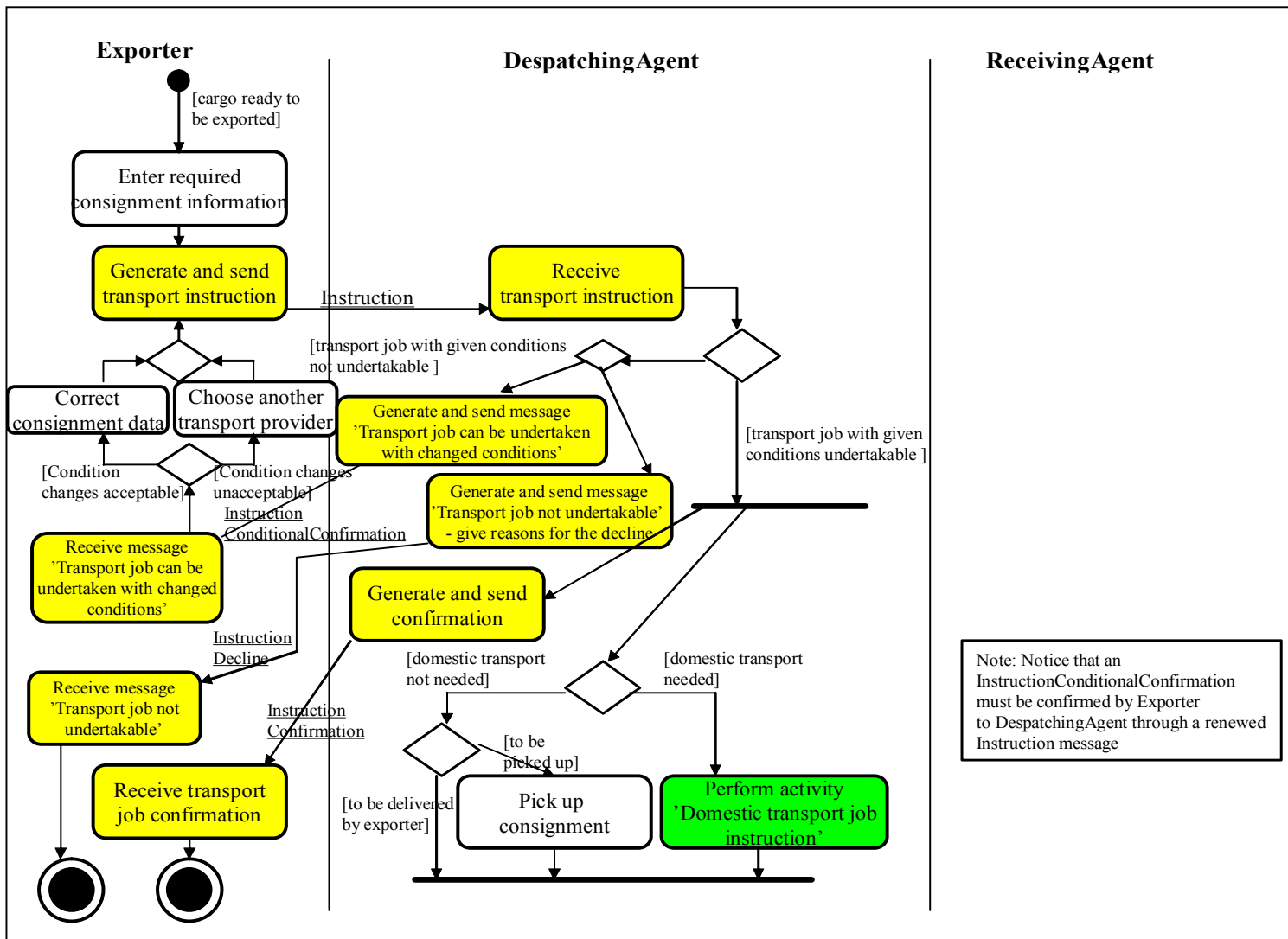




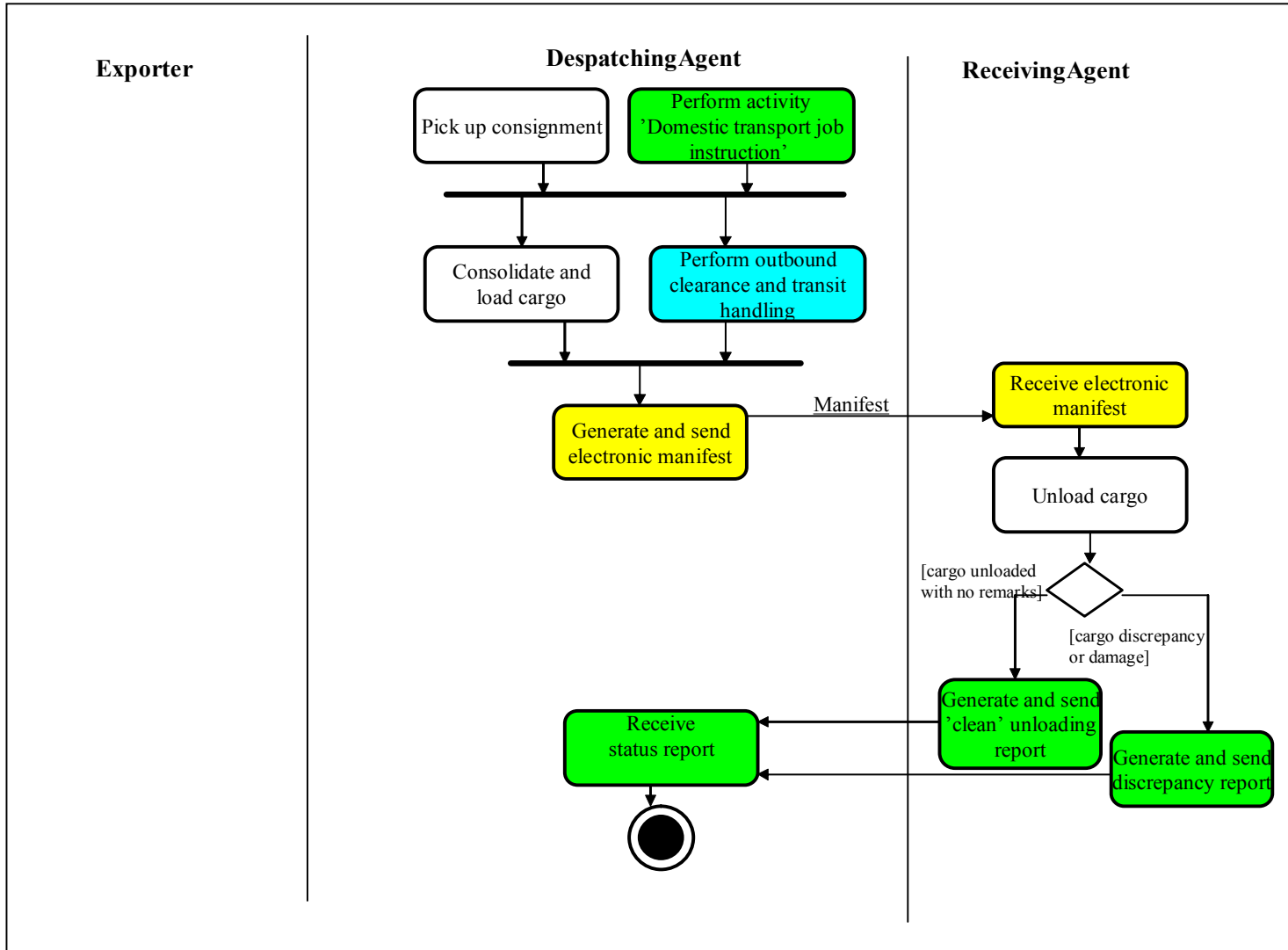
### 2.2.1.5 Collaboration: International import transport (ImportManifest/Manifest)



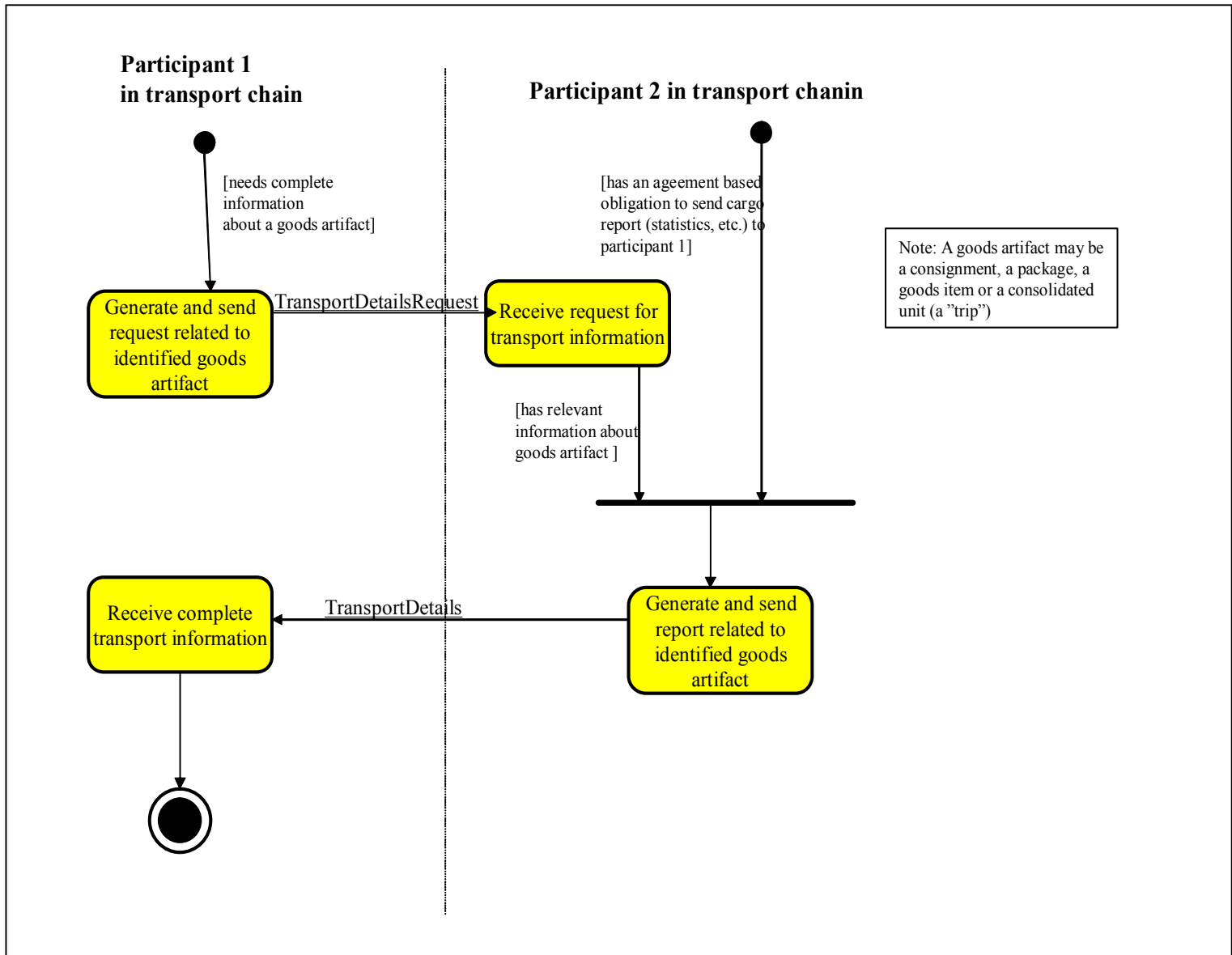
### 2.2.1.6 Collaboration: International export transport (ExportManifest)



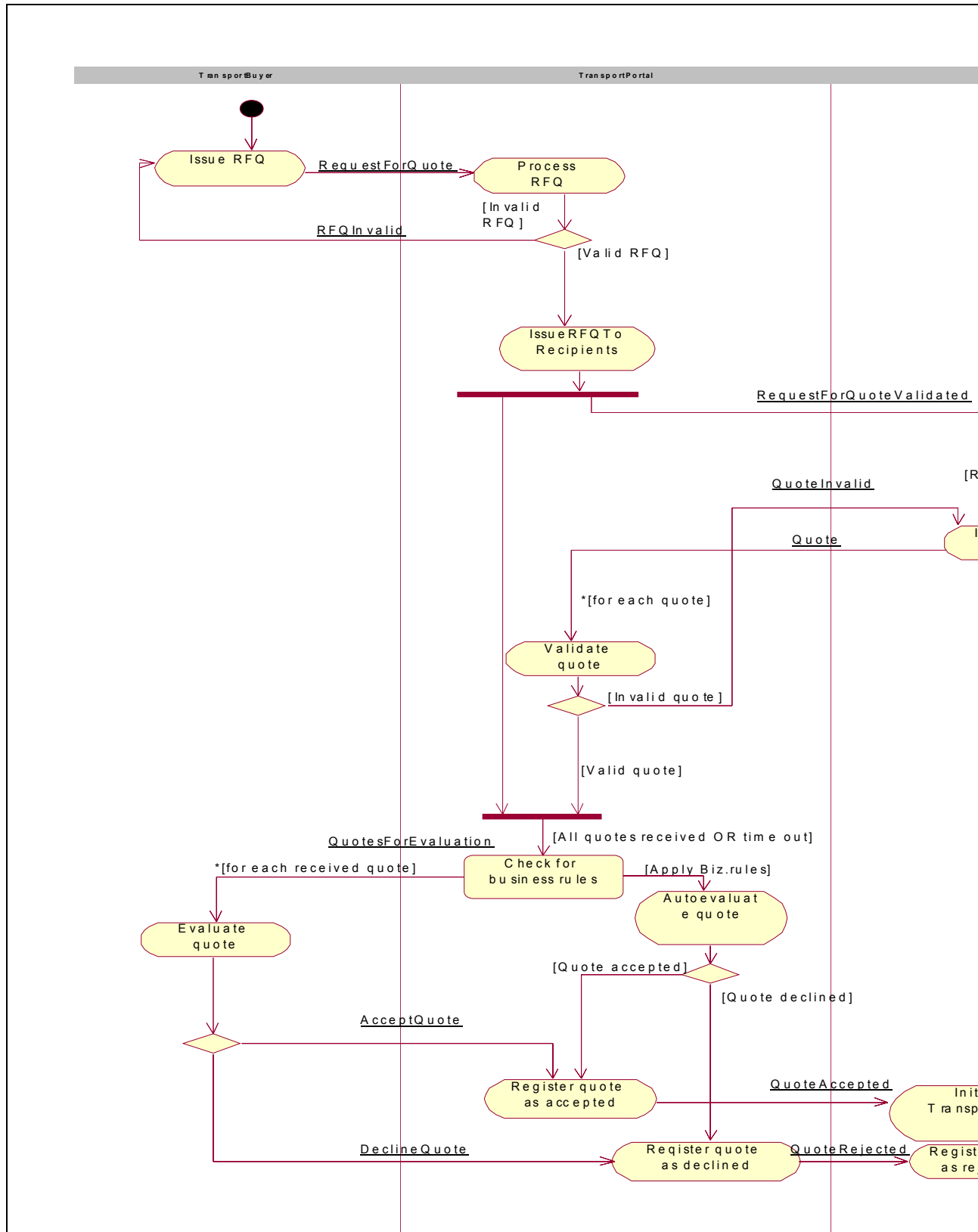
## Collaboration: International export transport (page 2)



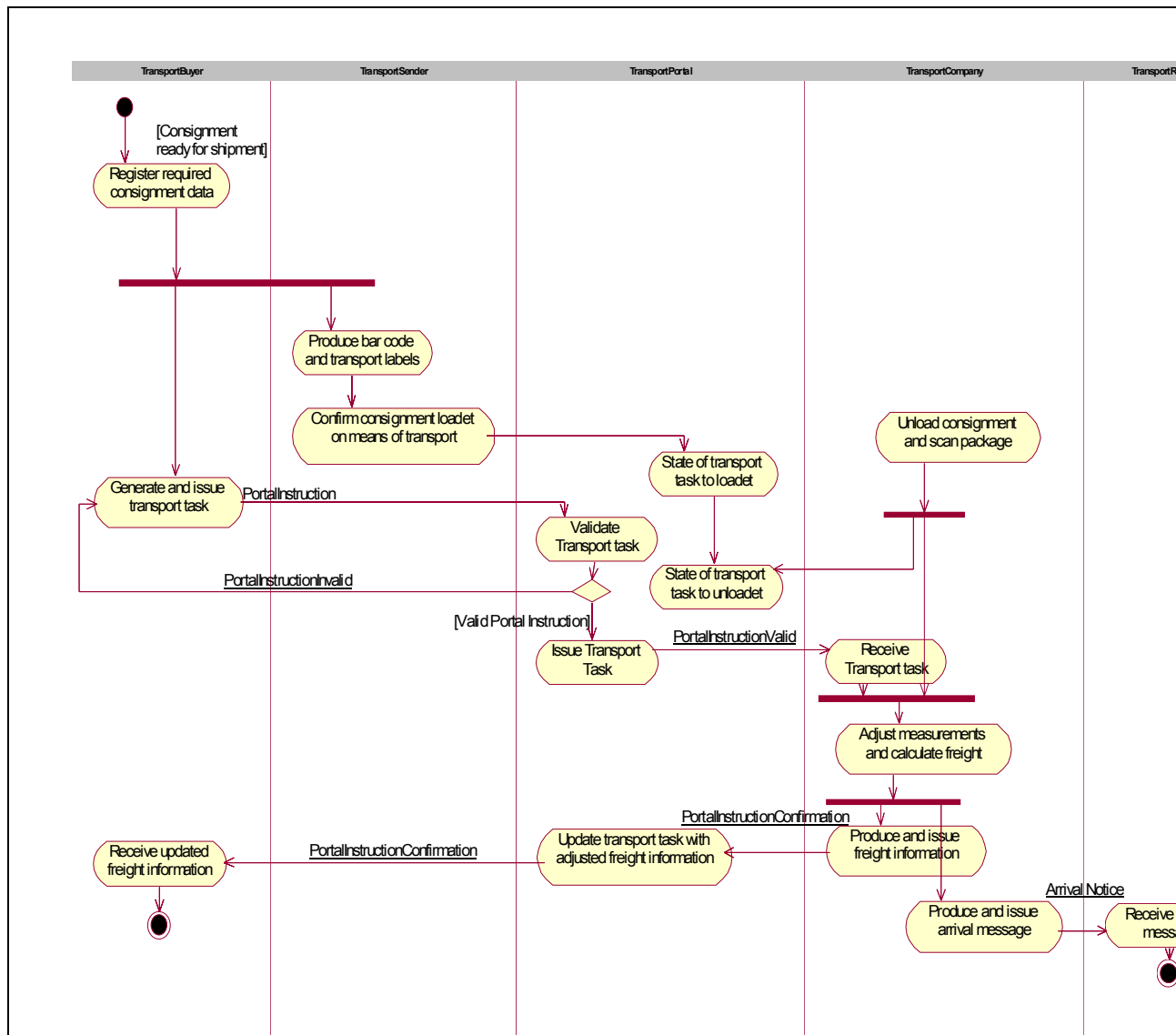
### 2.2.1.7 Collaboration: Complete transport information (Information)



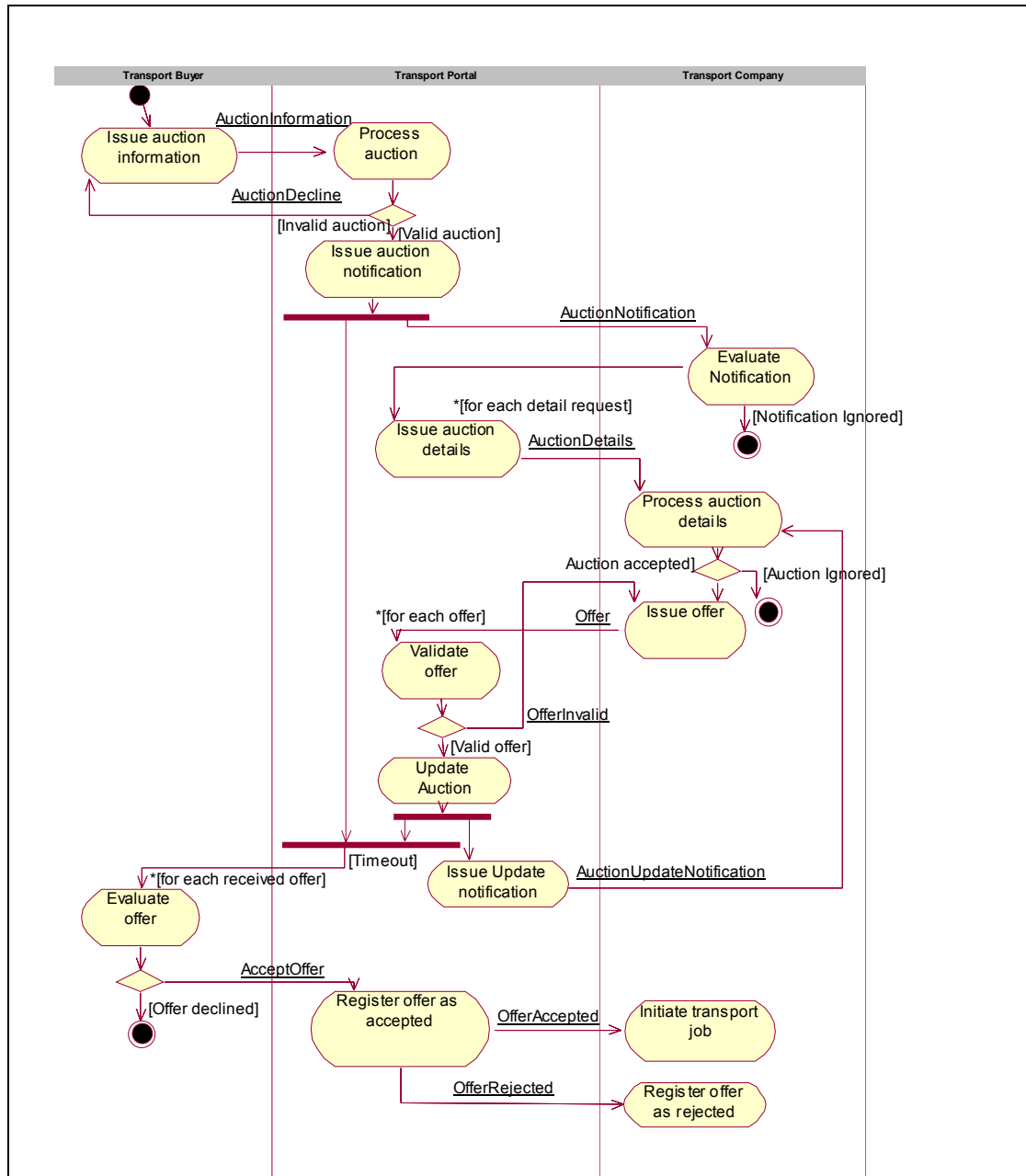
### 2.2.1.8 Collaboration: Quote through a transport portal (PortalSpotQuote)



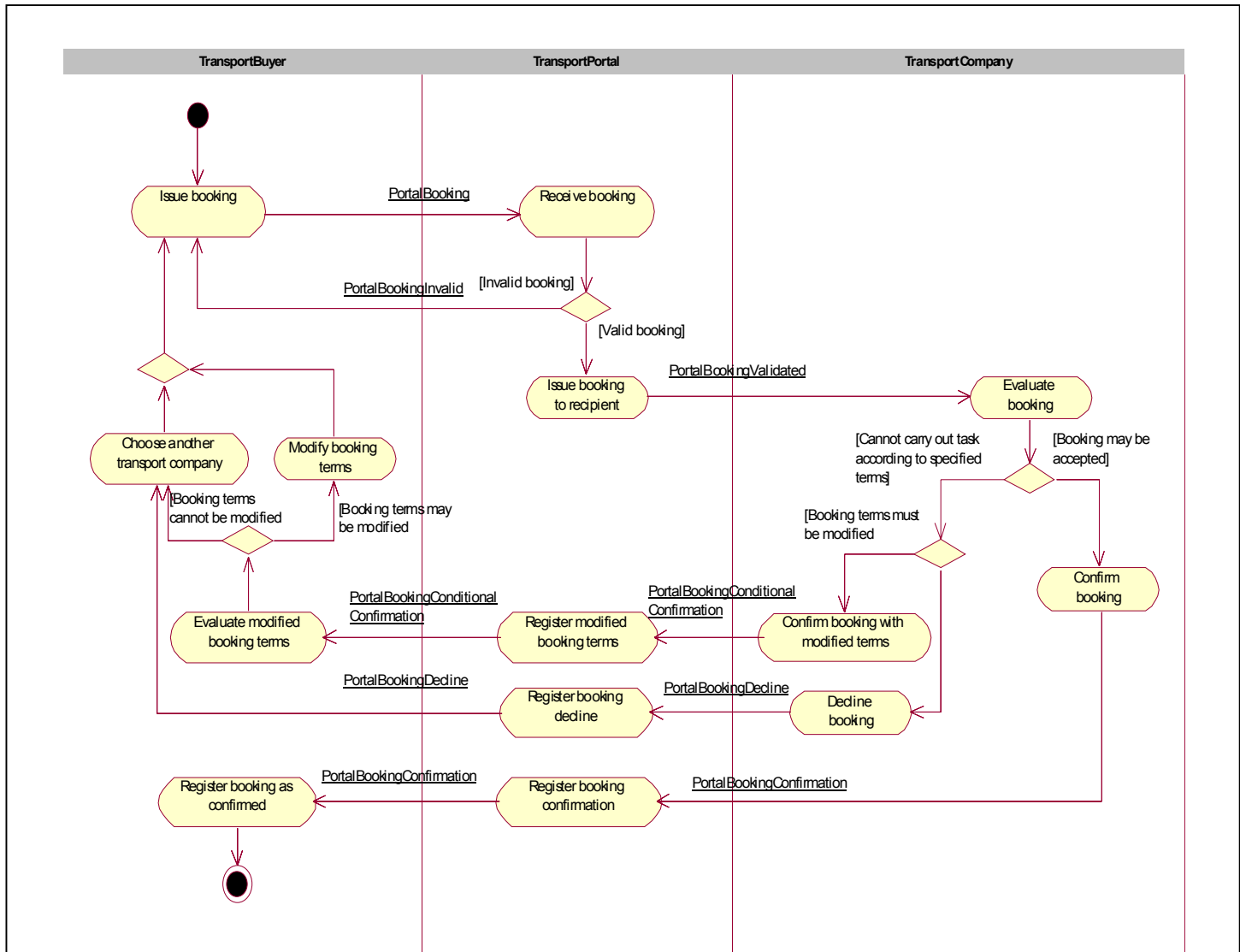
### 2.2.1.9 Collaboration: Transport job instruction through a transport portal (PortalInstruction)



### 2.2.1.10 Collaboration: Auction at a transport portal (PortalAuction)



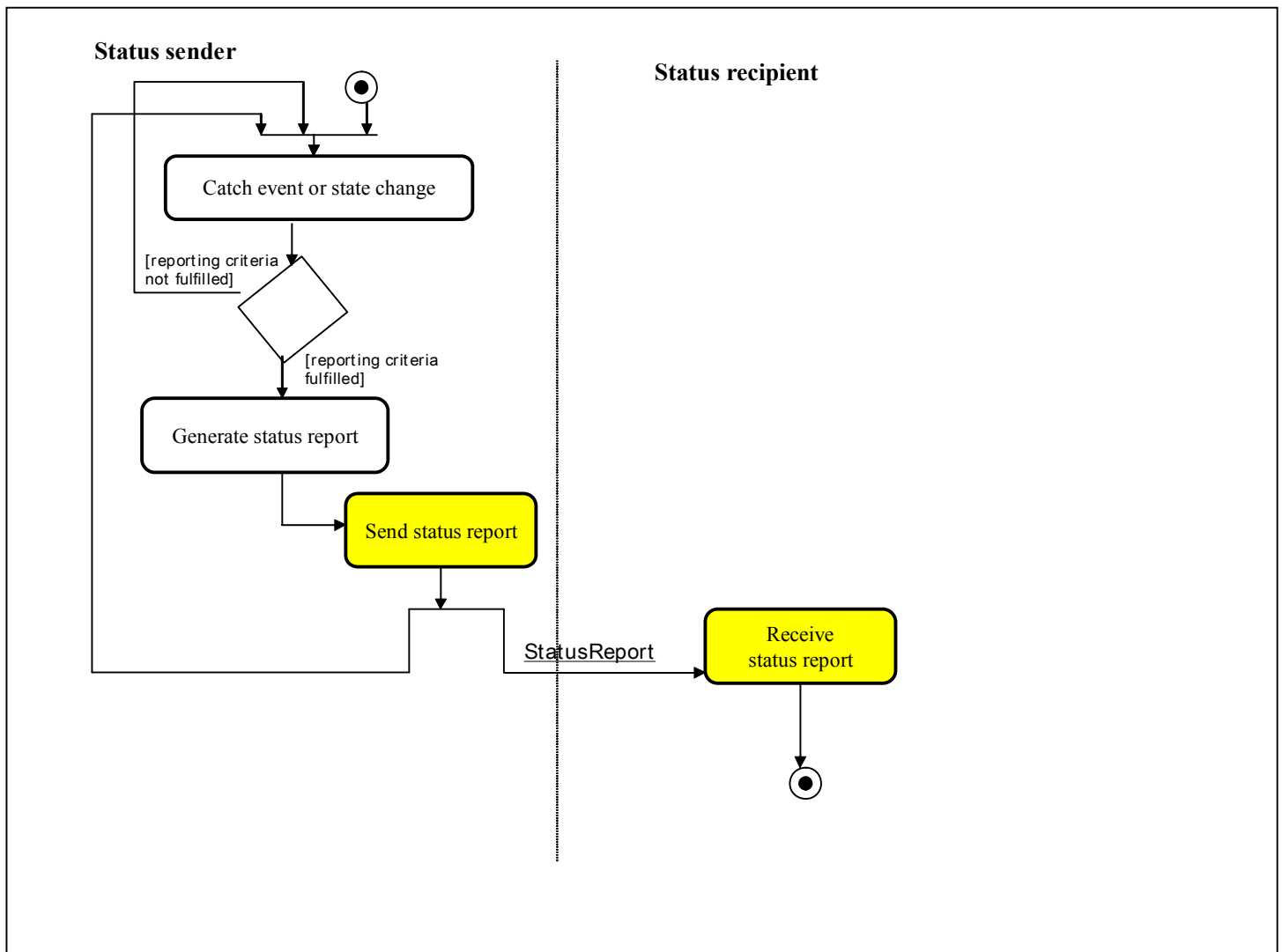
### 2.2.1.11 Collaboration: Booking through a transport portal (PortalBooking)



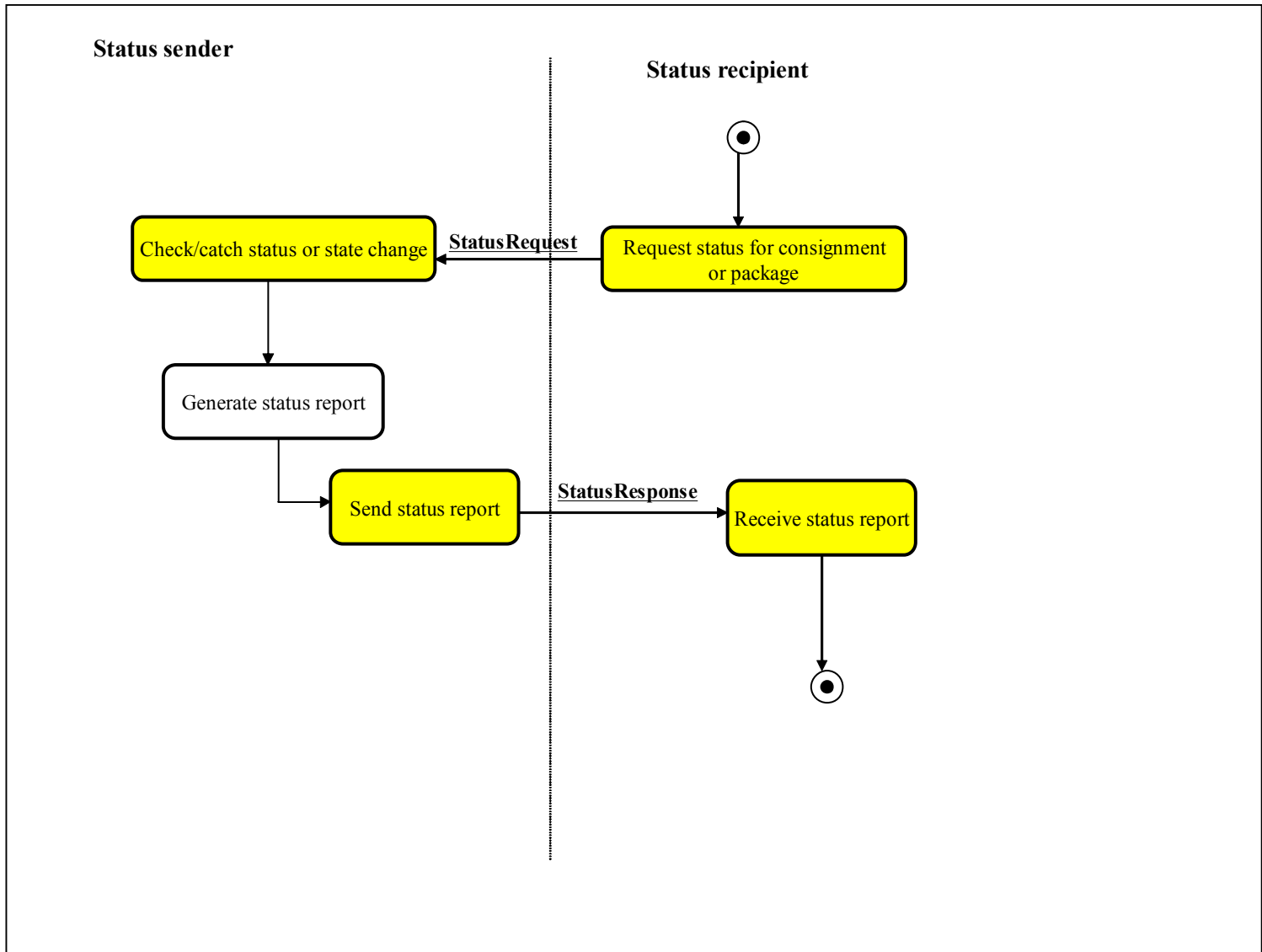


## 2.2.2 Collaborations in the Track&Trace functional area.

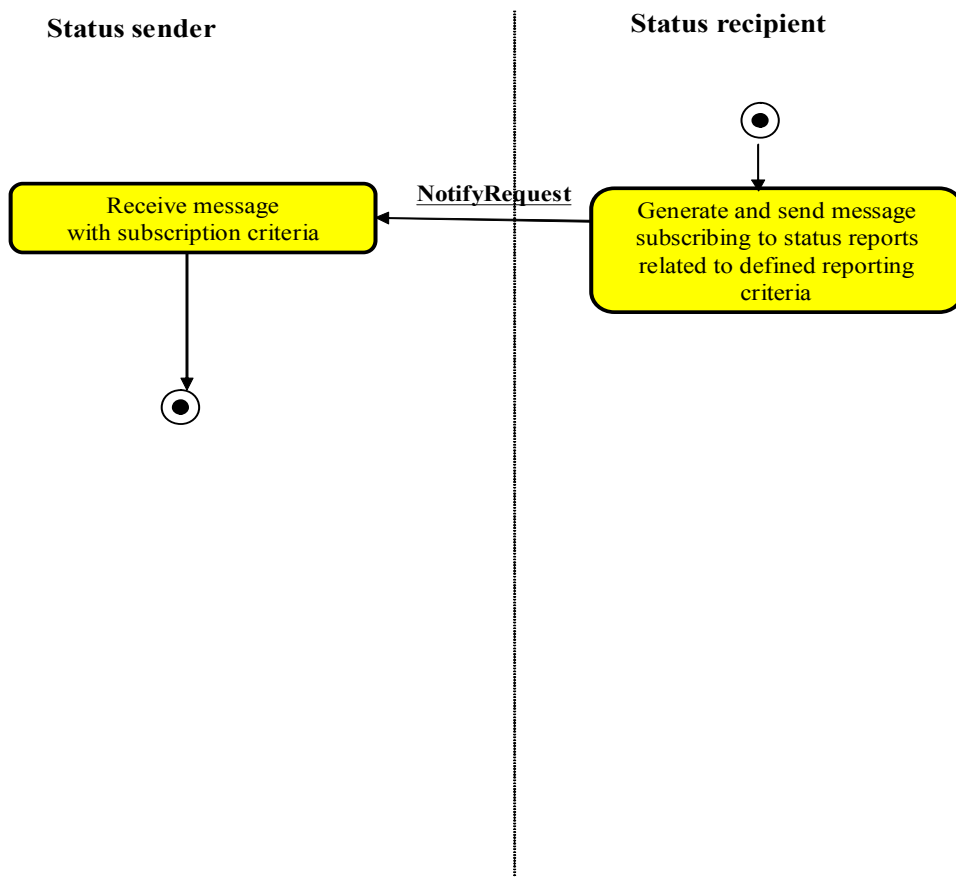
### 2.2.2.1 Collaboration: Status report (ReportStatus)



### 2.2.2.2 Collaboration: Status request with response (RequestCurrentStatus)



### 2.2.2.3 Collaboration: Status report subscription (NotifyStatus)



## 2.3 Domain model – semantic descriptions of classes and attributes

This chapter describes all the information contained in the transport domain model. This includes classes, attributes and applied codes. Each class and each attribute is given a semantic description defining the meaning content of the respective information object within our transport business context. We further specify the data types (based on core component types). In some cases we provide enhanced explanations and short examples.

All the information objects related to all the collaborations in both functional areas Transport Job and Track&Trace are described.

In the column **Usage** we include additional information in the form of two letter code combinations – one defining a profile, the other one giving a recommendation for the Track&Trace functional area:

- **DNoM**– this class and/or attribute is part of *the mandatory subset* of the **DomesticNOMin** profile.
- **TT** – this class and/or attribute is *recommended for use* within the Track&Trace functional area (which does not preclude it from being relevant for the Transport Job functional area).

The relationships of the information model are described in chapters 2.4 and 2.5.

In the semantic descriptions all names of model objects (classes or attributes) are specified with an initial upper case letter and in singular mode. Further the “object oriented” *dot notation* is used to express containment (f.ex. CustomsInformation.BondedWarehouseNo = the BondedWarehouseNo attribute of the CustomsInformation class).

Class name	Semantic description	Attributes	Type	Specifications	Usage	Codes
<b>Address</b>	The location of Party. May be a physical location or the location of the PO box (depending on the subclass).					
	Postal location identification according to the Norwegian (and mostly European) template	PostalCode	Text		<b>DNoM</b> <b>TT</b>	
	The place name which the local or national postal organization has linked to Address.PostalCode.	City	Text		<b>DNoM</b> <b>TT</b>	
	Postal location identification according to the template used in the US, UK, etc.	ZipCode	Text	The choice between PostalCode and ZipCode depends on the value of CountryCode	<b>TT</b>	
	Identification of an area encompassing several Address.City, f.ex. County in Great Britain or State in the US.	StateOrRegion	Text		<b>TT</b>	
	Code uniquely identifying the country in which Address.City is	CountryCode	Code		<b>TT</b>	Ref.

Class name	Semantic description	Attributes	Type	Specifications	Usage	Codes
	located.					ISO 3166-1 (two-letter code)
<b>PhysicalAddress</b>	The physical location of Party.				<b>DNoM</b>	
	Name of street or road containing the physical location of Party.	Street	Text	If the street name and street number is to be represented as one attribute, the attribute AddressLine is to be used.	<b>TT</b>	
	Number or Number/letter connected to the physical location of PhysicalAddress.Street and which further confines the physical location of Party.	StreetNo	Text	If the street name and street number is to be represented as one attribute, the attribute AddressLine is to be used.	<b>TT</b>	
	Further identification or confinement of the location of Party (f.ex. floor, building: ex. 2 <sup>nd</sup> floor)	AddressLine	Text	Repeatable	<b>DNoM (Mandatory if relevant)</b> <b>TT</b>	
<b>PostalAddress</b>	PO box address				<b>DNoM</b>	

Class name	Semantic description	Attributes	Type	Specifications	Usage	Codes
	The post office to which the PO box is associated in the official postal box address.	PostOffice	Text		TT	
	Number or other coded identification identifying a PO box, possibly in combination with PostalAddress.PostOffice.	POBox	Text		TT	
<b>Action</b>	Action which has been taken or which is recommended to be taken after an exception/discrepancy event				TT	
	Definition of Action	ActionCode	Code		TT	
	Denotes whether Action has been or will be taken by the message sender, whether it is recommended to be taken by the message recipient or whether it is given	ActionType	Code		TT	Requested Information Performed

Class name	Semantic description	Attributes	Type	Specifications	Usage	Codes
	purely informational.					
<b>Offer</b>	The response of Party.TransportCompany to a request for quote or opened auction at a transport portal.					
	The portal's reference to Offer	OfferId	Text			
<b>ChargeItem</b>	Information about one type of transport related charges (f.ex. freight). Contains information to be presented on the transport invoice.					
	Code identifying ChargeItem.	ChargeCode	Code			<a href="http://www.edipro.no/transportXML/codes/chargeCod">www.edipro.no/transportXML/codes/chargeCod</a>



Class name	Semantic description	Attributes	Type	Specifications	Usage	Codes
						<a href="#">es</a>
	Text describing ChargeItem.	ChargeText	Text			
	Code denoting whether VAT has been or is to be calculated based on ChargeItem.NetAmount.	VATCode	Code			0=VAT not to be debited 1=VAT to be debited
	Charged amount including VAT (if VAT is chargeable)	GrossAmount	Amount			
	Charged amount excluding VAT	NetAmount	Amount			
	Charged NetAmount according to the issuer's ordinary price list	TariffAmount	Amount			
	ChargeItem.TariffAmount – ChargeItem.NetAmount	Discount	Amount			
	Additional information or comment to ChargeItem	Comment	Text			
<b>Charges</b>	The collection of charges (= the set of ChargeItem objects) for a transport service related to					

Class name	Semantic description	Attributes	Type	Specifications	Usage	Codes
	Consignment, GoodsItem or Package = FreightPayer's cost					
<b>Cod</b>	Delivery restriction. The goods are only to be delivered if the content of Cod has been fulfilled (f.ex. through the valid receipt for a paid amount)			Cod may be related to the freight charges as well as the goods value.		
	Type of Cod (f.ex. reimbursement, international cheque, bank receipt, etc.)	CodType	Code			<a href="http://www.edipro.no/transportXML/codes/codTypeCodes">http://www.edipro.no/transportXML/codes/codTypeCodes</a>
	Client related concept for the sales/purchase ledger accounting. Identifies a payment transaction (in Norway referred to as a KID).	TransactionIdentifier	Text			
	The amount to be paid before the goods may be delivered.	CodAmount	Amount			

Class name	Semantic description	Attributes	Type	Specifications	Usage	Codes
	Control digit related to Cod.CodAmount .	ControlDigit	Numeric 1			
	Document number referred to by the delivery restriction statement.	DocumentId	Text	Repeatable		
<b>Consignment</b>	Goods transported on behalf of OrderingParty from one or more Consignor to one or more Consignee.				<b>DNoM</b>	
	A reference number uniquely identifying Consignment	ConsignmentId	Text	For Norwegian domestic transports the consignment number recommendation of Norsk EDIPRO should be applied.	<b>DNoM</b> <b>TT</b>	
	Identifies the type of ConsignmentId (f.ex. the Norwegian standard defined by Norsk EDIPRO)	IdType	Code	For Norwegian domestic transport transportXML recommends EANSSCC.	<b>DNoM</b> <b>TT</b>	<b>EANSSCC</b> =Serial Shipping Container Code as defined in Norsk EDIPRO's recommendation "Dokumentasjon i

Class name	Semantic description	Attributes	Type	Specifications	Usage	Codes
						innenlandstransport ” <b>Other=</b> ”Default”
	Number of pallet places ordered for the transport of Consignment or occupied during the transport of Consignment.	NoOfPalletFootPrints	Quantity			
	Value of the commodity or commodities contained in Consignment.	GoodsValue	Amount			
	Amount for which the commodity or commodities contained in Consignment, have been insured through the type of insurance identified by Consignment. TypeOfInsurance.	InsuranceValue	Amount			
	Type of insurance.	TypeOfInsurance	Text			
	Description of the commodities contained in Consignment.	GoodsDescription	Text		<b>DNoM</b>	
	Total number of weight units (of the type denoted by unitCode) including number of weight units of	TotalGrossWeight	Measure		<b>DNoM</b>	<b>KGM</b> =kilogram <b>GRM</b> =gram <b>TON</b> =ton

Class name	Semantic description	Attributes	Type	Specifications	Usage	Codes
	the packaging material.					
	Total number of volume units (of the type denoted by unitCode).	TotalVolume	Measure			<b>DMQ</b> =cubic decimetres <b>MTQ</b> =cubic metres
	Total number of length meters occupied by Consignment in MeansOfTransport	LoadingMetres	Measure			<b>MTR</b> =loading metres
	Text, number or other id contributing to the identification of Consignment	Marking	Text			
	Number of packages (pieces)	NumberOfPackages	Quantity		<b>DNoM</b>	<b>PCE</b> =number
<b>CommercialInvoice</b>	Invoice from goods seller to goods buyer (normally Exporter and Importer respectively).					
	A number given by the goods seller which uniquely identifies CommercialInvoice in relation to himself.	InvoiceNo	Text			

Class name	Semantic description	Attributes	Type	Specifications	Usage	Codes
	The date the goods seller has entered on CommercialInvoice when it was issued.	InvoiceDate	Text			
	The total amount which the goods seller has entered on CommercialInvoice as the charge for or value of the commodities to which CommercialInvoice pertains.	InvoiceAmount	Amount			
<b>Contact</b>	A means – or medium – through which the associated Party may be contacted or communicated with.					
	Telephone number.	PhoneNo	Text			
	Telefax number.	TelefaxNo	Text			
	Address for electronic mail.	EmailAddress	Text			
	Web address.	WebAddress	Text			

Class name	Semantic description	Attributes	Type	Specifications	Usage	Codes
	Name of the person whom Contact relates to.	ContactName	Text	If the code value Receipt is used in ContactFunction, this attribute contains the signature of the person signing a proof of delivery.	<b>TT</b>	
	Job title, department or other function of Contact.ContactName.	ContactFunction	Text		<b>TT</b>	<b>Receipt</b> (used in POD information – TT).
<b>CustomsInformation</b>	Information to be used for the customs clearance of the commodity or commodities contained in Consignment.					
	Number allocated by the customs authorities when Consignment is released for delivery and use after completed customs clearance.	CustomsClearance No	Numeric			

Class name	Semantic description	Attributes	Type	Specifications	Usage	Codes
	Statistical value used for the calculation of import VAT and duties when the commodities contained in Consignment are customs cleared.	CustomsValue	Amount			
	Insurance amount which together with the goods value, freight charges and other value components add up to create CustomsValue.	InsuranceAmount	Amount			
	Serial number, a.o. identifying a BondedWarehouseParty, which is allocated to the consolidation of which Consignment is part when Consignment enters a bonded warehouse or is sent in direct transit.	BondedWarehouse No (007)	Text			
	Sequential number within CustomsInformation.BondedWarehouse useNo.	SequenceNo	Numeric			



Class name	Semantic description	Attributes	Type	Specifications	Usage	Codes
	Sequential number within CustomsInformation.BondedWarehouseNo.SequenceNo. BondedWarehouseNo, SequenceNo and SubLevelSequenceNo together identify Consignment.	SubLevelSequenceNo	Numeric			
<b>CustomsLine</b>	Information about one commodity group or category for customs clearance and statistical purposes.					
	The country in which the commodity group has been produced.	CountryOfOrigin	Code			ISO 3166-1 (two-letter code)
	Statistical number according to the customs tariff.	StatisticalNo	Code			<b>HS</b> =Harmonized System
	No of packages (pieces).	NumberOfPackages	Quantity			
	Net weight in kilograms as defined by the customs authorities.	NetWeight	Measure			
	Gross weight in kilograms as defined by the customs authorities.	GrossWeight	Measure			

Class name	Semantic description	Attributes	Type	Specifications	Usage	Codes
	The total customs value of the commodity group defined by CustomLine.	StatisticalValue	Amount			
	Quantity in a unit – different from kilograms – as required specified by the customs authorities.	QuantityOtherUnit	Quantity			
	A textual description of the goods declared in CustomsLine easing a unique customs related identification.	GoodsDescription	Text			
<b>DangerousGoods</b>	Information about Consignment, GoodsItem or Package for which specific rules and regulations defined by the ADR or UMDG specifications apply.					
	Please refer to the ADR manual chapter 5.4 (and corresponding chapters for other DG regulation conventions).	UNNo	Numeric			
	Please refer to the ADR manual	Commodity	Text			

Class name	Semantic description	Attributes	Type	Specifications	Usage	Codes
	chapter 5.4 (and corresponding chapters for other DG regulation conventions).					
	Please refer to the ADR manual chapter 5.4 (and corresponding chapters for other DG regulation conventions).	Class	Numeric			
	Please refer to the ADR manual chapter 5.4 (and corresponding chapters for other DG regulation conventions).	Group	Text			
<b>DateAndTimes</b>	A point in time or a time interval.					
	Year A.D. (f.ex. 2002).	Year	Numeric 4		<b>DNoM</b> <b>TT</b>	
	Month as a sequence number (1-12) within DateAndTimes.Year.	Month	Numeric 2		<b>DNoM</b> <b>TT</b>	
	Day as a sequence number (1-31)	Day	Numeric		<b>DNoM</b>	

Class name	Semantic description	Attributes	Type	Specifications	Usage	Codes
	within DateAndTimes.Month.		2		TT	
	Hour sequence number (0-23) within DateAndTimes.Day.	Hour	Numeric 2		TT	
	Minute sequence number (0-59) within DateAndTimes.Hour.	Minute	Numeric 2		TT	
<b>DeliveryDate</b>	DateAndTimes at which Consignment, GoodsItem or Package is expected to be delivered, required to be delivered or has been delivered at the premises of DeliveryParty.					
<b>ArrivalDate</b>	DateAndTimes at which Consignment, GoodsItem or Package is expected to arrive, required to arrive or has arrived at a specific location.					
<b>LoadingDate</b>	DateAndTimes at which Consignment, GoodsItem or Package is expected to be loaded, required to be loaded or has loaded					

Class name	Semantic description	Attributes	Type	Specifications	Usage	Codes
	onto MeansOfTransport at a specific location.					
<b>MessageDate</b>	DateAndTimes at which the current message instance has been generated.				<b>DNoM</b>	
<b>PickupDate</b>	DateAndTimes at which Consignment, GoodsItem or Package is expected to be picked up, required to be picked up or has been picked up at the premises of DespatchParty.				<b>DNoM</b>	
<b>EventDate</b>	DateAndTimes at which Event has happened.				<b>Only TT</b>	
<b>EarliestPickupDate</b>	The earliset DateAndTimes at which Consignment, GoodsItem or Package is expected to be or required to be picked up at the premises of DespatchParty.					
<b>LatestPickupDate</b>	The latest DateAndTimes at which Consignment, GoodsItem or Package is expected to be or required to be picked up at the premises of DespatchParty.					

Class name	Semantic description	Attributes	Type	Specifications	Usage	Codes
<b>EarliestDeliveryDate</b>	The earliest DateAndTimes at which Consignment, GoodsItem or Package is expected to be or required to delivered at the premises of DeliveryParty.					
<b>LatestDeliveryDate</b>	The latest DateAndTimes at which Consignment, GoodsItem or Package is expected to be or required to be delivered at the premises of DeliveryParty.					
<b>DeadLine</b>	The deadline for giving a response to a request for quote or taking part in an auction. Further the DateAndTimes at which a given Offer is no longer valid.					
<b>Equipment</b>	An artifact necessarily accompanying Consignment or GoodsItem on TransportLeg.			Examples: Exchangable pallet, container, trailer, grid, cage, etc.		
	Category or type of Equipment.	EquipmentType	Code			<b>CN</b> – Container <b>EFP</b> – Europallet

Class name	Semantic description	Attributes	Type	Specifications	Usage	Codes
						<b>RR</b> – Railway wagon <b>PA</b> – Other pallet <b>RF</b> – Flat bed <b>T</b> – Trailer <b>ULD</b> – Air cotainer or pallet <b>RAC</b> - Rack <b>CA</b> - Cage
	Identification of Equipment.	EquipmentId	Text			
	Method through which the identification of Equipment may be automatically read or caught	ReadingMethod	Text			Bar codes 1d, 2d <b>RFID</b> <b>PDF417</b> <b>CODE39</b>
	Size linked to the identification of Equipment (f.ex. 40 feet container)	Size	Text			
	Number of weight units (of the type denoted by unitCode).	Weight	Measure			<b>KGM</b> =kilograms <b>GRM</b> =gram <b>TON</b> =ton
	One of 3 dimensions of Equipment, the other two being Width and Length.	Height	Measure			<b>MTR</b> =Metres <b>DTM</b> =Decimetres <b>CMT</b> =Centimetres
	One of 3 dimensions of Equipment, the other two being Height and	Width	Measure			<b>MTR</b> =Metres <b>DTM</b> =Decimetres

Class name	Semantic description	Attributes	Type	Specifications	Usage	Codes
	Length.					<b>CMT</b> =Centimetres
	One of 3 dimensions of Equipment, the other two being Width and Height.	Length	Measure			<b>MTR</b> =Metres <b>DTM</b> =Decimetres <b>CMT</b> =Centimetres
	Number of volume units (of the type denoted by unitCode).	Volume	Measure			<b>DMQ</b> =cubic decimetres <b>MTQ</b> =cubic metres
<b>Event</b>	Registered (caught) action performed on or registered lack of action due to be performed on Consignment or Package involving a state change for Consignment or Package, or registered current status for Consignment or Package.				TT	<b>Only applicable in TT</b>
	Denoting whether Event is a physical state (condition), an administrative event or an exception/discrepancy.	Type	Code		<b>TT</b>	PhysicalState PhysicalEvent AdministrativeState AdministrativeEvent Deviation
	Event is (or is not) the last relevant Event registered for Consignment or Package	CurrentState	Indicator	Requesting current status and responding to this request.	<b>TT</b>	YES NO



Class name	Semantic description	Attributes	Type	Specifications	Usage	Codes
	Package					
	Definition of Event	EventCode	Code		TT	<a href="http://www.edipro.no/transportXML/codes/eventCodes">www.edipro.no/transportXML/codes/eventCodes</a>
<b>FreeText</b>	Text which may be written freely and without restrictions.					
	The usage intended for Text.	Usage	Code		TT	ICN=Information for Consignee SIC=Sender's instruction to carrier HAN=Handling instructions AAI=General instructions ACD=Reason (valid in TT) TRA=EventDescription(Valid in TT)
	Instruction or other text to be used as specified by FreeText.Usage.	Text	Text		TT	
	Number of characters in FreeText.Text.	NoOfCharacters	Numeric		TT	

Class name	Semantic description	Attributes	Type	Specifications	Usage	Codes
<b>GoodsItem</b>	A collection of Package displaying a set of common characteristics (usually the same GoodsDescription and the same PackageTypeCode).					
	Description of the goods contained in GoodsItem.	GoodsDescription	Text			
	Number of Package contained in GoodsItem.	NumberOfPackages	Quantity			<b>PCE</b> =number
	Type of package or packaging material (f.ex. box, roll, pallet, etc.)	PackageTypeCode	Code			Use UN/CEFACT rec. no. 21
	Total number of weight units (of the type denoted by unitCode) including number of weight units of the packaging material.	GrossWeight	Measure			<b>KGM</b> =kilogram <b>GRM</b> =gram <b>TON</b> =ton
	Total number of volume units (of the type denoted by unitCode).	Volume	Measure			<b>DMQ</b> =cubic decimetres <b>MTQ</b> =cubic metres
	Total number of weight units (of the	NetWeight	Measure			<b>KGM</b> =kilogram

Class name	Semantic description	Attributes	Type	Specifications	Usage	Codes
	type denoted by unitCode) excluding number of weight units of the packaging material.					<b>GRM</b> =gram <b>TON</b> =ton
	Total number of length meters occupied by GoodsItem in MeansOfTransport	LoadingMetres	Measure			<b>MTR</b> =loading metres
	Text, number or other id contributing to the identification of GoodsItem	Marking	Text			
<b>TermsOfDelivery</b>	Terms agreed between goods seller and goods buyer expressed through current INCOTERMS or COMBITERMS, defining a.o. which parties are to pay for which transport services involved.				<b>DNoM</b>	
	Coded indentification of TermsOfDelivery	TODConditionCode	Code		<b>DNoM</b>	
	Denotes whether TermsOfDelivery.TODConditionCod	TODConditionCode List	Code		<b>DNoM</b>	<b>Incoterms</b> =Incoterms 2000

Class name	Semantic description	Attributes	Type	Specifications	Usage	Codes
	e is a COMBITERMS or INCOTERMS code.					<b>Combiterms</b> =Combiterms 1990
	Location in respect to which TermsOfDelivery is defined	TODLocation	Text	Ex: CIP STAVANGER.		
<b>Location</b>	Geographical location or place.					
	Unique identification of Location.	LocationId	Text		<b>TT</b>	
	Denotes which type of identification is employed in Location.LocationId.	IdType	Code	Examples:  140: Skåne, Østlandet 163: Hedmark fylke, Pennsylvania, Shropshire, Norrbotten.	<b>TT</b>	<b>16</b> – Postal code <b>139</b> – Port. <b>140</b> – Mutually agreed geographical area. <b>145</b> – IATA. <b>162</b> – Country. <b>163</b> - Official name of region or area within a country. (state, county,

Class name	Semantic description	Attributes	Type	Specifications	Usage	Codes
						fylke(Norway), etc.). <b>UNLOCODE</b> – UN location code <b>EAN</b> – EAN location no.
	Type of Location (f.ex. terminal, ramp, hub)	LocationType	Code		<b>TT</b>	<a href="http://www.edipro.no/transportXML/codes/locationTypes">http://www.edipro.no/transportXML/codes/locationTypes</a> : Terminal Ramp Hub Pub Gate Port IndustrialArea Address
	The name used to denote Location	LocationName	Text			<b>TT</b>
	Code uniquely identifying the country in which Location.LocationName is localized.	CountryCode	Code			<b>TT</b> ISO 3166-1
<b>StartLocation</b>	Location at which TransportLeg starts.					

Class name	Semantic description	Attributes	Type	Specifications	Usage	Codes
<b>EndLocation</b>	Location at which TransportLeg ends.					
<b>EventLocation</b>	Location at which Event happens or is valid.					
<b>MeansOfTransport</b>	Unit actively contributing to the carriage of Consignment, GoodsItem or Package.			This means that passive carriers like trailers or containers are to be considered as equipment.		
	Identification of MeansOfTransport, f.ex. license plate.	MeansOfTransportId	Text			
	The country in which MeansOfTransport is registered.	CountryCode	Code			ISO 3166-1
	The name of MeansOfTransport, mostly relevant for sea carriers (f.ex. <b>S/S FRYD</b> )	Name (Navn)	Text			
	Type of MeansOfTransport (f.ex. truck with trailer, truck, tugmaster)	MeansOfTransportType	Code			<b>11</b> – ship/boat <b>6</b> – airplane <b>1</b> – truck/truck with trailer

Class name	Semantic description	Attributes	Type	Specifications	Usage	Codes
						<b>14</b> – flat bed  (mutually defined codes may be applied)
<b>Package</b>	A physical unit due to be transported, currently being transported or having been transported containing one or more commodities collected and packed as one physical unit.				<b>DNoM</b>	
	A reference number uniquely identifying Package.	PackageId	Text		<b>DNoM</b> <b>TT</b>	
	Identifies the type of PackageId (f.ex. the Norwegian standard defined by Norsk EDIPRO)	IdType	Code	For Norwegian domestic transport transportXML recommends EANSSCC.	<b>DNoM</b> <b>TT</b>	<b>EANSSCC</b> =Serial Shipping Container Code as defined in Norsk EDIPRO's recommendation "Dokumentasjon i innenlandstransport "  <b>Other</b> ="Default"
	One of 3 dimensions of Package,	Height	Measure		<b>TT</b>	<b>MTR</b> =metres

Class name	Semantic description	Attributes	Type	Specifications	Usage	Codes
	the other two being Width and Length.					<b>DTM</b> =decimeters <b>CMT</b> =centimetres
	One of 3 dimensions of Package, the other two being Height and Length.	Width	Measure		<b>TT</b>	<b>MTR</b> =metres <b>DTM</b> =decimeters <b>CMT</b> =centimetres
	One of 3 dimensions of Package, the other two being Width and Height.	Length	Measure		<b>TT</b>	<b>MTR</b> =metres <b>DTM</b> =decimeters <b>CMT</b> =centimetres
	Type of package or packaging material (coded form)	PackageTypeCode	Code			Use UN/ECE Rec. No. 21
	Type of package or packaging material (textual form)	PackageTypeText	Text			
	Total number of weight units (of the type denoted by unitCode) including number of weight units of the packaging material.	GrossWeight	Measure		<b>DNoM</b> <b>TT</b>	<b>KGM</b> =kilogram <b>GRM</b> =gram <b>TON</b> =ton
	Total number of weight units (of the type denoted by unitCode) excluding number of weight units of the packaging material.	NetWeight	Measure		<b>TT</b>	<b>KGM</b> =kilogram <b>GRM</b> =gram <b>TON</b> =ton



Class name	Semantic description	Attributes	Type	Specifications	Usage	Codes
	Total number of volume units (of the type denoted by unitCode).	Volume	Measure		TT	<b>DMQ</b> =cubic decimetres <b>MTQ</b> =cubic metres
	Total number of length meters occupied by Package in MeansOfTransport	LoadingMetres	Measure		TT	<b>MTR</b> =loading metres
	Text, number or other id contributing to the identification of Package	Marking	Text			
<b>Party</b>	A participant in a transport chain, f.ex. Consignee, Consignor					
	Unique identification of Party.	PartyId	Text		<b>DNoM –(if Name is not included)</b>  TT	
	The type of identification employed in Party.PartyId.	IdType	Code	If the 91 code is used, the information is supposed to be known –	<b>DNoM – (dersom PartyId)</b>	<b>9</b> = EAN location no <b>82</b> =Official company register (Norway:

Class name	Semantic description	Attributes	Type	Specifications	Usage	Codes
	(f.ex. organization no, internal client id, etc.)			otherwise complete party information must be given	<b>TT</b>	Enhetsregisteret in Brønnøysund) <b>87</b> =Defined by transport provider <b>91</b> =Defined by another party (explicitly Consignee's client id in Consignor's system)
	The official (and legal) name of a company, organization or person.	Name	Text		<b>DNoM (if Partyld is not used)</b> <b>TT</b>	
	Unique identification of a company allocated by fiscal or other authorities in a country.  (In Norway: "enhetsnr" for companies and "personnr" for persons)	VATNo	Text		<b>TT</b>	
	Party's client id in the system of the customs authorities.	CustomsCreditNo	Text			
<b>Consignee</b>	The legal receiver of Consignment, GoodsItem or Package.				<b>DNoM</b>	

Class name	Semantic description	Attributes	Type	Specifications	Usage	Codes
<b>DeliveryParty</b>	Party at whose premises Consignment, GoodsItem or Package is to be delivered or has been delivered.				<b>DNoM</b>	
<b>OrderingParty</b>	Party ordering the transport service described by TransportJob.					
<b>FreightPayer</b>	Party who is to pay for (some or all of) the transport services described by TransportJob					
<b>Consignor</b>	The legal sender of Consignment, GoodsItem or Package.				<b>DNoM</b>	
<b>DespatchParty</b>	Party from whose premises Consignment, GoodsItem or Package is to be picked up or despatched or has been picked up or despatched.				<b>DNoM</b>	
<b>CashOnDeliveryPayee</b>	Party who is the receiver of a COD amount, or Party on behalf of whom the delivery restriction is implemented.					

Class name	Semantic description	Attributes	Type	Specifications	Usage	Codes
	Bank id, account type and serial no serialized to form a complete bank account no (in Norway 11 digits)	AccountNo	Text			
	Name of bank	BankName	Text			
	Code identifying CashOnDeliveryPayee.BankName – used in international bank transaction	SwiftCode	Text			
<b>BondedWarehouseParty</b>	Party to whom the customs authorities have allocated a license empowering him to register and store imported goods before it has been customs cleared.					
<b>Forwarder</b>	Party offering transport and other logistics services (f.ex. customs clearance).					
<b>TransportCompany</b>	Party who undertakes and is responsible for parts of or the whole transport of Consignment , GoodsItem or Package.					
<b>Exporter</b>	Goods owner at outbound customs					

Class name	Semantic description	Attributes	Type	Specifications	Usage	Codes
	border crossing.					
<b>Importer</b>	Goods owner at inbound customs border crossing.					
<b>DespatchingAgent</b>	Forwarder who is the sender of manifested cargo in an international transport.					
<b>ReceivingAgent</b>	Forwarder who is the receiver of manifested cargo in an international transport.					
<b>MessageRecipient</b>	Party to whom the current instance of the TransportJob message is addressed.			Normally the message sender and recipient will be addressed in an envelope encompassing the message (f.ex. a SOAP envelope) – we include the possibility of placing this information in the message itself as a backup mechanism.		
<b>MessageSender</b>	Party generating the current instance of the TransportJob message.			See comment above.		

Class name	Semantic description	Attributes	Type	Specifications	Usage	Codes
<b>InvitedTransportCompany)</b>	TransportCompany who is invited to place Offer following a request for quote or opened auction at a transport portal.					
<b>Reason</b>	The reason for or cause of a registered or caught Event.				TT	Only applicable in TT
	Definition of Reason	ReasonCode	Code		TT	Code list:  <a href="http://www.edipro.no/transportXML/codes/reasonCodes">http://www.edipro.no/transportXML/codes/reasonCodes</a>
<b>Reference</b>	Information originating from a transaction other than the current transport job transaction.			Should only be used if the object which is referenced is not part of the current instance of the transport model (ie the message instance which is currently exchanged) – in other words: Reference is a		

Class name	Semantic description	Attributes	Type	Specifications	Usage	Codes
				reference "out of the message".		
	Identification of Reference.	ReferenceNo	Text			
	Type of ReferenceNo (f.ex. Consignor's order no)	IdType	Code			AAO=Consignee's reference CU=Consignor's reference BN=booking number FF=Forwarder's reference ICO=insurance reference SS=seller's reference CO=buyer's reference AEP=FreightPayer's reference
	Reference text.	Text	Text			

Class name	Semantic description	Attributes	Type	Specifications	Usage	Codes
<b>Service</b>	Identification of a product or service offered by TransportCompany or Forwarder (f.ex. groupage, part load, temperature regulated goods, express, cod service etc.)				<b>DNoM</b>	
	Code identifying Service	BasicServiceCode	Code	Ref. TransportCompany/Forwarder.	<b>DNoM</b>	Recommended codes for the Request function:  DoorDoor Express Groupage
	Additional service linked to BasicServiceCode.	AdditionalServiceCode	Code	Ref. TransportCompany/Forwarder.  Repeatable.		
	Priority handling independent of the service definition and only related to other requests/orders made by the sender	Priority	Code			<b>A, B, C.... – A=Highest</b>
	Indicator telling whether Consignment, GoodsItem or Package is to be transported within certain temperature limits.	TemperatureControlledGoodsInd	Indicator			YES NO



Class name	Semantic description	Attributes	Type	Specifications	Usage	Codes
	Highest allowed temperature.	MaxTemperature	Measure	In Celsius		
	Lowest allowed temperature.	MinTemperature	Measure	In Celsius		
	Ideal temerature.	IdealTemperature	Measure	In Celsius		Ex.: Lollipop –17.5 C
<b>Source</b>	Automatic or manual source/instrument used to catch and deliver data.				<b>TT</b>	<b>Only applicable in TT.</b>
	Unique identification of Source (f.ex. MAC address).	SourceId	Text		<b>TT</b>	
	Operator or system employing Source (f.ex. driver).	UsedBy	Text		<b>TT</b>	
	Type of Source. Denotes whether Source is manual (f.ex. driver keying in data) or automatic (f.ex. through scanning or optically reading).	Type	Code		<b>TT</b>	<b>Manual Automatic</b>

Class name	Semantic description	Attributes	Type	Specifications	Usage	Codes
<b>TransportJob</b>	Action initiating transport, or definition of transport service actions.				<b>DNoM</b>	
<b>TransportLeg</b>	Part of the total carriage way delimited by StartLocation and EndLocation.					
	Precarriage (pickup), main transport or postcarriage (distribution).	TransportLegType	Code			<b>10</b> – precarriage <b>20</b> – main transport <b>30</b> – postcarriage
	Identification of TransportLeg.	TransportLegId	Text	F.ex: Identification of a route.		
	Transport mode	ModeOfTransport	Code			<b>10</b> – sea carriage <b>12</b> – railway wagon on board ship <b>16</b> – truck on board ship <b>17</b> – trailer on board ship <b>20</b> – railway <b>23</b> – truck/trailer on railway wagon <b>30</b> - road transport <b>40</b> –air transport <b>50</b> – post

Class name	Semantic description	Attributes	Type	Specifications	Usage	Codes
						<b>60</b> – electronic transmission <b>70</b> – permanent installations <b>80</b> - transport on domestic water (river transport) <b>90</b> – proprietary carriage

### 2.3.1 Generic attribute names in XML message instances:

<b>id</b>  <b>href="#xxx"</b>	<p><b>id</b> is used to identify a <i>basic object</i> internally in a message - <b>href</b> is used in a <i>structure object</i> to refer back to a basic object. Example:</p> <pre>&lt;BasicObject&gt;   &lt;Package id="4"&gt;</pre> <pre>&lt;Structure&gt;   &lt;ConsignmentStructure&gt;     &lt;Package href="#4"&gt;</pre>
<b>subClass</b>	<p>Used to identify a subclass. The name of the superclass is the XML element name (the tag). Example:</p> <pre>&lt;Location subClass="StartLocation"&gt;</pre>
<b>auxiliary attributes in core component types</b>	<p>Attributes conveying necessary additional information to the element content itself, f.ex. measurement unit, currency code, code list identifier, etc. The attributes are listed in the Core Component Types section in chapter 1.6. They are also defined through core component type definitions in the XSD schemas. The auxiliary attributes are not syntactically mandatory.</p> <p>Example:</p> <pre>&lt;GrossAmount currencyIdentificationCode="SEK"&gt;</pre>

### 2.3.2 Attributes in the root element of an XML message instance:

<b>version</b>	Version. Per October 2002 = "3.0".
<b>definedBy</b>	Must = "Norsk EDIPRO".
<b>domain</b>	The attribute is included in order to identify the domain model referred to by this message instance. Value should be = "transportXML".
<b>collaboration</b>	Identifies one of the collaborations defined by transportXML (in version 3.0 14 collaborations have been defined – each of them is documented in a separate activity model).
<b>messageFunction</b>	Denotes the message function. This should be one of the identifiers which are specified as the name of a transition from one participant to another in one of the defined activity models in transportXML (collaborations 1..14)
<b>profile</b>	Denotes a syntactic subset of TransportJob or TrackAndTrace. In version 3.0 3 profiles have been defined: DomesticNO (domestic profile), DomesticNOMin (minimal mandatory domestic solution profile) and TransportPortal (market place profile).
<b>messageId</b>	Unique id for the message within the current information exchange

	setting.
<b>updateIndicator</b>	<p>Denotes whether this is an original message, an update message or an instruction to delete a message. Only mandatory if the value is different from Original. Values:</p> <p>Original Update Deletion</p>
<b>testIndicator</b>	<p>YES=This is a test message NO=This is a production message (not mandatory if value=NO)</p>

Example of a TransportJob root element:
<pre> &lt;TransportJob   version="3.0"   definedBy="Norsk EDIPRO"   domain="transportXML"   collaboration="Instruction"   messageFunction="InstructionConfirmation"   profile="DomesticNO"   messageId="1001"&gt; </pre>

## **2.4 Transport Job – Domain modell: Semantic descriptions of relationships**

Class 1	Card.	Class 2	Card.	Name of relationship	Semantic description of relationship
Consignment	0..*	Cod	0..1	MayOnlyBeDeliveredAgainst	Consignment is only to be delivered if the delivery restriction conditions of the Cod object is satisfied (f.ex. that a cod amount has been paid by the consignee).
GoodsItem	0..*	Cod	0..1	MayOnlyBeDeliveredAgainst	GoodsItem is only to be delivered if the delivery restriction conditions of the Cod object is satisfied.
Package	0..*	Cod	0..1	MayOnlyBeDeliveredAgainst	Package is only to be delivered if the delivery restriction conditions of the Cod object is satisfied.
COD	1	CashOnDeliveryPayee	1	MustBeSatisfiedOnBehalfOf	The conditions of the Cod object must be satisfied on behalf of CashOnDeliveryPayee.
Consignment	0..*	Consignor	0..*	HasAsLegalSender	Consignor is the legal sender of Consignment.
Consignment	0..*	DespatchParty	0..*	IsPickedUpAtAddress	Consignment is to be or has been picked up at PhysicalAddress belonging to DespatchParty. If DespatchParty's cardinality = 0, Consignment is to be or has been picked up at PhysicalAddress belonging to Consignor
Consignment	0..*	FreightPayer	0..*	IsInvoicedTo	Rendered transport and other services related to Consignment is to be invoiced to FreightPayer
Consignment	0..*	OrderingParty	0..1	IsPerformedFor	Transport and other services related to Consignment are performed at the order of OrderingParty
Consignment	0..*	BondedWarehouseParty	0..1	IsTakenOutFrom	Consignment previously placed into a bonded warehouse of BondedWarehouseParty is to be transitioned from or has been transitioned from this warehouse.

Class 1	Card.	Class 2	Card.	Name of relationship	Semantic description of relationship
Consignment	0..*	BondedWarehouseParty	0..1	IsPlacedInto	Consignment is to be or has been transitioned into BondedWarehouseParty's bonded warehouse.
Consignment	0..*	Consignee	0..*	HasAsLegalConsignee	Consignee is the legal receiver of Consignment
Consignment	0..*	DeliveryParty	0..*	DeliveredAtAddresses	Consignment is to be or has been delivered at PhysicalAddress belonging to DeliveryParty. If DeliveryParty's cardinality = 0, Consignment is to be or has been delivered at PhysicalAddress belonging to Consignee
Consignment	0..*	TransportCompany	0..*	IsTransportedBy	Consignment is transported by TransportCompany
Consignment	0..*	Importer	0..1	HasGoodsOwnerAtImport	Consignment has Importer as the legal commodity owner when Consignment passes the inbound customs border.
Consignment	0..*	Exporter	0..1	HasGoodsOwnerAtExport	Consignment has Exporter as the legal commodity owner when Consignment passes the outbound customs border.
Consignment	0..*	DespatchingAgent	0..*	IsDespatchedViaForwarder	Consignment is despatched via – and normally consolidated and manifested by – the forwarder identified by DespatchingAgent
Consignment	0..*	ReceivingAgent	0..*	IsReceivedViaForwarder	Consignment is received via – and normally deconsolidated by - the forwarder identified by ReceivingAgent
InvitedTransportCompany	0..*	Offer	0..*	IsInvitedToPlace	Invitation to place Offer is issued to or applicable for InvitedTransportCompany
Consignment	0..*	Forwarder	0..*	HasInternationalLogisticsServicesPerformedBy	Customs declaration, export clearance, transit handling, consolidation and/or other



Class 1	Card.	Class 2	Card.	Name of relationship	Semantic description of relationship
					logistics services related to Consignment are performed by Forwarder for an international transport.
Package	0..*	Consignor	0..*	HasAsLegalSender	Consignor is the legal sender of Package.
Package	0..*	DespatchParty	0..*	IsPickedUpAtAddress	Package is to be or has been picked up at PhysicalAddress belonging to DespatchParty. If DespatchParty's cardinality = 0, Package is to be or has been picked up at PhysicalAddress belonging to Consignor
Package	0..*	FreightPayer	0..*	IsInvoicedTo	Rendered transport and other services related to Package is to be invoiced to FreightPayer
Package	0..*	OrderingParty	0..1	IsPerformedFor	Transport and other services related to Package are performed at the order of OrderingParty
Package	0..*	BondedWarehouseParty	0..1	IsTakenOutFrom	Package previously placed into a bonded warehouse of BondedWarehouseParty is to be transitioned from or has been transitioned from this warehouse.
Package	0..*	BondedWarehouseParty	0..1	IsPlacedInto	Package is to be or has been transitioned into BondedWarehouseParty's bonded warehouse.
Package	0..*	Consignee	0..*	HasAsLegalConsignee	Consignee is the legal receiver of Package.
Package	0..*	DeliveryParty	0..*	DeliveredAtAddresses	Package is to be or has been delivered at PhysicalAddress belonging to DeliveryParty. If DeliveryParty's cardinality = 0, Package is to be or has been delivered at PhysicalAddress belonging to Consignee
Package	0..*	TransportCo	0..*	IsTransportedBy	Package is transported by

Class 1	Card.	Class 2	Card.	Name of relationship	Semantic description of relationship
		Company			TransportCompany
Package	0..*	Importer	0..1	HasGoodsOwnerAt Import	Package has Importer as the legal commodity owner when Package passes the inbound customs border.
Package	0..*	Exporter	0..1	HasGoodsOwnerAt Export	Package has Exporter as the legal commodity owner when Package passes the outbound customs border.
Package	0..*	Despatching Agent	0..*	IsDespachedViaForwarder	Package is despatched via – and normally consolidated and manifested by – the forwarder identified by DespatchingAgent
Package	0..*	ReceivingAgent	0..*	IsReceivedViaForwarder	Package is received via – and normally deconsolidated by - the forwarder identified by ReceivingAgent
Package	0..*	Forwarder	0..*	HasInternationalLogisticsServicesPerformedBy	Customs declaration, export clearance, transit handling, consolidation and/or other logistics services related to Package are performed by Forwarder for an international transport.
GoodsItem	0..*	Consignor	0..*	HasAsLegalSender	Consignor is the legal sender of GoodsItem.
GoodsItem	0..*	DespatchParty	0..*	IsPickedUpAtAddress	GoodsItem is to be or has been picked up at PhysicalAddress belonging to DespatchParty. If DespatchParty's cardinality = 0, GoodsItem is to be or has been picked up at PhysicalAddress belonging to Consignor
GoodsItem	0..*	FreightPayer	0..*	IsInvoicedTo	Rendered transport and other services related to GoodsItem is to be invoiced to FreightPayer
GoodsItem	0..*	OrderingParty	0..1	IsPerformedFor	Transport and other services related to GoodsItem are performed at the order of OrderingParty

Class 1	Card.	Class 2	Card.	Name of relationship	Semantic description of relationship
GoodsItem	0..*	BondedWarehouseParty	0..1	IsTakenOutFrom	GoodsItem previously placed into a bonded warehouse of BondedWarehouseParty is to be transitioned from or has been transitioned from this warehouse.
GoodsItem	0..*	BondedWarehouseParty	0..1	IsPlacedInto	GoodsItem is to be or has been transitioned into BondedWarehouseParty's bonded warehouse.
GoodsItem	0..*	Consignee	0..*	HasAsLegalConsignee	Consignee is the legal receiver of GoodsItem.
GoodsItem	0..*	DeliveryParty	0..*	DeliveredAtAddresses	GoodsItem is to be or has been delivered at PhysicalAddress belonging to DeliveryParty. If DeliveryParty's cardinality = 0, GoodsItem is to be or has been delivered at PhysicalAddress belonging to Consignee
GoodsItem	0..*	TransportCompany	0..*	IsTransportedBy	GoodsItem is transported by TransportCompany
GoodsItem	0..*	Importer	0..1	HasGoodsOwnerAtImport	GoodsItem has Importer as the legal commodity owner when GoodsItem passes the inbound customs border.
GoodsItem	0..*	Exporter	0..1	HasGoodsOwnerAtExport	GoodsItem has Exporter as the legal commodity owner when GoodsItem passes the outbound customs border.
GoodsItem	0..*	DespatchingAgent	0..*	IsDespatchedViaForwarder	GoodsItem is despatched via – and normally consolidated and manifested by – the forwarder identified by DespatchingAgent
GoodsItem	0..*	ReceivingAgent	0..*	IsReceivedViaForwarder	GoodsItem is received via – and normally deconsolidated by - the forwarder identified by ReceivingAgent
GoodsItem	0..*	Forwarder	0..*	HasInternationalLogisticsServicesPerformedBy	Customs declaration, export clearance, transit handling, consolidation and/or other

Class 1	Card.	Class 2	Card.	Name of relationship	Semantic description of relationship
					logistics services related to GoodsItem are performed by Forwarder for an international transport.
Offer	0..*	Consignment	0..*	IsIssuedFor	Offer has been or is issued for Consignment
Offer	0..*	GoodsItem	0..1	IsIssuedFor	Offer has been or is issued for GoodsItem
Offer	0..*	Package	0..1	IsIssuedFor	Offer has been or is issued for Package
TransportJob	0..1	Consignment	0..*	Concerns	The function (message function) defined by TransportJob, pertains to Consignment.
TransportJob	0..1	GoodsItem	0..*	Concerns	The function (message function) defined by TransportJob, pertains to GoodsItem.
TransportJob	0..1	Package	0..*	Concerns	The function (message function) defined by TransportJob, pertains to Package.
Party	0..*	Contact	0..*	MaybeContactedThrough	Party may be contacted through Contact
Party	1	Address.PhysicalAddress	1	IsLocatedAt	Party is physically located at Address
Party	1	Address.PostalAddress	1	ReceivesHisMailAt	Address specifies how mail destined for Party is to be addressed.
Party	0..1	FreeText	0..*	IsTheSubjectOf	FreeText gives information related to Party.
Party	0..1	Reference	0..*	RefersTo	An external object identified by Reference is related to Party.
GoodsItem	0..*	Consignment	0..1	IsPartOf	GoodsItem is part of Consignment.
Package	0..*	Consignment	0..1	IsPartOf	Package is part of Consignment.

Class 1	Card.	Class 2	Card.	Name of relationship	Semantic description of relationship
Package	0..*	GoodsItem	0..1	IsPartOf	Package is part of GoodsItem.
Equipment	0..*	Consignment	0..1	IsUsedWhenTransporting	Equipment is used when Consignment is transported or handled.
Equipment	0..*	GoodsItem	0..1	IsUsedWhenTransporting	Equipment is used when GoodsItem is transported or handled.
Equipment	0..*	Package	0..1	IsUsedWhenTransporting	Equipment is used when Package is transported or handled.
Consignment	0..*	TransportLeg	1..*	IsTransportedOn	Consignment is transported from TransportLeg.StartLocation to TransportLeg.EndLocation
GoodsItem	0..*	TransportLeg	1..*	IsTransportedOn	GoodsItem is transported from TransportLeg.StartLocation to TransportLeg.EndLocation
Package	0..*	TransportLeg	1..*	IsTransportedOn	Package is transported from TransportLeg.StartLocation to TransportLeg.EndLocation
MeansOfTransport	0..1	TransportLeg	1	IsActiveCarrierOn	Consignment or GoodsItem or Package is transported by MeansOfTransport on TransportLeg
TransportLeg	1	Location.StartLocation	1	StartsAt	TransportLeg starts at StartLocation
TransportLeg	1	Location.EndLocation	1	EndsAt	TransportLeg ends at EndLocation
MeansOfTransport	1..*	TransportCompany	1	IsAtTheDisposalOf	MeansOfTransport is at the disposal of TransportCompany for transport identified by (the relationship to) TransportLeg.
TransportLeg	1	DespatchParty	0..1	StartsAt	TransportLeg starts at PhysicalAddress belonging to DespatchParty.
TransportLeg	1	DeliveryParty	0..1	EndsAt	TransportLeg ends at

Class 1	Card.	Class 2	Card.	Name of relationship	Semantic description of relationship
					PhysicalAddress belonging to DeliveryParty.
Consignment	0..*	DangerousGoods	0..1	Contains	Consignment contains goods which are classified and declared by DangerousGoods, and this declaration may not be broken down on smaller goods units (goods item, package)
GoodsItem	0..*	DangerousGoods	0..1	Contains	GoodsItem contains goods which are classified and declared by DangerousGoods, and this declaration may not be broken down on package level.
Package	0..*	DangerousGoods	0..1	Contains	GoodsItem contains goods which are classified and declared by DangerousGoods.
Consignment	0..*	LoadingDate	0..*	IsLoadedAtPointInTime	Consignment is loaded on MeansOfTransport on the given LoadingDate.
Consignment	0..*	DeliveryDate	0..*	IsDeliveredAtPointInTime	Consignment is expected to be delivered or has been delivered on DeliveryDate.
Consignment	0..*	ArrivalDate	0..*	ArrivesAtPointInTime	Consignment arrives denoted location on ArrivalDate.
Consignment	0..*	PickupDate	0..*	IsPickedUpAtPointInTime	Consignment is expected to be picked up or has been picked up on PickupDate.
Consignment	0..*	EarliestPickupDate	0..*	IsNotToBePickedUpUntil	Consignment is not to be picked up until EarliestPickupDate.
Consignment	0..*	LatestPickupDate	0..*	IsToBePickedUpBefore	Consignment must be picked up before or on LatestPickupDate.
Consignment	0..*	EarliestDeliveryDate	0..*	IsNotToBeDeliveredUntil	Consignment is not to be delivered until EarliestDeliveryDate.

Class 1	Card.	Class 2	Card.	Name of relationship	Semantic description of relationship
Consignment	0..*	LatestDeliveryDate	0..*	IsToBeDeliveredBefore	Consignment must be delivered before or on LatestDeliveryDate.
TransportJob	1	MessageDate	0..1	IsGeneratedAtPointInTime	TransportJob is generated on MessageDate.
Package	0..*	LoadingDate	0..*	IsLoadedAtPointInTime	Package is loaded on MeansOfTransport on the given LoadingDate.
Package	0..*	DeliveryDate	0..*	IsDeliveredAtPointInTime	Package is expected to be delivered or has been delivered on DeliveryDate.
Package	0..*	ArrivalDate	0..*	ArrivesAtPointInTime	Package arrives denoted location on ArrivalDate.
Package	0..*	PickupDate	0..*	IsPickedUpAtPointInTime	Package is expected to be picked up or has been picked up on PickupDate.
Package	0..*	EarliestPickupDate	0..*	IsNotToBePickedUpUntil	Package is not to be picked up until EarliestPickupDate.
Package	0..*	LatestPickupDate	0..*	IsToBePickedUpBefore	Package must be picked up before or on LatestPickupDate.
Package	0..*	EarliestDeliveryDate	0..*	IsNotToBeDeliveredUntil	Package is not to be delivered until EarliestDeliveryDate.
Package	0..*	LatestDeliveryDate	0..*	IsToBeDeliveredBefore	Package must be delivered before or on LatestDeliveryDate.
GoodsItem	0..*	LoadingDate	0..*	IsLoadedAtPointInTime	GoodsItem is loaded on MeansOfTransport on the given LoadingDate.
GoodsItem	0..*	DeliveryDate	0..*	IsDeliveredAtPointInTime	GoodsItem is expected to be delivered or has been delivered on DeliveryDate.
GoodsItem	0..*	ArrivalDate	0..*	ArrivesAtPointInTime	GoodsItem arrives denoted location on ArrivalDate.
GoodsItem	0..*	PickupDate	0..*	IsPickedUpAtPointInTime	GoodsItem is expected to be picked up or has been picked up on PickupDate.

Class 1	Card.	Class 2	Card.	Name of relationship	Semantic description of relationship
GoodsItem	0..*	EarliestPickupDate	0..*	IsNotToBePickedUpUntil	GoodsItem is not to be picked up until EarliestPickupDate.
GoodsItem	0..*	LatestPickupDate	0..*	IsToBePickedUpBefore	GoodsItem must be picked up before or on LatestPickupDate.
GoodsItem	0..*	EarliestDeliveryDate	0..*	IsNotToBeDeliveredUntil	GoodsItem is not to be delivered until EarliestDeliveryDate.
GoodsItem	0..*	LatestDeliveryDate	0..*	IsToBeDeliveredBefore	GoodsItem must be delivered before or on LatestDeliveryDate.
Offer	0..*	DeadLine	0..1	MustBeGivenBefore	Offer relating to Consignment or GoodsItem or Package must be given before or on DeadLine.
Consignment	0..1	FreeText	0..*	IsTheSubjectOf	FreeText gives information related to Consignment.
Consignment	0..1	Reference	0..*	RefersTo	An external object identified by Reference is related to Consignment.
Package	0..1	FreeText	0..*	IsTheSubjectOf	FreeText gives information related to Package.
Package	0..1	Reference	0..*	RefersTo	An external object identified by Reference is related to Package.
Consignment	0..*	TermsOfDelivery	0..1	IsPerformedAccordingTo	Transport and other services related to Consignment are performed and charged in accordance with TermsOfDelivery.
Package	0..*	TermsOfDelivery	0..1	IsPerformedAccordingTo	Transport and other services related to Package are performed and charged in accordance with TermsOfDelivery.
GoodsItem	0..*	TermsOfDelivery	0..1	IsPerformedAccordingTo	Transport and other services related to GoodsItem are performed and charged in accordance with TermsOfDelivery.



Class 1	Card.	Class 2	Card.	Name of relationship	Semantic description of relationship
ChargeItem	1..*	Charges	1	IsContainedIn	ChargeItem is a debit item which is accumulated into the total freight cost of FreightPayer as Charges.
Consignment	0..*	Charges	0..*	IsDebitedWith	Transport and other services performed related to one or more Consignment has a total cost represented by Charges debitable to FreightPayer.
Package	0..*	Charges	0..*	IsDebitedWith	Transport and other services performed related to one or more Package has a total cost represented by Charges debitable to FreightPayer.
GoodsItem	0..*	Charges	0..*	IsDebitedWith	Transport and other services performed related to one or more GoodsItem has a total cost represented by Charges debitable to FreightPayer.
Offer	0..1	Charges	0..*	IsGivenAtPrice	Offer is given by TransportCompany based on the price specified by Charges and containing ChargeLine objects.
Consignment	0..*	Service	0..*	IsTransportedInAccordanceWith	Transport and other services related to Consignment are performed in accordance with the Service definition.
Package	0..*	Service	0..*	IsTransportedInAccordanceWith	Transport and other services related to Package are performed in accordance with the Service definition.
GoodsItem	0..*	Service	0..*	IsTransportedInAccordanceWith	Transport and other services related to GoodsItem are performed in accordance with the Service definition.
Consignment	0..*	CustomsInformation	0..*	HasCustomsInformation	CustomsInformation pertains to Consignment
CustomsInformation	1	CommercialInvoice	1..*	LinksConsignmentsOrGoodsItemsOrPackagesTo	Commodities transported as Consignment or GoodsItem or Package has been sold as specified in

Class 1	Card.	Class 2	Card.	Name of relationship	Semantic description of relationship
					CommercialInvoice. CommercialInvoice is the basis for the customs line specification and calculation of customs value at customs clearance..
CustomsInformation	1	CustomsLineItem	0..*	HasCustomsLine	CustomsLine is a customs item when the related Consignment or GoodsItem or Package is customs cleared.
Exporter	1	CustomsInformation	1..*	IsGoodsOwnerByExportOfGoodsRelatedTo	Exporter is the goods owner by export of the related Consignment or GoodsItem or Package and is denoted as such at export clearance.
Importer	1	CustomsInformation	1..*	IsGoodsOwnerByImportOfGoodsRelatedFrom	Importer is the goods owner by import of the related Consignment or GoodsItem or Package and is denoted as such at import customs declaration.
Forwarder	0..1	CustomsInformation	0..*	PerformsCustomsClearanceRelatedTo	Forwarder performs import customs clearance (if the related Consignment or GoodsItem or Package is being imported) or outbound export clearance (if the related Consignment or GoodsItem or Package is being exported).

## 2.5 Track & Trace – Domain model: Semantic descriptions of relationships.

Class 1	Card.	Class 2	Card.	Name of relationship	Semantic description of relationship
Event	1..*	Consignment	0..*	IsStateOrEventFor	Event is a caught state or event for Consignment.
Event	1..*	Package	0..*	IsStateOrEventFor	Event is a caught state or event for Package.
Event	1..*	(subclass of) Party	0..*	PertainsToPartyRole	Event is directly related to a specific role which is played by Party (f.ex. Consignee's signed receipt for Consignment)
Reason	0..1	Event	1	IsCauseOf	Reason is the cause of Event
Reason	0..1	Reason	0..1	IsCauseOf	Reason is the cause of Reason ("reason chain")
Source	0..*	Event	1	IsSourceOf	Source is the instrument used to catch or read Event.
Action	0..*	Event	1	IsFollowUpOf	Action is a reported or recommended follow-up action for Event (possibly intended to correct the registered exception)
Event	0..*	FreeText	0..*	IsFurtherDescribed By	A further description of Event in FreeText form.
Reason	0..*	FreeText	0..*	IsFurtherDescribed By	A further description of Reason in FreeText form.
Action	0..*	FreeText	0..*	IsFurtherDescribed By	A further description of Action in FreeText form.
Event	1	EventLocation	0..1	IsRegisteredAtPlace	Event has been registered as pertaining to EventLocation.
Event	1..*	EventDate	0..*	IsValidForPointInTime	Event is valid for the point in time denoted by EventDate or the interval denoted by

Class 1	Card.	Class 2	Card.	Name of relationship	Semantic description of relationship
					EventDate.
MessageDate	0..1	TrackAndTrace	1	IsGeneratedAtPointInTime	The current message instance of TrackAndTrace has been generated on MessageDate.
Party	1	Address	0..*	HasAddress	Party has address = Address, and this address is relevant for the Event which is requested or reported.
Contact	0..*	Party	1	IsSigningOnBehalfOf	Concerns the events DeliveryAtDeliveryParty and ProvedDeliveryAtDeliveryParty: Contact signs to verify the receipt of Consignment or Package on behalf of Party.
(subclass of) Party	0..1	Consignment	0..*	Is<subclass>forConsignment	Party has a specific role directly related to Event being requested or reported for Consignment.
(subclass of) Party	0..1	Package	0..*	Is<subclass>forPackage	Party has a specific role directly related to Event being requested or reported for Package.

## 2.6 Domain model: Code lists

**chargeCodes** (<http://www.edipro.no/transportXML/codes/chargeCodes>)

Code	Explanation
PreCost	Precarriage costs abroad
Precarriage	Precarriage
MainCarriage	Main transport
PostCarriage	Postcarriage
Terminal	Terminal handling
CustClearance	Customs clearance
DangGoods	Dangerous goods fee
Special	Special goods fee
Distribution	Distribution
Private	Fee for private delivery
Exchange	Currency exchange fee
Fuel	Fuel charge
COD	COD fee
Postage	Postage and papers
ExpClearance	Export clearance

## **codTypeCodes**

(<http://www.edipro.no/transportXML/codes/codTypeCodes>)

Code	Explanation
CashOnDeliveryGoodsValue	Cash on Delivery – goods value
CashOnDeliveryFreightValue	Cash on Delivery – freight cost
CashOnDeliveryPaymentDocToPrint	Freigh COD – payment document to be printed
CashOnDeliveryPaymentDocAttached	Freigh COD – payment document attached
CashAgainstDocuments	Cash against Documents
CashAgainstDocumentsNetFreight	Cash against Documents – net freight
CashAgainstDocumentsGrossFreight	Cash against Documents – gross freight
LightlySalted	Lightly salted

## eventCodes

(<http://www.edipro.no/transportXML/codes/eventCodes>)

PhysicalState codes	Explanation / Comment
ReadyForPickup	
LoadedAtDespatchParty	
ReceivedAtTerminal	
ObservedAtLocation	Location to be given
ReloadedAtLocation	
LoadedForTransport	
LoadedForDistribution	
DeliveredAtDeliveryParty	
ProvedDeliveryAtDeliveryParty	The message sender gives by this a legally valid proof of delivery
PlacedInStock	

AdministrativeState codes	Explanation / Comment
OutboundCustomsCleared	
InboundCustomsCleared	
Invoiced	
Paid	
ClaimRegistered	
ClaimHandled	

## reasonCodes

(<http://www.edipro.no/transportXML/codes/reasonCodes>)

Note that this code list contains **reason codes** (applicable for the ReasonCode attribute) as well as **exception codes** (applicable for the EventCode attribute).

Code	Explanation / Comment
Other	
UnauthorizedPerson	
PremisesClosed	
PartyUnavailable	
PartyUnknown	
AddressIncomplete	
AddressUnknown	
AwaitingClearance	
AwaitingPayment	
Stolen	
Damaged	
DamagedStopped	
NotDeliverable	
LeftOverAtTerminal	
TemperatureConstraintsViolated	
DamagedNotForward	
DamagedStopped	
DamagedBroughtForward	
DamagedPackaging	
ShortShipped	The whole of the reported object is missing
MissingItem	"Manko" – a part of the reported object is missing; f.ex. one package in a consignment.



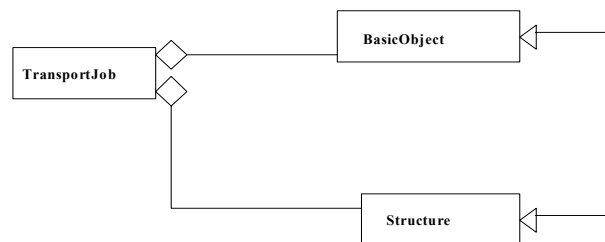
Extraltem	Surplus item or items in the reported object; f.ex. extra packages as compared to manifested or advised consignment information.
Returned	
NotLoadable	
NotDeliverable	
ForwardedToWrongDestination	
LeftOverAtTerminal	
PlacedInStock	
Delayed	
DeliveryRefused	

### 3 Message model - TransportJob

#### 3.1 Message model for TransportJob

##### 3.1.1 Message model TransportJob – High level view of BasicObject – Structure

The message model of TransportJob describes the structure of the XML message type TransportJob. Together with the corresponding XSD schema for TransportJob the message model gives the formal definition of the syntax of the message type.

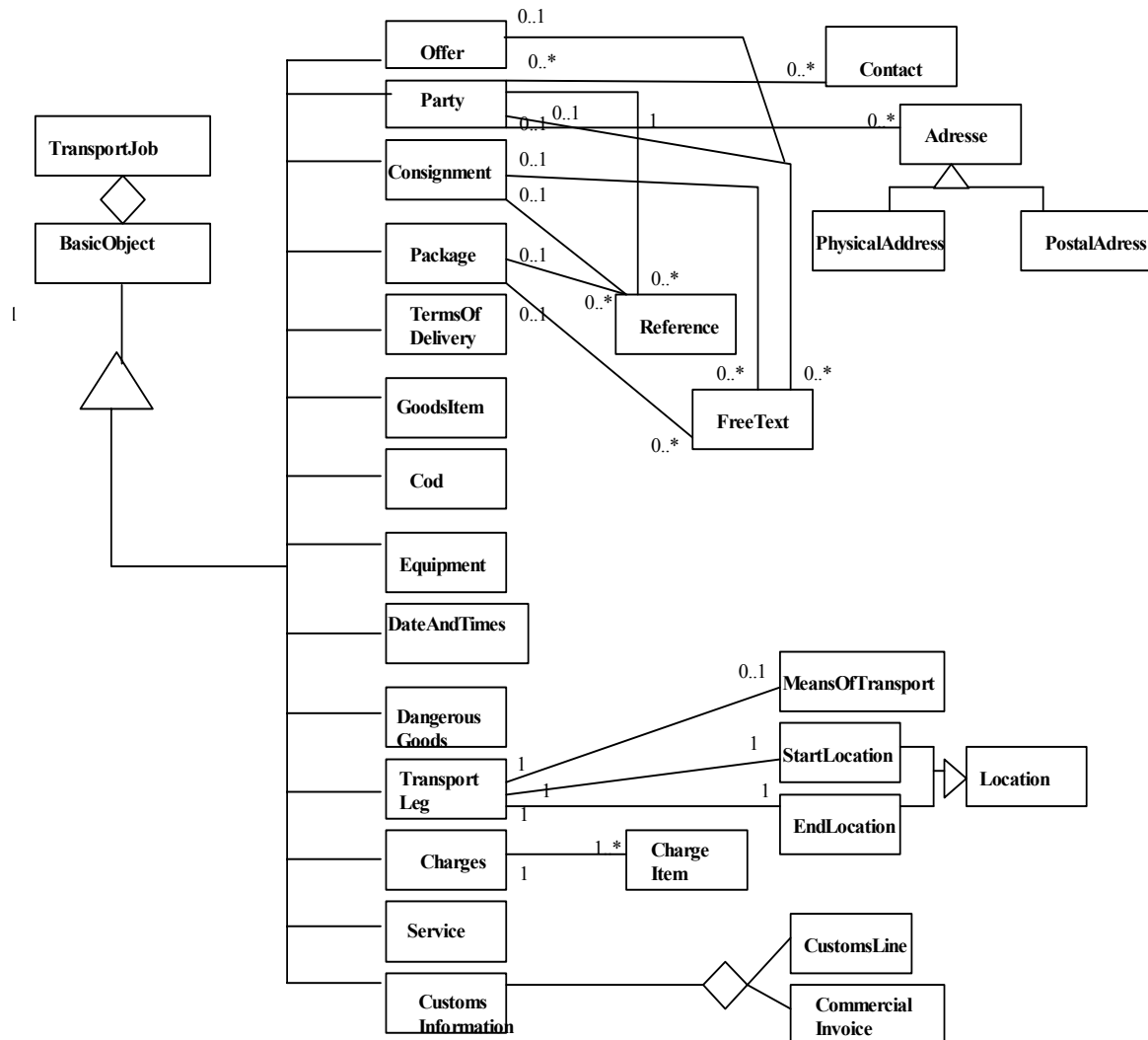


The message model consists of **a BasicObject part** and **a Structure part**.

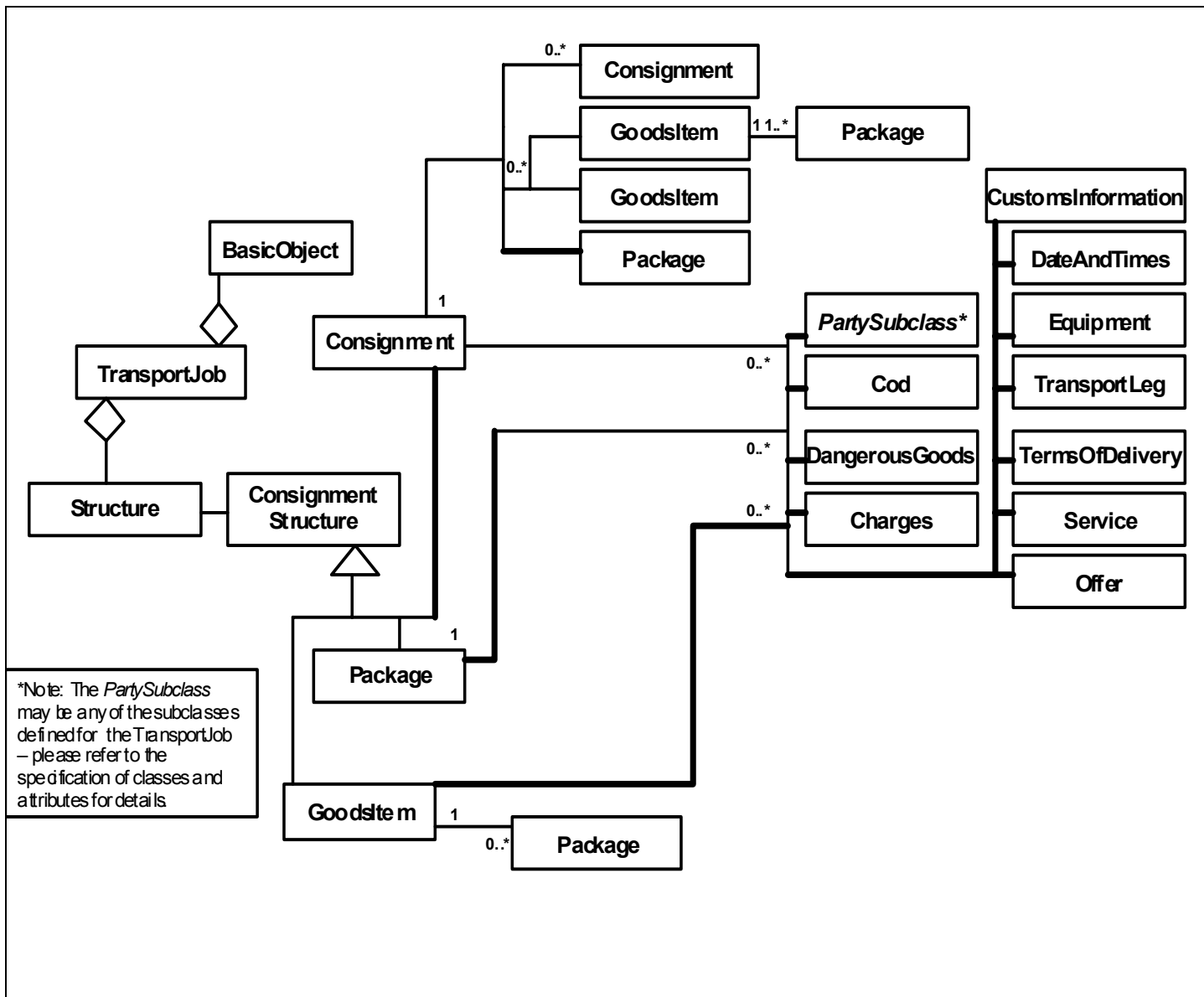
For the Structure part the following rule applies:

- Each structure object (TransportStructure, EquipmentStructure, ConsignmentStructure, etc) is optional and may exist only once.
- Root objects in the respective structures (TransportLeg, Equipment, Consignment, Package, etc) are mandatory and may be repeated as many times as are necessary (one EquipmentStructure object may in other words "consist of" one or more equipment structures).

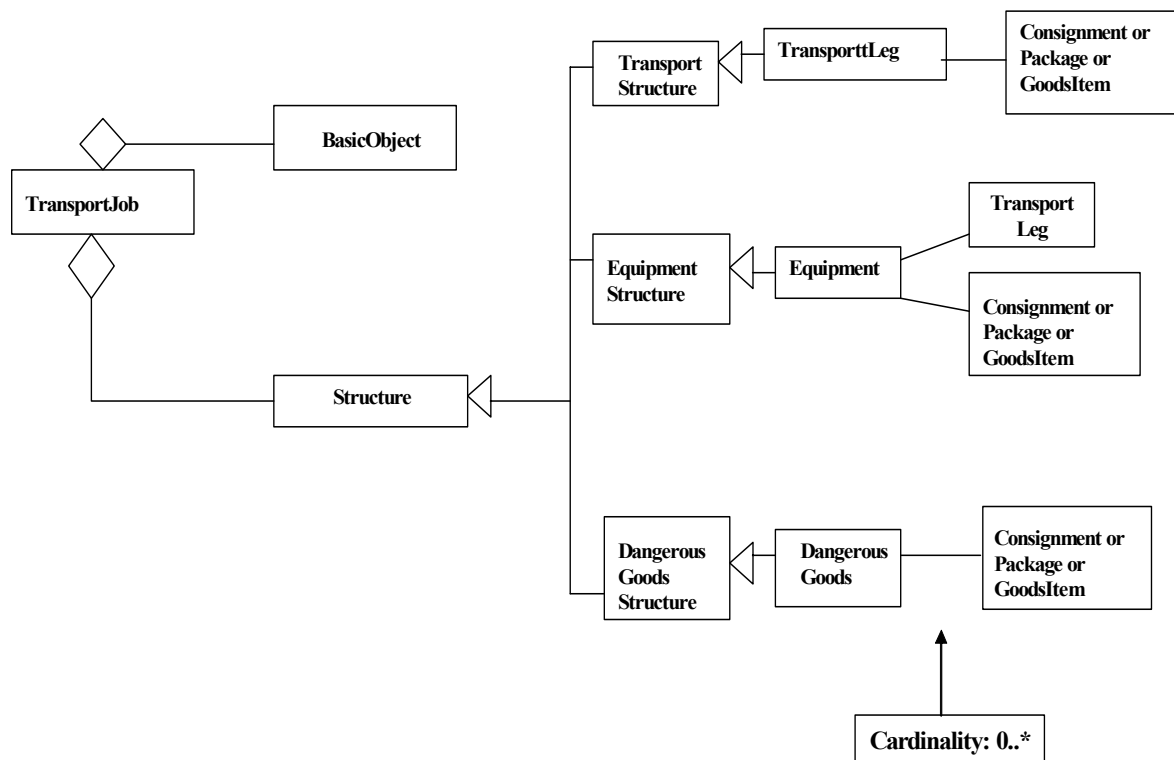
### 3.1.2 Message model TransportJob - BasicObject



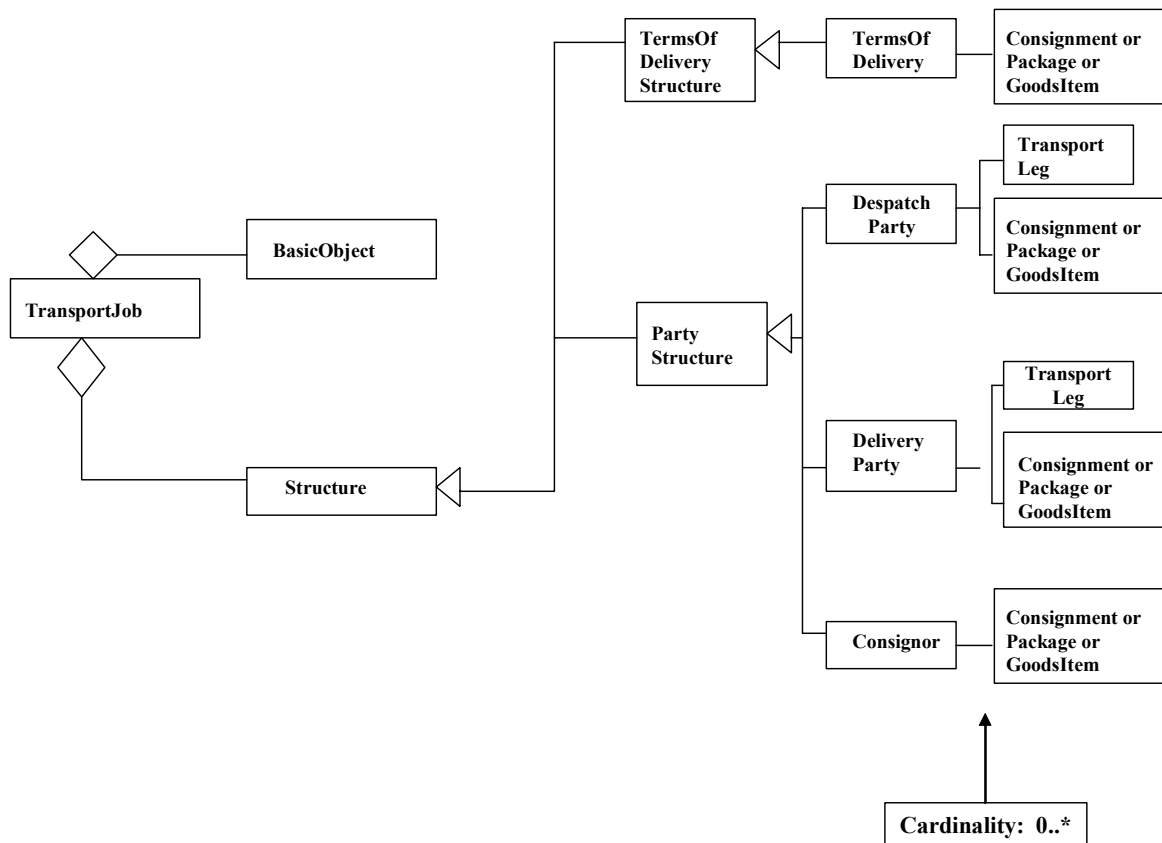
### 3.1.3 Message model TransportJob - Structure part 1: Consignment



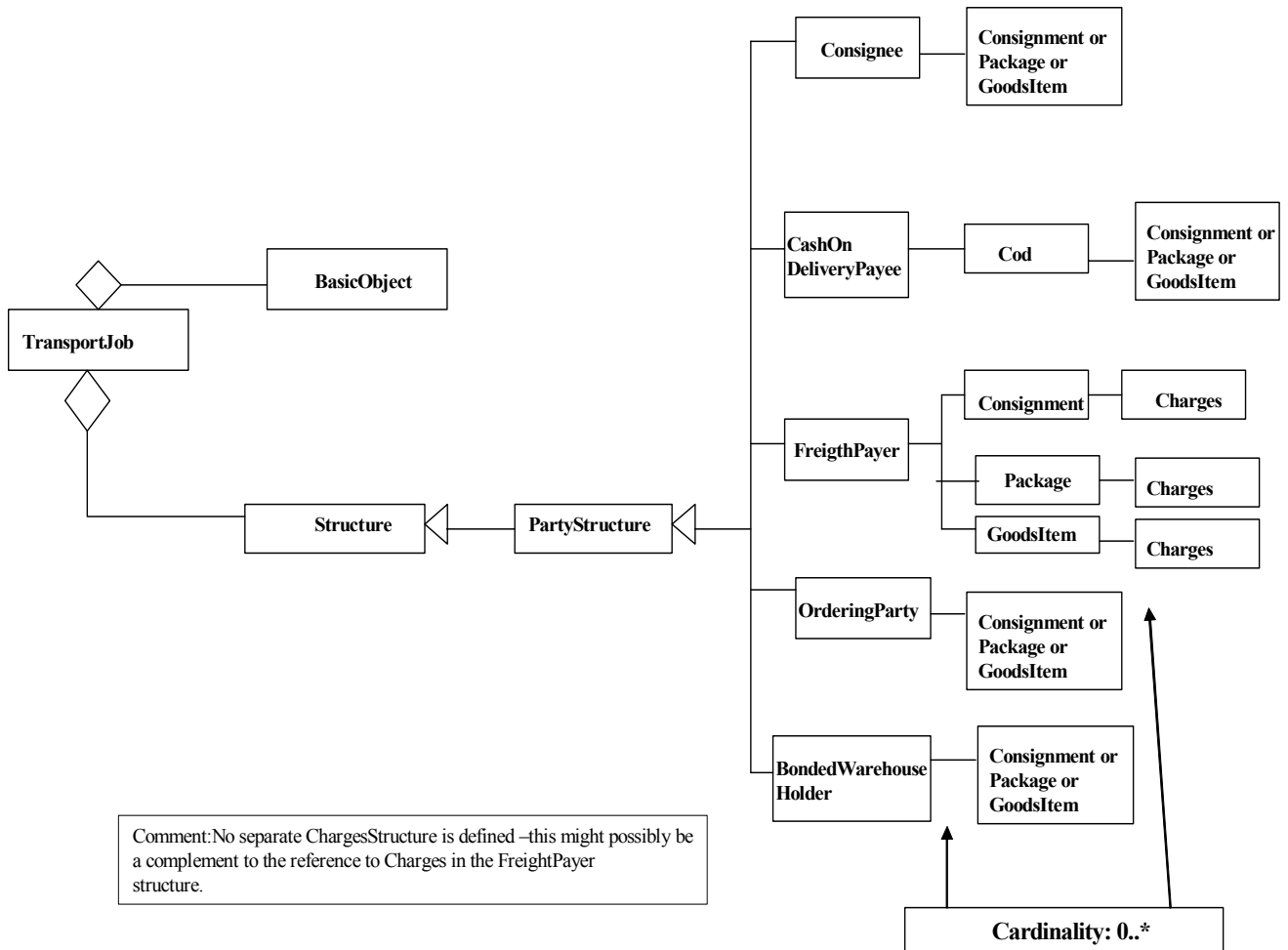
### 3.1.4 Message model TransportJob - Structure part 2: Transport, Equipment, DangerousGoods



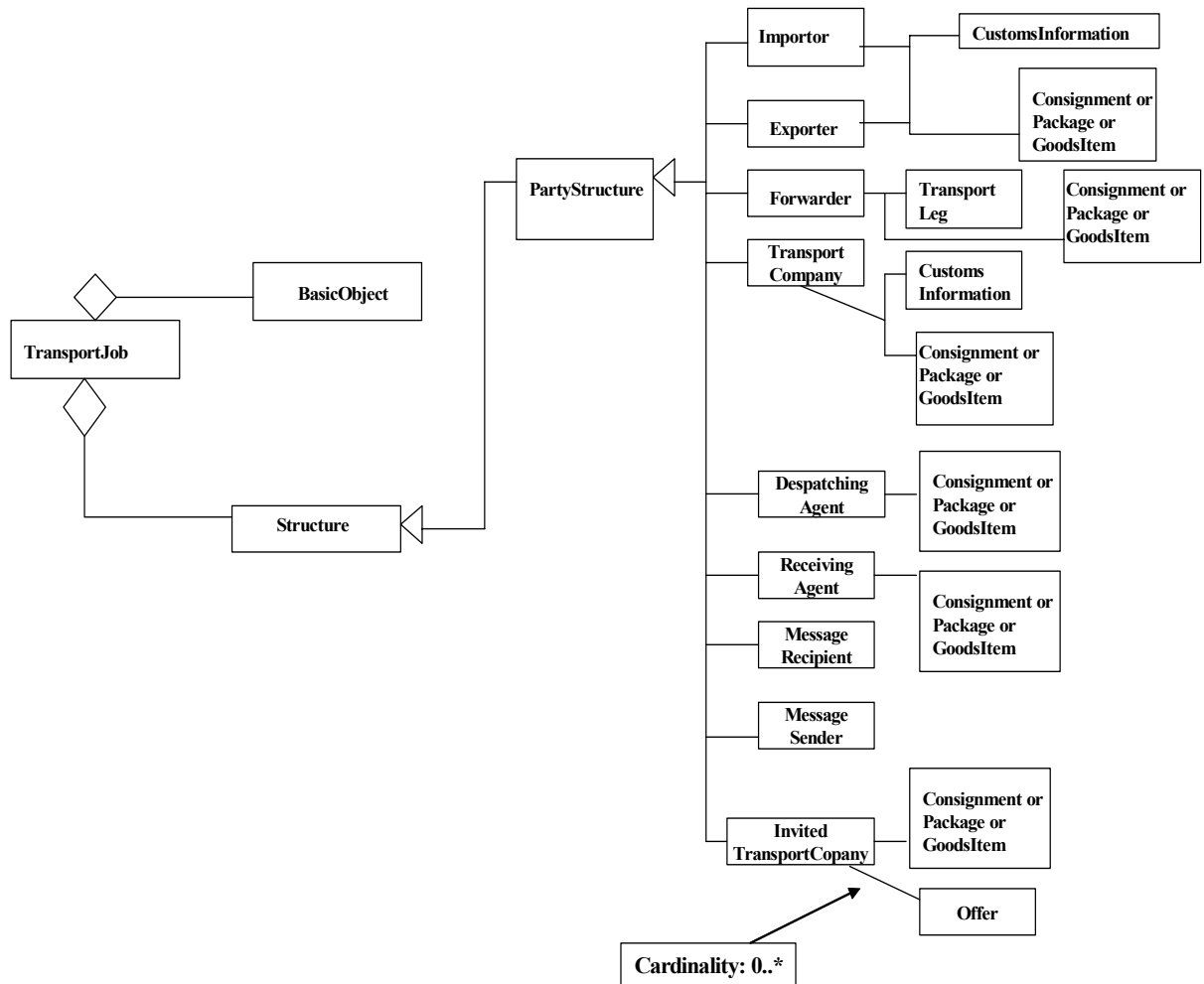
### 3.1.5 Message model TransportJob - Structure part 3: TermsOfDelivery, Party (1)



### 3.1.6 Message model TransportJob - Structure part 4: Party (2)

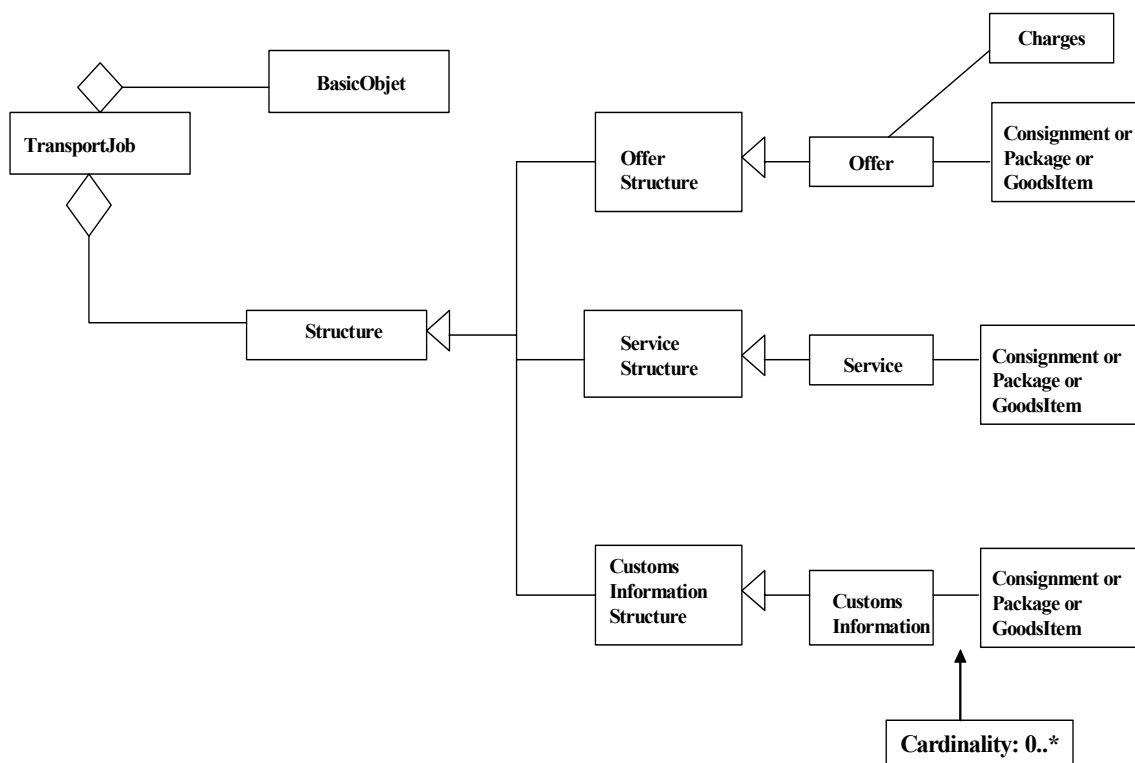


### 3.1.7 Message model TransportJob - Structure part 5: Party (3)

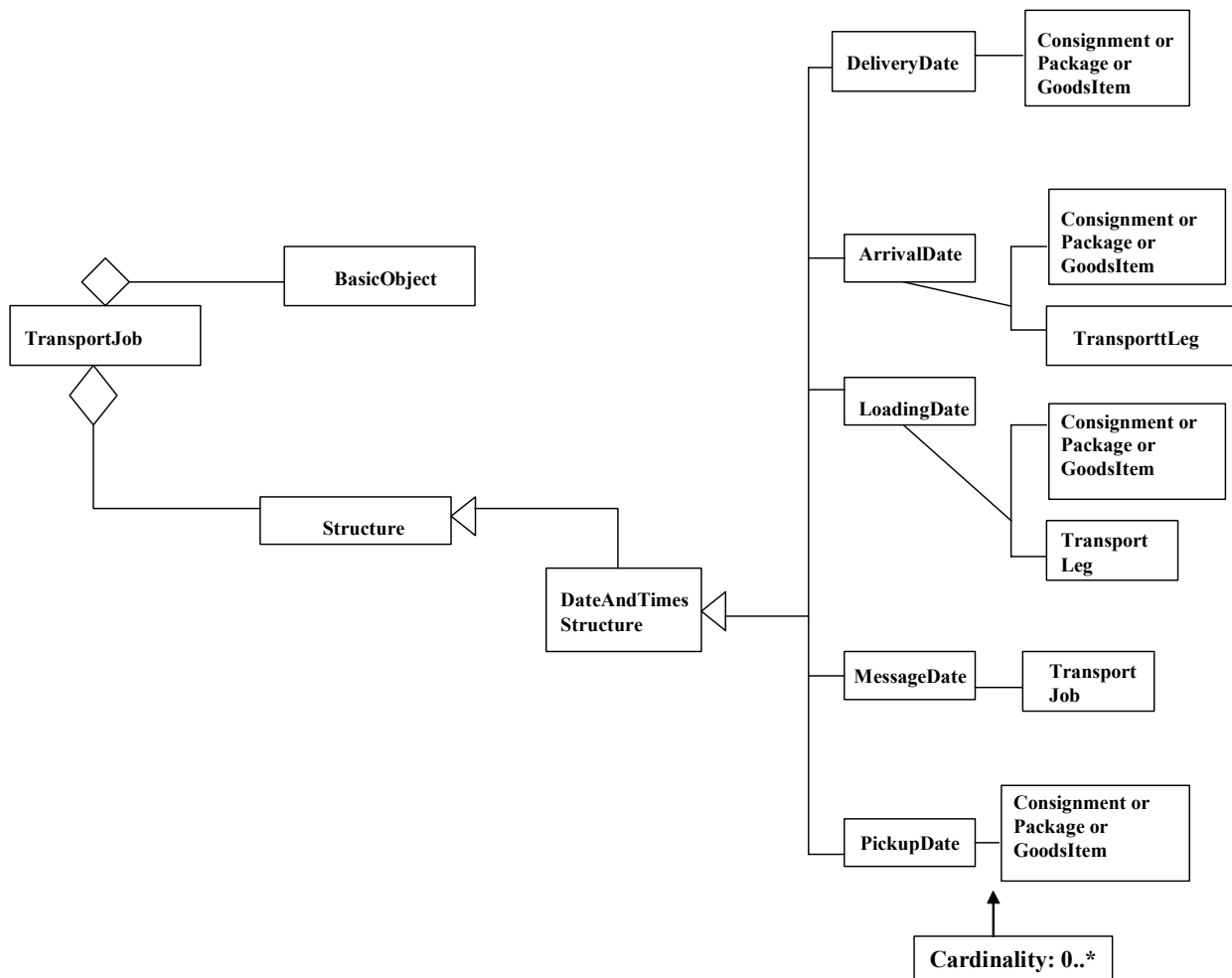




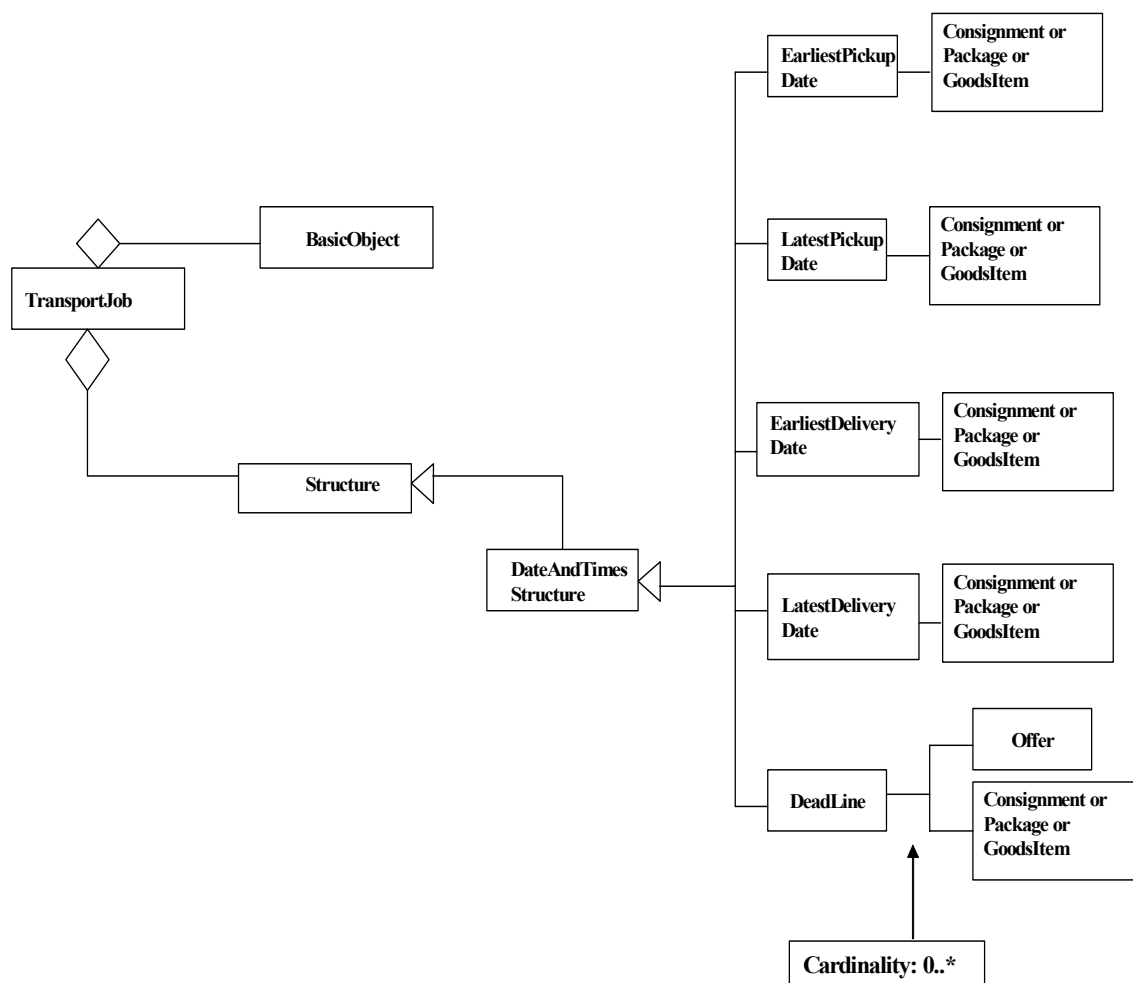
### 3.1.8 Message model TransportJob - Structure part 6: Offer, Service, CustomsInformation



### 3.1.9 Message model TransportJob - Structure part 7: DateAndTimes (1)



### 3.1.10 Message model TransportJob - Structure part 8: DateAndTimes (2)



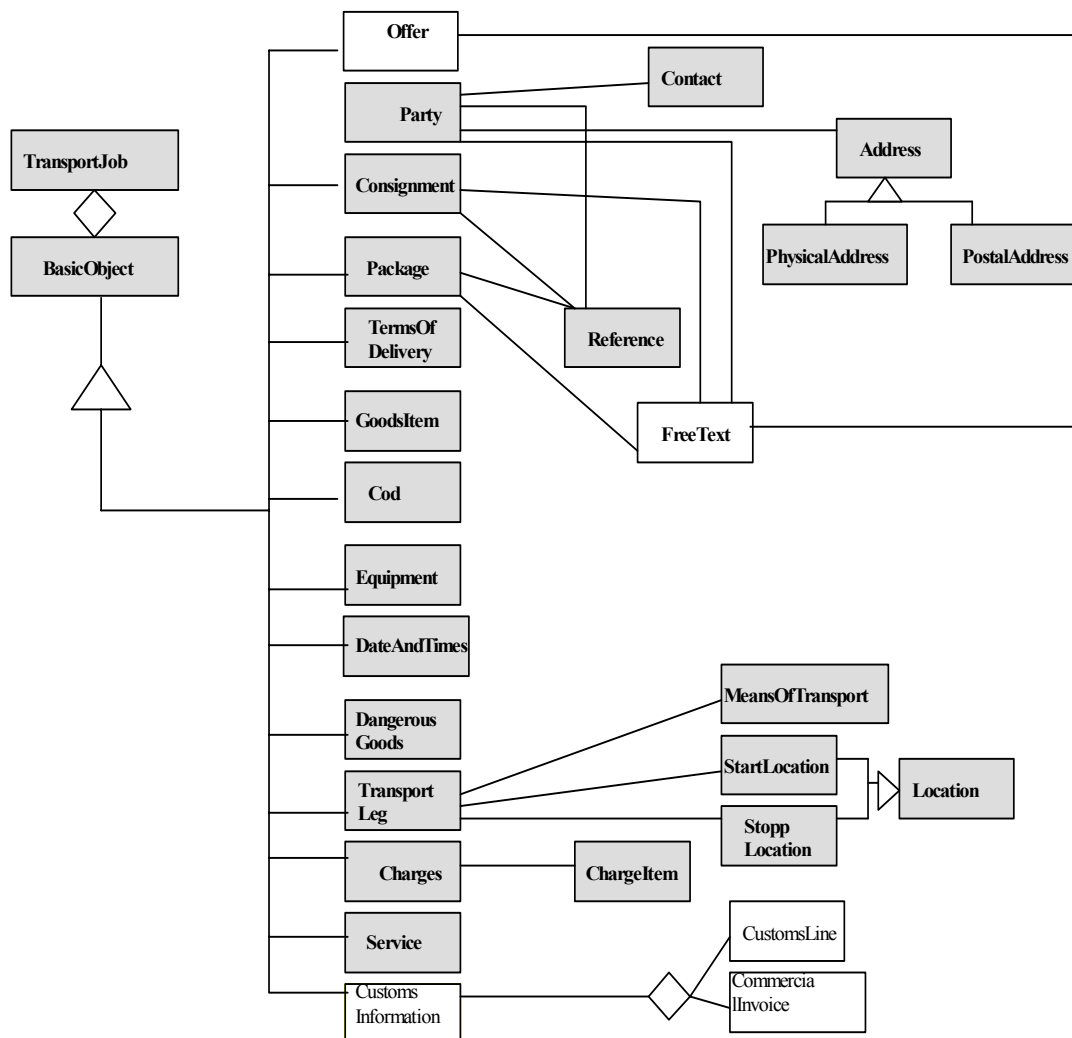
## 3.2 Message model TransportJob – Profile: DomesticNO

This message model describes the subset of TransportJob used by the **DomesticNO** profile – a profile defined for domestic transport in Norway.

Where cardinalities are not supplied, please refer to the “full” TransportJob message model.

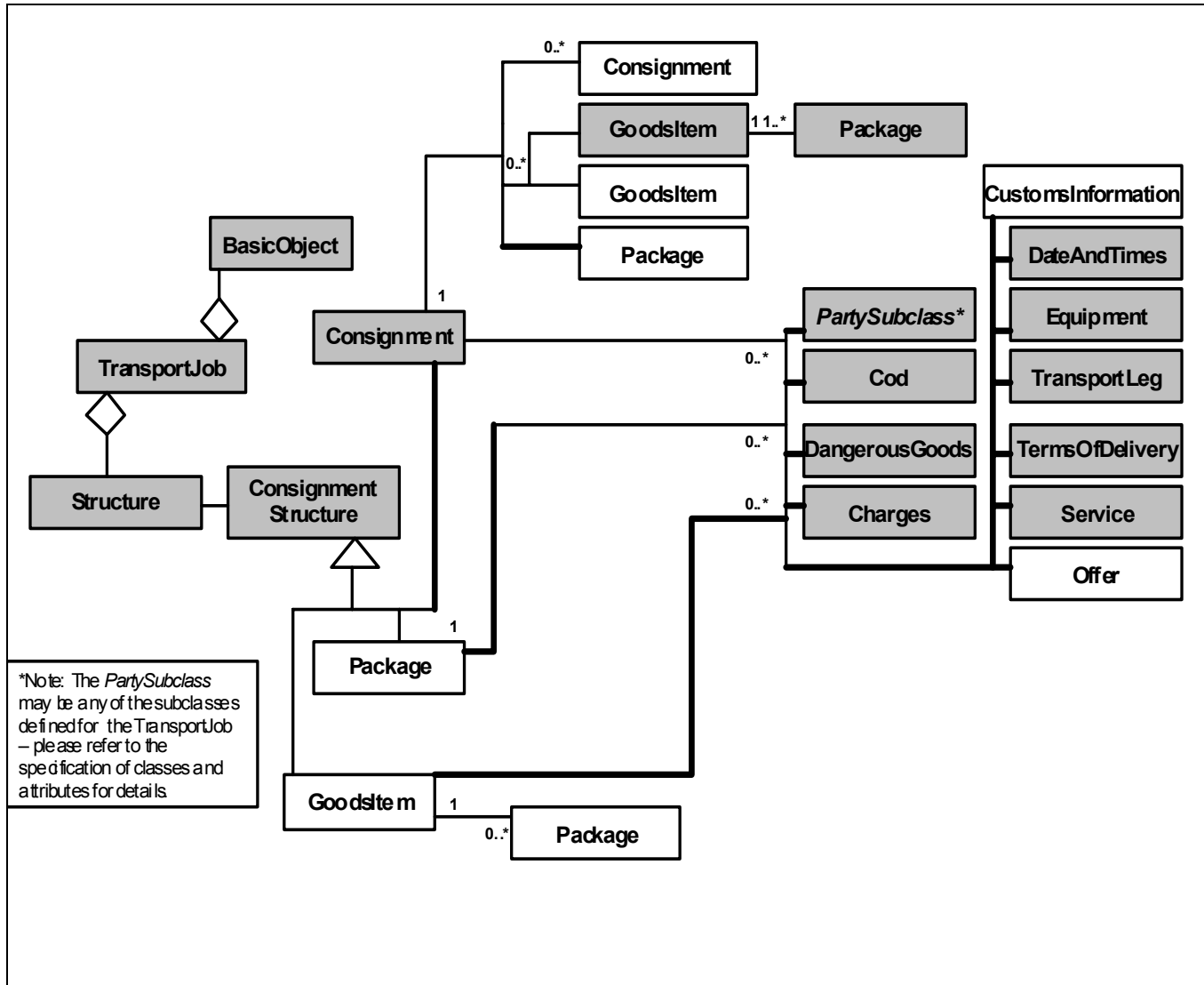
### 3.2.1 Profile DomesticNO – BasicObject

The basic object classes used by the profile are coloured grey.

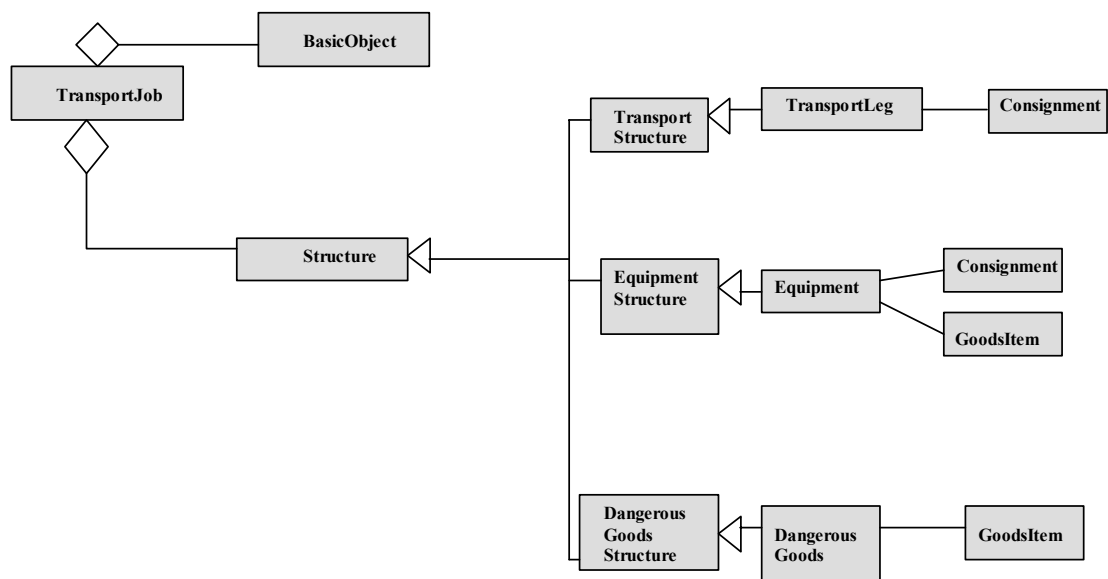


### 3.2.2 Profile DomesticNO – Structure part 1: Consignment

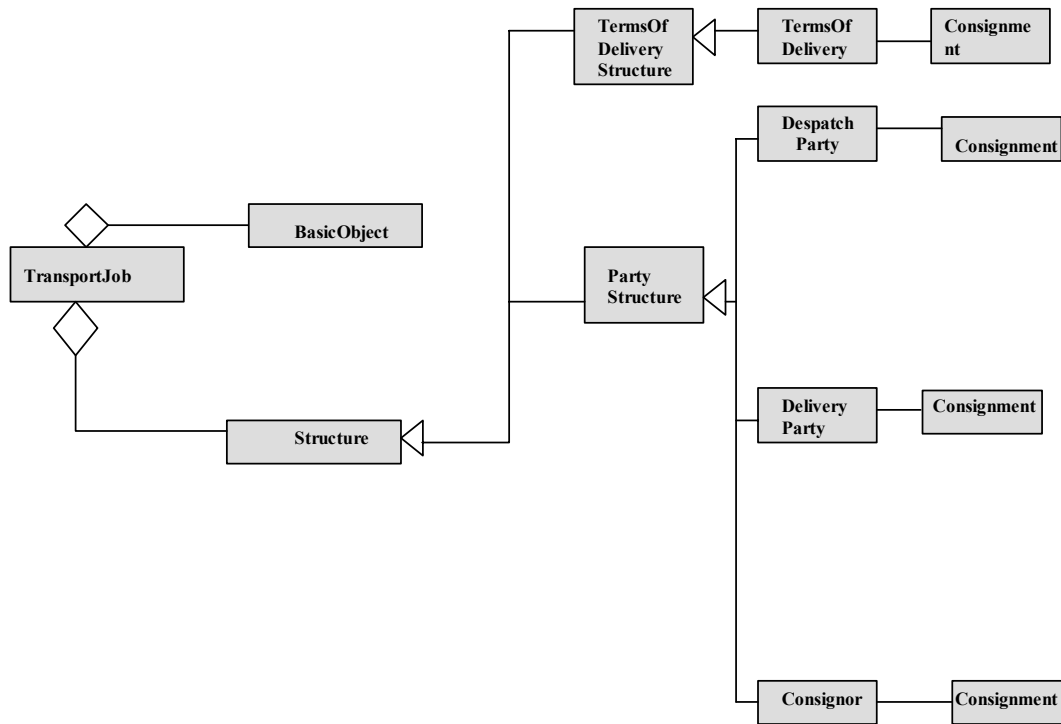
The classes used by the profile are coloured grey.



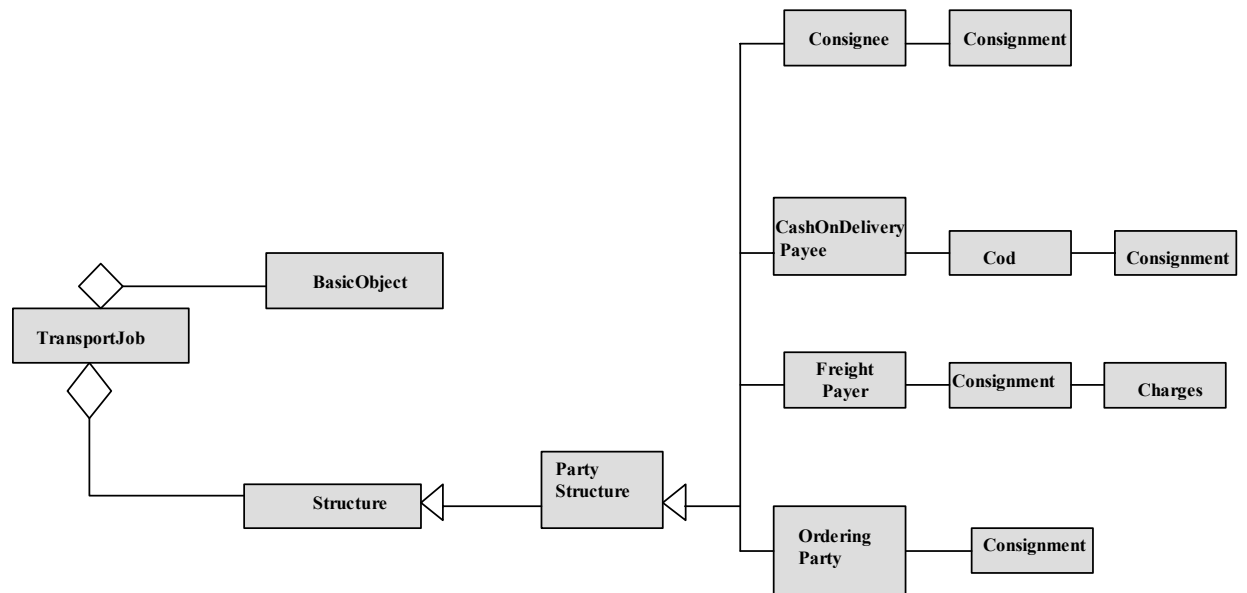
### 3.2.3 Profile DomesticNO - Structure part 2: Transport, Equipment, DangerousGoods



### 3.2.4 Profile DomesticNO - Structure part 3: TermsOfDelivery, Party(1)

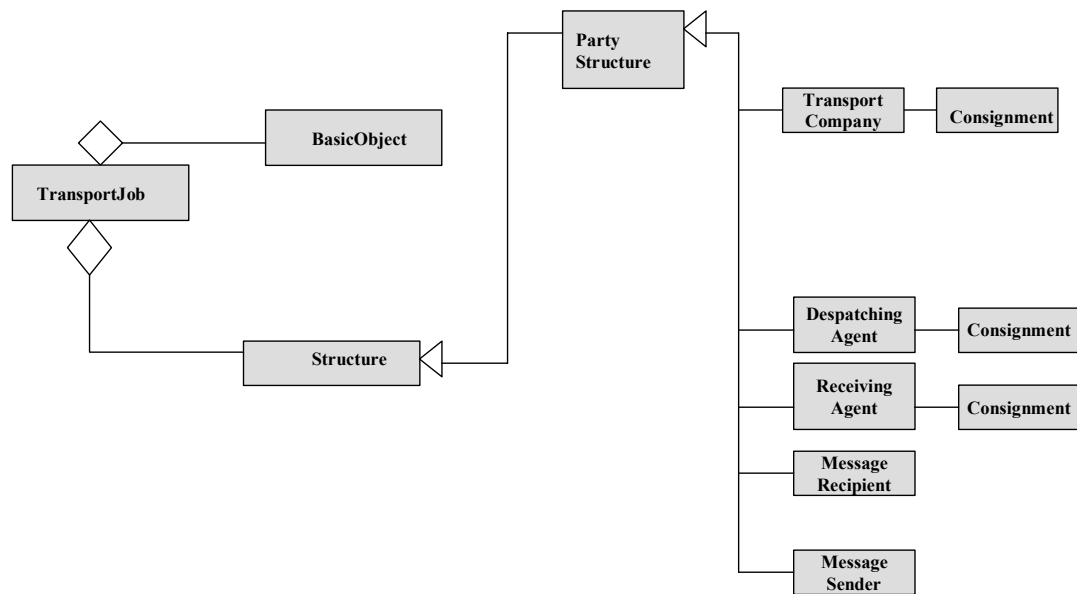


### 3.2.5 Profile DomesticNO - Structure part 4: Party (2)

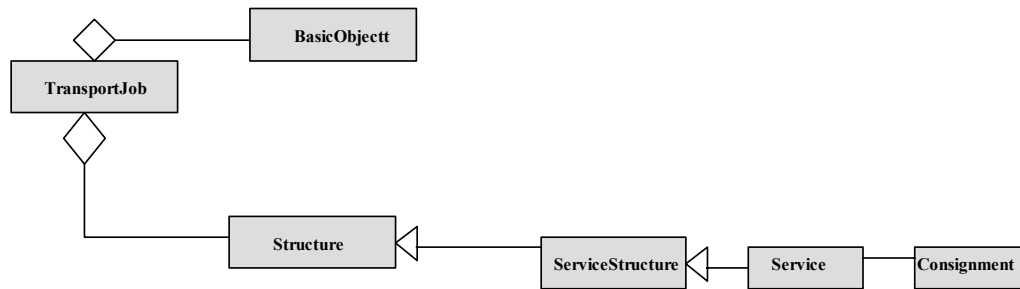




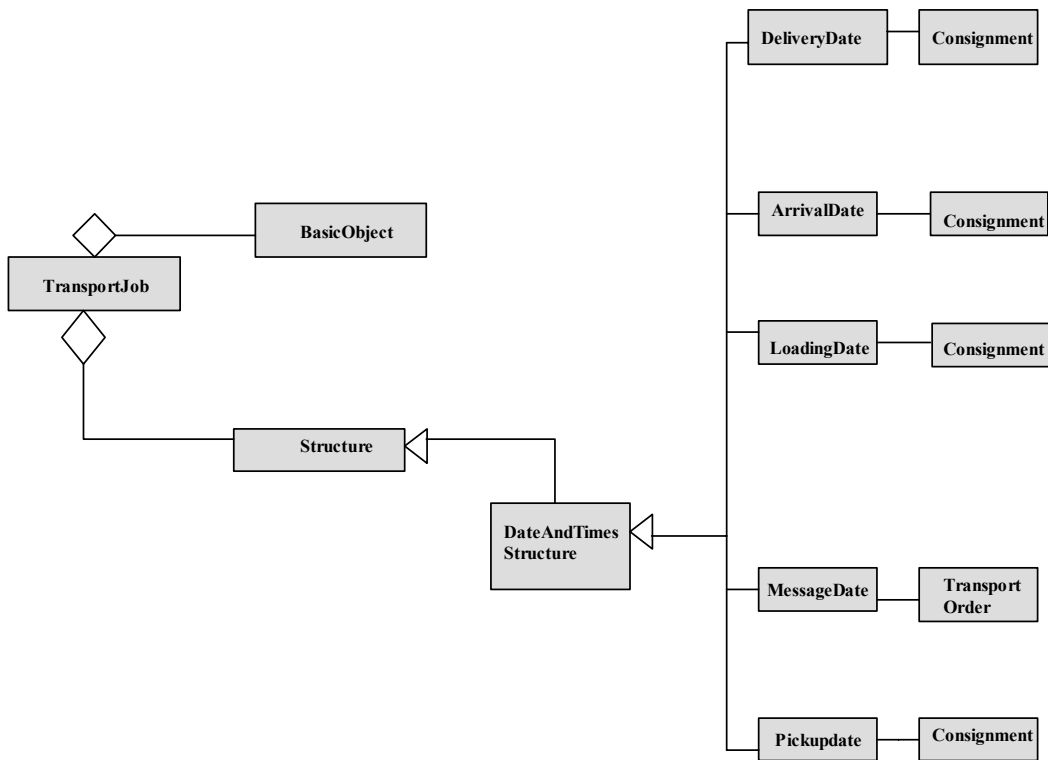
### 3.2.6 Profile DomesticNO - Structure part 5: Party (3)



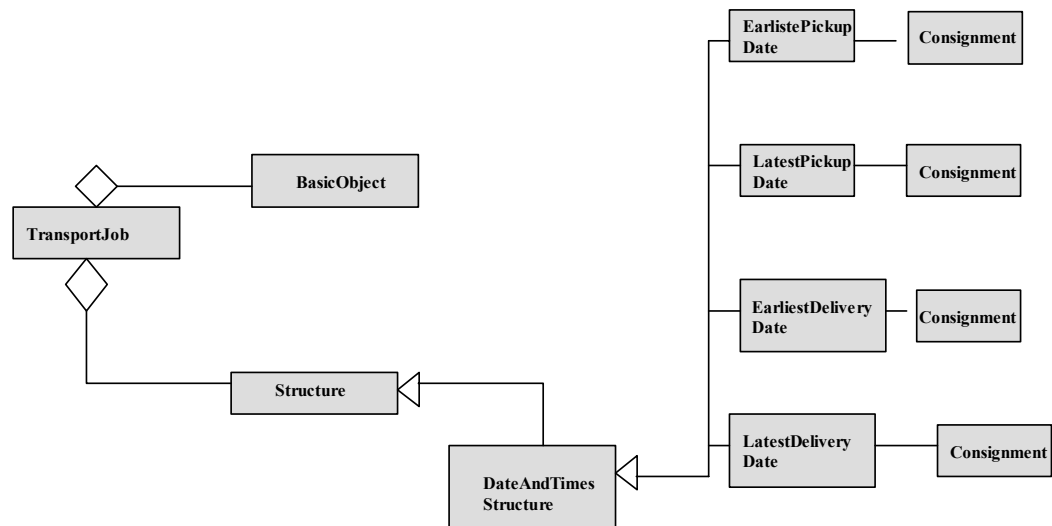
### 3.2.7 Profile DomesticNO - Structure part 6: Service



### 3.2.8 Profile DomesticNO - Structure part 7: DateAndTimes (1)



### 3.2.9 Profile DomesticNO - Structure part 8: DateAndTimes (2)



### 3.2.10 The DomesticNOMin profile: A recommendation for the use of a minimal set of mandatory information objects for domestic transport

The domestic profile DomesticNO describes a subset of the TransportJob message type as a set of selected classes in the message model. This profile does not define any limitation of the model attributes existing in the selected classes. Further it does not prescribe which information set is mandatory in an instance of a domestic transport message.

In addition to the DomesticNO profile transportXML has considered it a need to define ***a minimal mandatory information subset of the TransportJob message*** having the property such that if a sender of a transport job instruction sends a message containing this minimal subset, a Norwegian domestic transport provider is to *guarantee that the transport job will be performed based on the given information*.

This minimal subset is based on the "ordinary" domestic subset (DomesticNO), but it selects the attributes in each respective class required as mandatory in a message instance. We consider the resulting subset – in which *all the identified minimal information is mandatory* – as a profile of its own to be denoted in the TransportJob root element by the name **DomesticNOMin**.

In the list of classes and attributes in chapter 2.3 the class and attribute set of the DomesticNOMin profile is identified by the code **DNoM** in the "Usage" column.

## 3.3 Message model TransportJob – Profile: TransportPortal

The TransportPortal profile defines a subset of the TransportJob message model for use when exchanging messages as part of collaborations in which the transport portal is one of the participants.

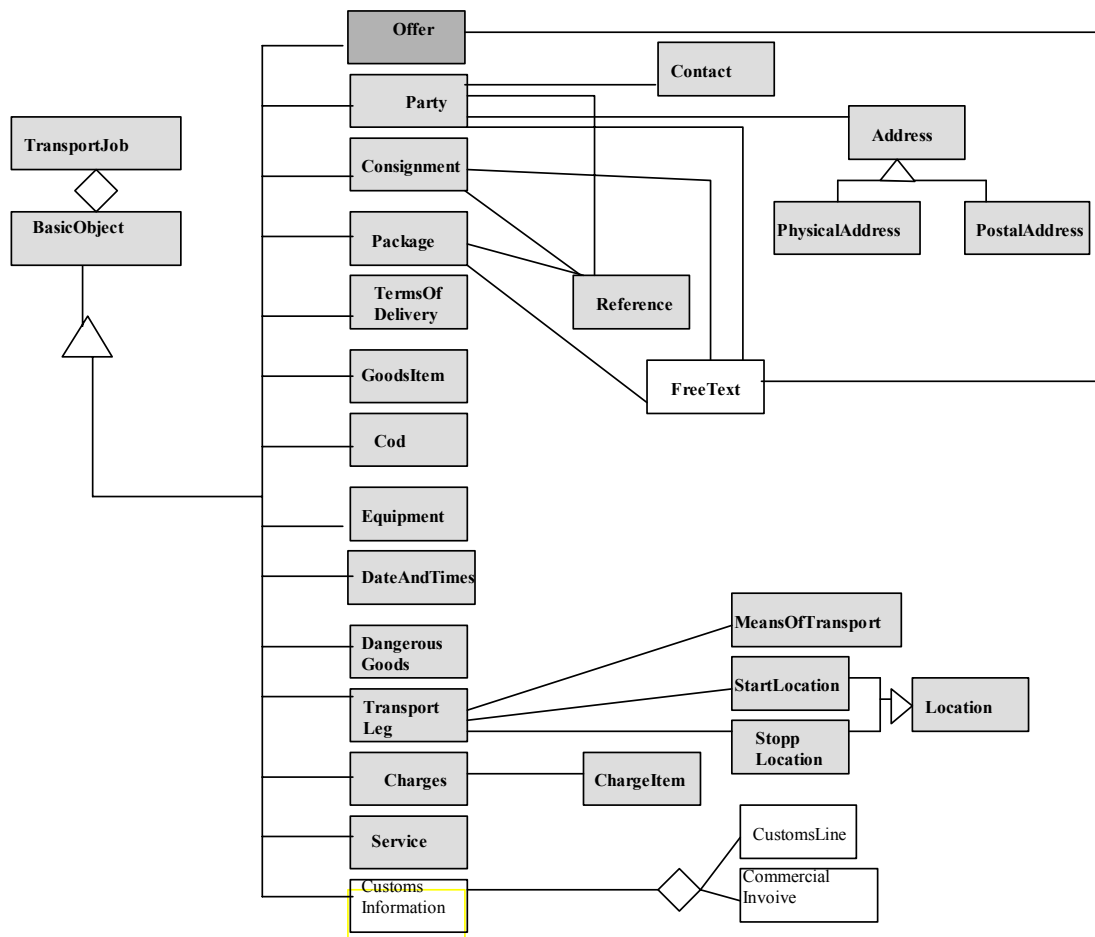
The profile is based on the DomesticNO profile with the following extensions:

1. New structure part: OfferStructure
2. Additions to the DateAndTimesStructure
3. Additions to the PartyStructure

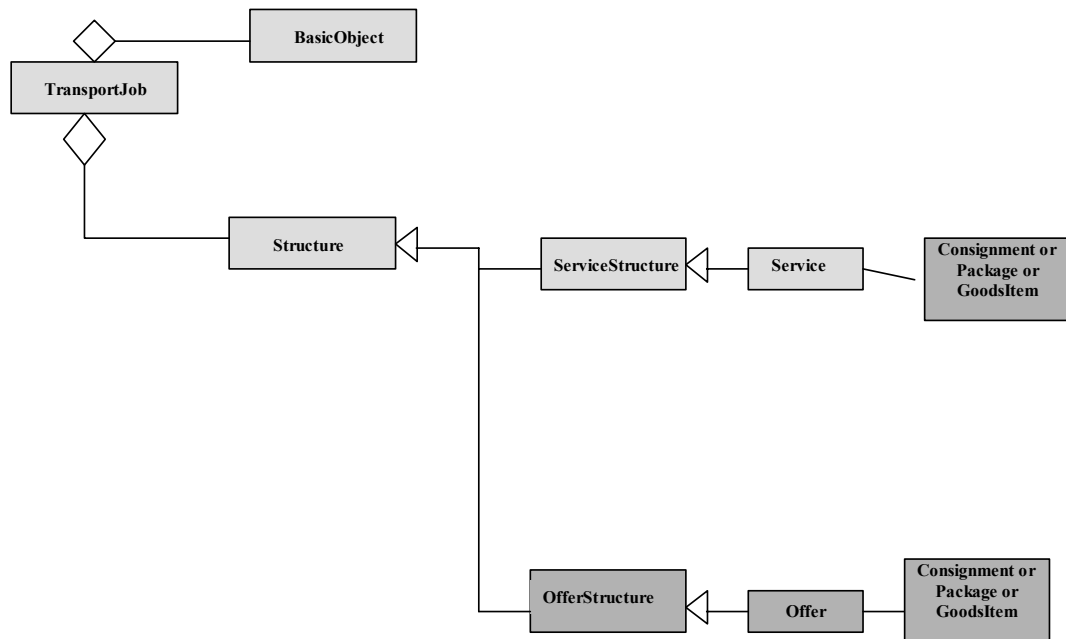
The classes belonging to the profile are marked in grey colour. The structure parts which are identical to the DomesticNO profile are not shown in the following diagrams.

### 3.3.1 Profile TransportPortal - BasicObject

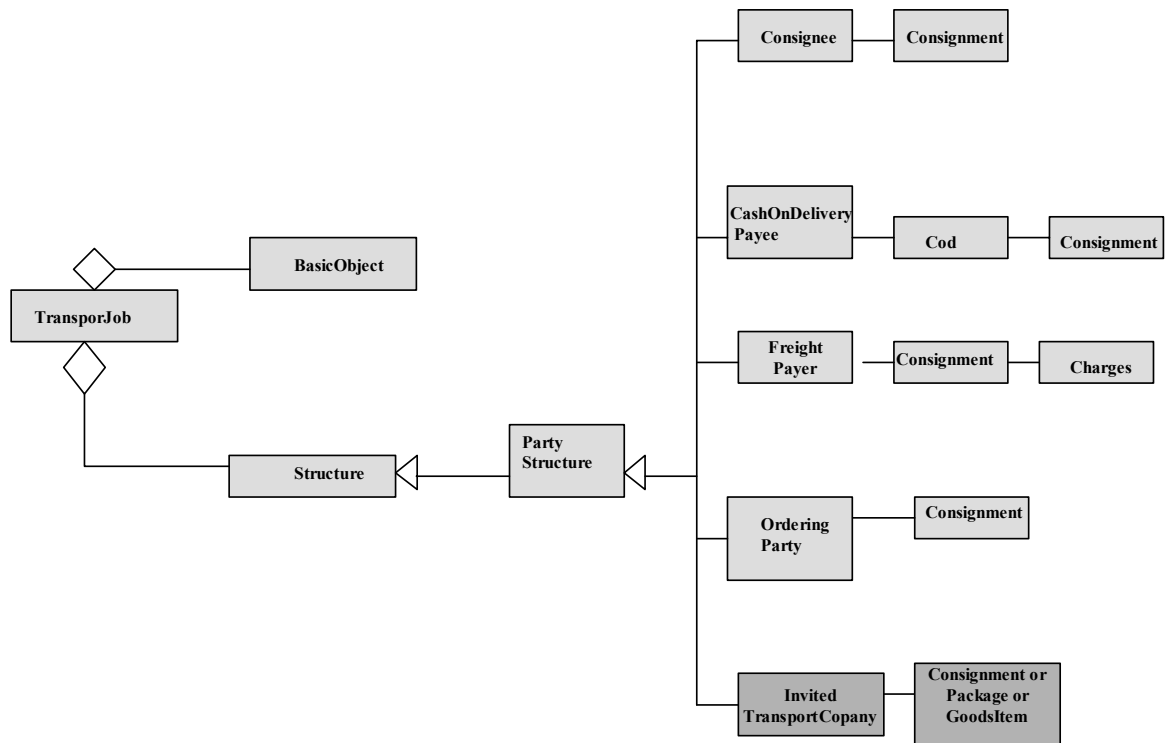
For cardinalities please refer to the ordinary TransportJob message model.



### 3.3.2 Profile TransportPortal - Structure part 1: Offer

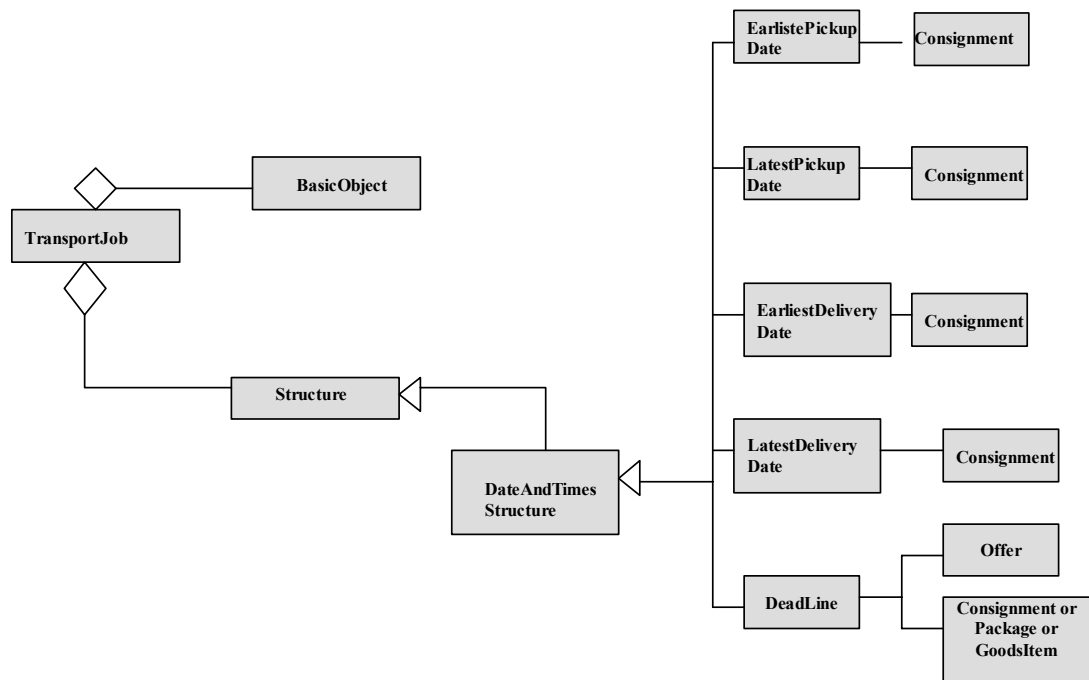


### 3.3.3 Profile TransportPortal – Structure part 2: Party



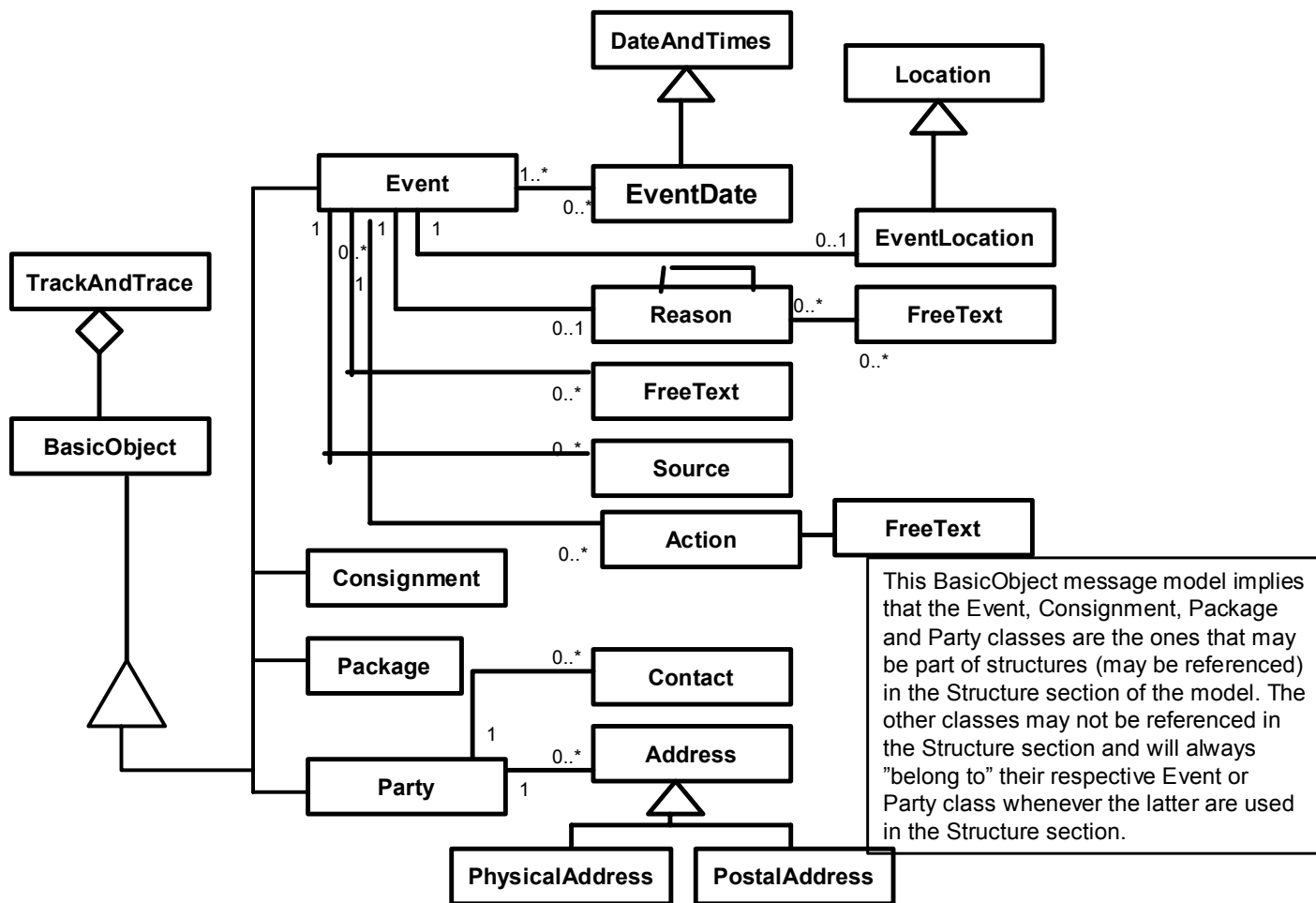


### 3.3.4 Profile TransportPortal – Structure part 3: DateAndTimes

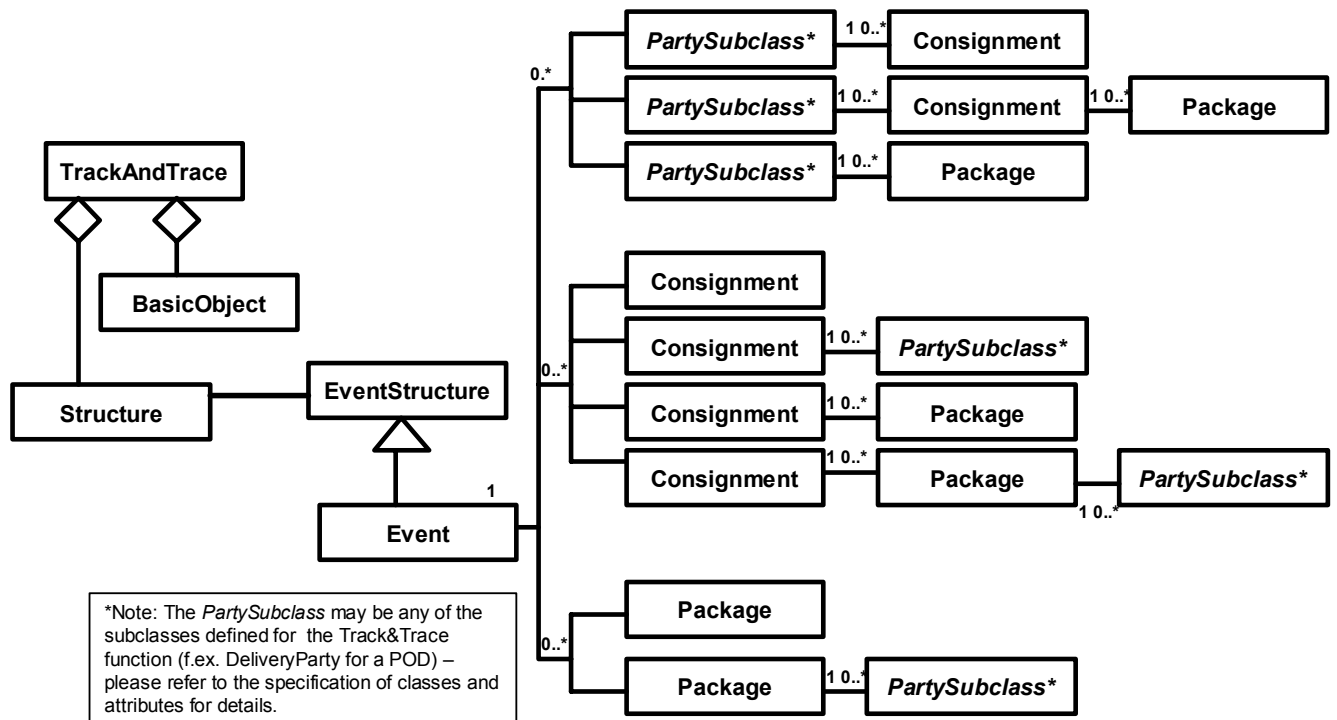


## 4 Message model - TrackAndTrace

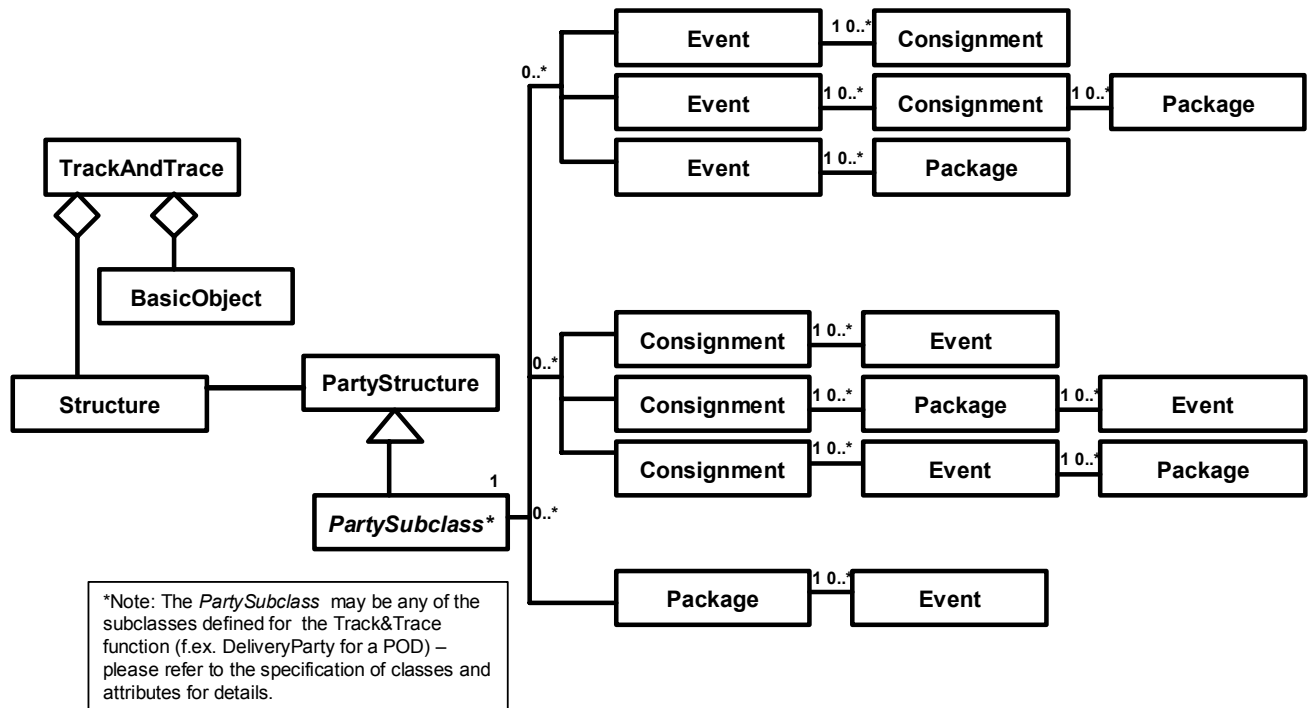
### 4.1 Message model TrackAndTrace : BasicObject



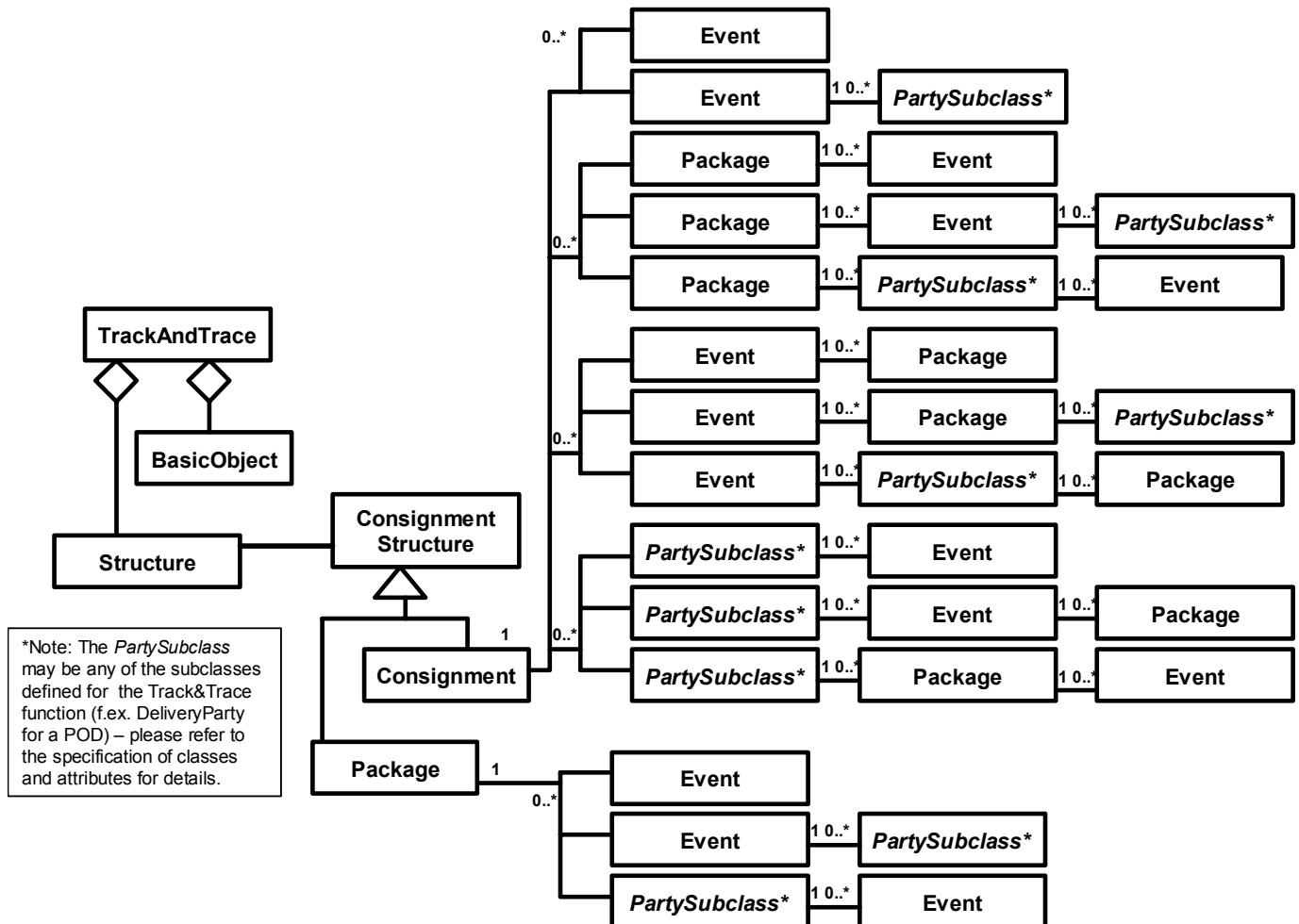
## 4.2 Message model TrackAndTrace – Structure part 1: Event



### 4.3 Message model TrackAndTrace – Structure part 2: Party



#### 4.4 Message model TrackAndTrace – Structure part 3: Consignment



## 5 XML messages

### 5.1 Comprehensive sequential run-through with example: TransportJob

This chapter presents a complete sequential run-through of the TransportJob message type. It contains as such most – if possibly not all – constructs which may be present in a message instance.

The real values which have been applied in elements and attributes, are only examples (they are, though, relevant and possible example values). They are not supposed, however, to be necessarily connected. No real TransportJob message instance would therefore look like the complete example presented below.

In order to get a complete picture of the message type, all possible elements and attributes are included – in certain cases with empty values. The latter ones would naturally not be included in a real message instance.

We refer to **Semantic descriptions of classes and attributes** for a full definition and description of the objects which may be present in the message, including valid code values.

This sequential run-through is intended as a help for those who are to implement the message type. The formal definition of the message type is given by *the message model* and the corresponding *XSD schema*.

### The TransportJob root element

The TransportJob root element is to contain the 7 attributes tabled below:

<b>version</b>	Version. As of October 2002="3.0".
<b>definedBy</b>	"Norsk EDIPRO".
<b>profile</b>	Identifies a subset of the message type, f.ex. "International", "TransportPortal" or "DomesticNO".
<b>domain</b>	Identies the primary domain model which this message type is based on – value = "transportXML"
<b>collaboration</b>	Identifies one of the collaborations which have been defined by transportXML (in version 3.0 11 collaborations have been defined for the Transport Job functional area – please refer to the activity models).
<b>messageFunction</b>	The message function. This is to be one of the identifiers which are specified as names of transitions from one participant to another in the activity diagram defining the collaboration identified by the <i>collaboration</i> attribute – such a transition is often called a <i>flow object</i> in the activity model.
<b>messageId</b>	A unique identifier for the message within the current message exchange setting.

In the root element there do additionally exist 2 optional attributes:  
updateIndicator and testIndicator:

<b>updateIndicator</b>	Original=This is an original message. Update=This is an update of the information in a previously sent message. Deletion=This is an instruction to delete the information in a previously sent message.
<b>testIndicator</b>	YES=This is a test. NO=This is a production message.

Following the root element we have the basic objects of the message, and after the basic objects there is the structure part describing how the basic objects are related to each other. We therefore have the following overall message structure:

```
<?xml version="1.0" encoding="UTF-8"?>
<TransportJob
  version="3.0"
  definedBy="Norsk EDIPRO"
  profile="International"
  domain="transportXML"
  collaboration="ExportManifest"
  messageFunction="Instruction"
  messageId="3"
  updateIndicator="Original"
  testIndicator="NO">

  <BasicObject>
    .....
  </BasicObject>
  <Structure>
    .....
  </Structure>
</TransportJob>
```

## BasicObject

The basic objects which are to be included, are placed as *the first part* of the message. This part is contained in **one BasicObject element**. The structures of which these objects are part, are defined by **one Structure element** as *the second part* of the message.

In the BasicObject part of the message each basic object is allocated an internal reference which is *unique for the object type in question* (the **id** attribute). The present Party objects may f.ex. be numbered from 1 and upwards. This id will be used in the Structure part of the message in order to refer back to this object.

We have the following basic objects:

- **Party**
- **Consignment**
- **GoodsItem**

- **Package**
- **PackageLeg**
- **DateAndTimes**
- **TermsOfDelivery**
- **Cod**
- **Equipment**
- **DangerousGoods**
- **Charges**
- **Service**
- **CustomsInformation**
- **Offer**

#### **Party:**

The Party object is in the BasicObject section described *independent of the function or role of the party* – this role is defined by the *subClass* attribute where this Party is part of a PartyStructure element in the second section of the message.

A Party may have a physical address and a postal address. The addresses are given as **Address** elements within the Party element – the kind of address (physical or postal) is defined by the **subClass** attribute of the Address element.

Linked to a Party there may be:

- **Contact** objects,
- **Reference** objects and
- **FreeText** objects.

The Reference element in TransportJob is *only intended to contain references to objects outside the current message instance*. This element is consequently never used to identify objects which are present in the message nor give internal references within the message.

```
<BasicObject>

<Party
  id="1"
  partyId="929219309"
  idType="82">

  <Name>Skandinavisk Transport System AS
  </Name>
  <VATNo>929219309</VATNo>
  <CustomsCreditNo/>

  <AccountNo>50010509442</AccountNo>
  <BankName>Den norske Bank</BankName>
  <SwiftCode/>

  <Address
    subClass="PhysicalAddress">
    <Street>Frydenbergveien</Street>
    <StreetNo>48</StreetNo>
    <AddressLine>3.etasje</AddressLine>
    <PostalCode>0565</PostalCode>
```



```

    <City>Oslo</City>
    <ZIPCode/>
    <StateOrRegion/>
    <CountryCode>NO</CountryCode>
  </Address>

  <Address
    subClass="PostalAddress">
    <POBox>313</POBox>
    <PostOffice>Økern</PostOffice>
    <PostalCode>0511</PostalCode>
    <City>Oslo</City>
    <ZIPCode/>
    <StateOrRegion/>
    <CountryCode>NO</CountryCode>
  </Address>

  <Contact>
    <PhoneNo>22043830</PhoneNo>
    <TelefaxNo>22043880</TelefaxNo>
    <EMailAddress>phansen@stsas.no
    </EMailAddress>
    <WebAddress>http://www.stsscandinavia.com
    </WebAddress>
    <ContactName>Petra Hansen</ContactName>
    <ContactFunction>Administration
    </ContactFunction>
  </Contact>
  <Contact>
    <PhoneNo>22043800</PhoneNo>
    <ContactFunction>Switchboard
    </ContactFunction>
  </Contact>

  <Reference>
    <ReferenceNo>23444</ReferenceNo>
    <IdType>FF</IdType>
    <Text>Oppdragsnummeret vårt</Text>
  </Reference>

```

<!--Notice that Reference is the only object in which the combination no/idType is not represented by XML attributes, but rather as XML elements. The reason is that ReferenceNo is not a primary key (identifier) of the Reference object – it is a reference from the containing object to an external object (“outside” the message) →

```

  <FreeText>
    <Usage>AAI</Usage>
    <Text>These are general instructions!</Text>
    <NoOfCharacters>31</NoOfCharacters>
  </FreeText>
</Party>

```

### Basic objects of goods structures:

TransportJob may contain 3 kinds of **goods structure objects**:  
consignment, goods item and package.

**Consignment:**

A Consignment object contains attributes describing goods and insurance values and the number of pallet places. The object further contains information related to the consignment as a whole, f.ex. total weights and goods description.

Linked to a Consignment there may be:

- **Reference** objects and
- **FreeText** objects.

**GoodsItem:**

The TransportJob message type enables the traditional goods item construction: a collection of packages displaying a set of common characteristics.

The UN/ECEs recommendation no. 21 is used as the code list for packing type codes.

**Package:**

Package is the smallest identifiable goods structure basic object in TransportJob.

A Package object must have a unique id – for Norwegian domestic transport we prescribe the Serial Shipping Container Code as it is defined in Norsk Edipros "Dokumentasjon i innenlandstransport" ("Documentation for Domestic Transport").

Linked to Package there may – as is the case for Consignment – be:

- **Reference** objects and
- **FreeText** objects.

In our example we explicitly describe 3 packages.

```
<Consignment
  id="1"
  consignmentId="70703202210020004" idType="EANSSCC">

  <NoOfPalletFootPrints>3</NoOfPalletFootPrints>
  <GoodsValue currencyIdentificationCode="NOK">12000</GoodsValue>
  <InsuranceValue currencyIdentificationCode="NOK">
    15000
  </InsuranceValue>
  <TypeOfInsurance>Full</TypeOfInsurance>
  <GoodsDescription>Gloves</GoodsDescription>
  <TotalGrossWeight unitCode="KGM">15.8</TotalGrossWeight>
  <TotalVolume unitCode="MTQ">3.8</TotalVolume>
  <LoadingMetres>0.5</LoadingMetres>
  <Marking>See packages</Marking>
```

```

<NumberOfPackages>10</NumberOfPackages>

<Reference>
  <ReferenceNo>A23</ReferenceNo>
  <IdType>CU</IdType>
  <Text/>
</Reference>

<FreeText>
  <Usage>HAN</Usage>
  <Text>OBS! Fragile!</Text>
  <NoOfCharacters>13</NoOfCharacters>
</FreeText>
</Consignment>

```

```

<GoodsItem
  id="1">
  <GoodsDescription>
    Gloves
  </GoodsDescription>
  <NumberOfPackages>8</NumberOfPackages>
  <PackageTypeCode>PK</PackageTypeCode>
  <Marking>ADR</Marking>
  <GrossWeight unitCode="KGM">12.8</GrossWeight>
  <NetWeight unitCode="KGM"/>
  <Volume unitCode="MTQ">2.4</Volume>
  <LoadingMetres>0.5</LoadingMetres>
</GoodsItem>

<GoodsItem
  id="2">
  <GoodsDescription>
    Hats
  </GoodsDescription>
  <NumberOfPackages>2</NumberOfPackages>
  <PackageTypeCode>PK</PackageTypeCode>
  <Marking>ADR</Marking>
  <GrossWeight unitCode="KGM">3</GrossWeight>
  <NetWeight unitCode="KGM"/>
  <Volume unitCode="MTQ">1.4</Volume>
  <LoadingMetres/>
</GoodsItem>

```

```

<Package
  id="1"
  packageId="370703200135950018"
  idType="EANSSCC">
  <Height unitCode="CMT">30</Height>
  <Width unitCode="CMT">15</Width>
  <Length unitCode="CMT">50</Length>
  <PackageTypeCode>PK</PackageTypeCode>
  <PackageTypeText>Packet</PackageTypeText>
  <Marking>ADR</Marking>
  <GrossWeight unitCode="KGM">10</GrossWeight>
  <NetWeight unitCode="KGM"/>
  <Volume unitCode="MTQ">0.0225</Volume>
  <LoadingMetres/>

  <Reference>
    <ReferenceNo>A23-1</ReferenceNo>
  </Reference>

```

```

    <IdType>CU</IdType>
    <Text/>
  </Reference>

  <FreeText>
    <Usage>HAN</Usage>
    <Text>Do not turn around</Text>
    <NoOfCharacters>18</NoOfCharacters>
  </FreeText>
</Package>

<Package
  id="2"
  packageId="370703200127750015"
  idType="EANSSCC">
  <Height unitCode="CMT">80</Height>
  <Width unitCode="CMT">10</Width>
  <Length unitCode="CMT">90</Length>
  <PackageTypeCode>PK</PackageTypeCode>
  <PackageTypeText/>
  <Marking>ADR</Marking>
  <GrossWeight unitCode="KGM">24</GrossWeight>
  <NetWeight unitCode="KGM"/>
  <Volume unitCode="MTQ">0.072</Volume>
  <LoadingMetres/>
</Package>

<Package
  id="3"
  packageId="370703200127740023"
  idType="EANSSCC">
  <Height unitCode="CMT">50</Height>
  <Width unitCode="CMT">100</Width>
  <Length unitCode="CMT">120</Length>
  <PackageTypeCode>CS</PackageTypeCode>
  <PackageTypeText>Box</PackageTypeText>
  <Marking>ADR</Marking>
  <GrossWeight unitCode="KGM">8</GrossWeight>
  <NetWeight unitCode="KGM">7</NetWeight>
  <Volume unitCode="MTQ">0.6</Volume>
  <LoadingMetres/>
</Package>

```

### TransportLeg :

The transport described by TransportJob, may be split into several transport legs. In the present documentation transportXML has specified codes for precarriage, main transport and postcarriage (but this set may naturally be extended).

A TransportLeg may be connected to one **StartLocation** and one **EndLocation** defining the end points of the transport leg. These objects are represented as subclasses of the **Location** object.

Additionally TransportLeg may be linked to a **MeansOfTransport** object.

```

<TransportLeg
  id="1">
  <TransportLegType>20</TransportLegType>

```

```
<TransportLegId>Route14</TransportLegId>
<ModeOfTransport>30</ModeOfTransport>

<MeansOfTransport>
  <MeansOfTransportId>PAT1
  </MeansOfTransportId>
  <CountryCode>NO</CountryCode>
  <Name>Postman Pat's Royal Mail truck</Name>
  <MeansOfTransportType>31
  </MeansOfTransportType>
</MeansOfTransport>

<Location
  subClass="StartLocation"
  locationId="2120"
  idType="16">
  <LocationType>IndustrialArea
  </LocationType>
  <LocationName>Sagstua industrial area
  </LocationName>
  <CountryCode>NO</CountryCode>
</Location>

<Location
  subClass="EndLocation"
  locationId="5000"
  idType="16">
  <LocationType/>
  <LocationName>Bergen</LocationName>
  <CountryCode>NO</CountryCode>
</Location>
</TransportLeg>
```

**DateAndTimes:**

transportXML has chosen to represent dates and times by specifically denoting: year, month, day, hour and minute.

As a basic object DateAndTimes is defined independent of its function or role, meaning that no subclass identification is present. In the **DateAndTimesStructure** element in section 2 of the message the respective roles are denoted through the subClass attribute.

```
<DateAndTimes
  id="1">
  <Year>2002</Year>
  <Month>12</Month>
  <Day>24</Day>
  <Hour>8</Hour>
  <Minute>30</Minute>
</DateAndTimes>
```

```
<DateAndTimes
  id="2">
  <Year>2003</Year>
  <Month>1</Month>
  <Day>2</Day>
  <Hour>12</Hour>
  <Minute>00</Minute>
</DateAndTimes>
```

**TermsOfDelivery:**

The terms of delivery should be specified using either **Incoterms** or **Combiterms**. **TermsOfDelivery** is defined as a basic object.

As for other basic objects we may describe several TermsOfDelivery objects, which in the structure section of the message may be related to the various consignments, goods items and packages – either as part of the ConsignmentStructure element or as part of the **TermsOfDeliveryStructure** element.

```
<TermsOfDelivery
  id="1">
  <TODOConditionCode>EXW
  </TODOConditionCode>
  <TODOConditionCodeList>
    Incoterms
  </TODOConditionCodeList>
  <TODOLocation>Sagstua</TODOLocation>
</TermsOfDelivery>
```

**Cod:**

Cod (Cash On Delivery) is also defined as a basic object og may later be related to the consignments, goods items and packages in the ConsignmentStructure or **CodStructure** element.

In transportXML we have defined a code list for cod types named:  
(<http://www.edipro.no.transportXML/codes/codTypeCodes>)

```
<Cod
  id="1">
  <CodType
    codeListIdentifier="http://www.edipro.no.transportXML/codes/codTypeCodes">
    CashOnDeliveryFreightValue
  </CodType>
  <TransactionIdentifier>123456
</TransactionIdentifier>
  <CodAmount currencyIdentificationCode="NOK">
    12450.50
  </CodAmount>
  <ControlDigit>4</ControlDigit>
  <DocumentId/>
</Cod>
```

### Equipment:

Equipment is defined as a basic object and may in section 2 be related to consignments, goods items and transport legs.

Properties for size and measurements may be given – these properties should, however, only be provided if the corresponding information is not implicitly given by the equipment type.

```
<Equipment
  id="1">
  <EquipmentType>CN</EquipmentType>
  <EquipmentId>L2347</EquipmentId>
  <ReadingMethod>PDF417
</ReadingMethod>
  <Size>40</Size>
  <Weight unitCode="KGM"/>
  <Height unitCode="CMT"/>
  <Width unitCode="CMT"/>
  <Length unitCode="CMT"/>
  <Volume unitCode="MTQ"/>
</Equipment>
```

### DangerousGoods:

We do not intend to convey all dangerous goods information in the TransportJob message – we have limited our selection of data to the information which *have to be conveyed between the participants* when transporting dangerous goods. This information set includes: UN no, commodity name, class and group.

```
<DangerousGoods
  id="1">
  <UNNo>1891</UNNo>
  <Commodity>
    Etylbromid
```

```
</Commodity>
<Class>6.1</Class>
<Group>II</Group>
</DangerousGoods>
```

### Charges:

The Charges object consists of a set of ChargeItem objects – each ChargeItem object contains information about one charge (= one charge line on an invoice).

Charges is a basic object which in the structure section of the message may be related to the respective consignments, goods items and packages in the ConsignmentStructure or **ChargesStructure** element.

We have in transportXML defined a code list for charge codes named: (<http://www.edipro.no/transportXML/codes/chargeCodes>)

```
<Charges
  id="1">
  <ChargeItem>
    <ChargeCode
      codeListIdentifier="http://www.edipro.no/transportXML/codes/chargeCodes">
      PreCarriage
    </ChargeCode>
    <ChargeText>Pickup</ChargeText>
    <VATCode>1</VATCode>
    <GrossAmount currencyIdentificationCode="NOK">124
    </GrossAmount>
    <NetAmount currencyIdentificationCode="NOK">100
    </NetAmount>
    <TariffAmount currencyIdentificationCode="NOK">150
    </TariffAmount>
    <Discount currencyIdentificationCode="NOK">50</Discount>
    <Comment/>
  </ChargeItem>

  <ChargeItem>
    <ChargeCode
      codeListIdentifier="http://www.edipro.no/transportXML/codes/chargeCodes">
      MainCarriage
    </ChargeCode>
    <ChargeText>
      Main transport from Trondheim to Sola
    </ChargeText>
    <VATCode>1</VATCode>
    <GrossAmount currencyIdentificationCode="NOK">2480
    </GrossAmount>
    <NetAmount currencyIdentificationCode="NOK">2000
    </NetAmount>
    <TariffAmount currencyIdentificationCode="NOK">2100
    </TariffAmount>
    <Discount currencyIdentificationCode="NOK">100</Discount>
    <Comment>Via Oslo</Comment>
  </ChargeItem>
</Charges>
```



**Service:**

We use a basic object to define the transport product or service – possibly including the description of additional services – for the consignments, goods items and/or packages being the subject of the TransportJob message. The link between the respective goods structures and the product specification is contained in the ConsignmentStructure or the **ServiceStructure** in the structure part of the message.

```
<Service
  id="1">
  <BasicServiceCode>DoorDoor
  </BasicServiceCode>
  <AdditionalServiceCode/>
  <Priority>B</Priority>
  <TemperatureControlledGoodsInd>YES
  </TemperatureControlledGoodsInd>
  <MaxTemp>25</MaxTemp>
  <MinTemp>24.5</MinTemp>
  <IdealTemp>24.8</IdealTemp>
</Service>
```

**CustomsInformation:**

The CustomsInformation object is to be used to convey required or wanted customs related information in connection with a transport service. We emphasize that this does not include all the information being necessary in order to perform a full customs clearance of the related goods.

CustomsInformation contains two other objects:

- **CustomsLine** and
- **CommercialInvoice**.

```
<CustomsInformation
  id="1">
  <CustomsClearanceNo>
    2002044341/10115/20020228
  </CustomsClearanceNo>
  <BondedWarehouseNo>
    200201079063002
  </BondedWarehouseNo>
  <SequenceNo>5</SequenceNo>
  <SubLevelSequenceNo/>
  <CustomsValue currencyIdentificationCode="EUR">
    243000
  </CustomsValue>
  <InsuranceAmount currencyIdentificationCode="NOK">
    1550000
  </InsuranceAmount>

  <CustomsLine>
    <CountryOfOrigin>ES</CountryOfOrigin>
```

```

    <StatisticalNo>94036020
    </StatisticalNo>
    <NumberOfPackages>4
    </NumberOfPackages>
    <NetWeight>35</NetWeight>
    <GrossWeight>40</GrossWeight>
    <QuantityOtherUnit/>
    <StatisticalValue currencyIdentificationCode="NOK">
      503000
    </StatisticalValue>
    <GoodsDescription/>
  </CustomsLine>

  <CustomsLine>
    <CountryOfOrigin>ES</CountryOfOrigin>
    <StatisticalNo>94016911
    </StatisticalNo>
    <NumberOfPackages>8
    </NumberOfPackages>
    <NetWeight>50</NetWeight>
    <GrossWeight>60</GrossWeight>
    <QuantityOtherUnit/>
    <StatisticalValue currencyIdentificationCode="NOK">
      275000
    </StatisticalValue>
    <GoodsDescription>Chemicals</GoodsDescription>
  </CustomsLine>

  <CommercialInvoice>
    <InvoiceNo>AR56877-T
    </InvoiceNo>
    <InvoiceDate>20020312</InvoiceDate>
    <InvoiceAmount currencyIdentificationCode="EUR">
      243000
    </InvoiceAmount>
  </CommercialInvoice>
</CustomsInformation>

```

### Offer:

The Offer basic object is used to convey special information about an offer when transport jobs are established through quote requests and auctions at a transport portal (market place). In the structure section of the message the Offer object may be related to the goods which are placed for quoting at the market place.

```

<Offer
  id="1"
  offerId="131210">
</Offer>

</BasicObject>

```

We hereby finish the BasicObject element.

## Structures (the Structure element).

After all the basic objects have been described in the BasicObject section of the message, the message proceeds with the structures in which these objects may be contained. ***This means that we have types of structures corresponding to the types of basic objects.*** These structures are:

- **ConsignmentStructure**
- **TransportStructure**
- **EquipmentStructure**
- **DangerousGoodsStructure**
- **TermsOfDeliveryStructure**
- **PartyStructure**
- **ServiceStructure**
- **CustomsInformationStructure**
- **DateAndTimesStructure**
- **OfferStructure**

The structure part of the message expresses the relationships between the basic objects which have been defined. A terms of delivery may f.ex. be defined for 1 consignment and additionally for a certain set of packages. A party may be allocated a role (let's say the role of consignor) and linked to 2 defined consignments. In this way we extensively reuse the information defined by the basic objects.

In the structures the references to basic objects are expressed using the **href** attribute. The value of **href** should be prefixed by the **#** character, upon which the value of the **id** attribute of the referenced basic object follows.

Most structures will refer to goods objects – combinations of Consignment, GoodsItem and Package. Other basic objects may, however, be included, f.ex. CustomsInformation and TransportLeg. The ConsignmentStructure element is special, since one way of using it is to reference all the basic objects related to the goods object at hand within one and the same Consignment, GoodsItem or Package element (cfr. further details below).

**Comments on the various structures** (note that in order to give complete examples we have used some references which are not part of the BasicObject element above – in a real message instance all **href** references must naturally exist as **id** values in the BasicObject section):

A **ConsignmentStructure** always specify how the goods objects are connected – which packages belong to which goods items, which goods items belong to which consignments, etc. We have the following variants:

- A consignment consists of packages - goods item is not used in this structure
- A consignment consists of goods items – package is not used in this structure
- A consignment consists of goods items which consists of packages

- A consignment consists of other consignments (f.ex. relevant for consolidations)
- A consignment is a structure of its own – no goods items or packages are included
- A goods item consists of packages
- A goods item is a structure of its own – no packages are included in the structure
- A package is a structure of its own

The above list clearly indicates that ***all three goods objects defined by transportXML (Consignment, GoodsItem and Package) may be used as the top level of a goods structure.*** We are consequently able to handle a goods item as a unit independent of any consignment – correspondingly we are able to handle a package as a unit independent of any goods item and/or consignment.

The XML code below shows the various variants of ConsignmentStructure connecting goods object basic objects:

```
<Structure>
  <ConsignmentStructure>
    <Consignment href="#1">
      <Package href="#2"/>
      <Package href="#3"/>
      <GoodsItem href="#1">
        <Package href="#1"/>
      </GoodsItem>
    </Consignment>

    <Consignment href="#2">
      <Consignment href="#11"/>
      <Consignment href="#12"/>
    </Consignment>

    <Consignment href="#3">
      <GoodsItem href="#2"/>
    </Consignment>
    <Consignment href="#7"/>

    <GoodsItem href="#8"/>
    <GoodsItem href="#9">
      <Package href="#21"/>
      <Package href="#22"/>
      <Package href="#23"/>
    </GoodsItem>

    <Package href="#4"/>
  </ConsignmentStructure>
```

In the TransportJob message there are ***two ways in which to use the ConsignmentStructure element.*** These two methods represent two different ways of reusing (i.e. referring to) the respective basic objects.

The first option ***constrains the use of ConsignmentStructure to define which goods objects – goods artifacts – are connected together to***

**form real-life goods structures.** This is the option shown in the example above. Using the ConsignmentStructure element in this way we never reference other basic objects than Consignment, GoodsItem and Package. Which other basic objects are related to the respective goods objects, are defined using other structure elements. As an example we would denote that a consignment has a certain terms of delivery by referencing the respective Consignment basic object in the TermsOfDelivery structure element.

The second option allows us in the ConsignmentStructure element **to reference all the basic objects containing information about the defined Consignment structure, GoodsItem structure or Package structure – in addition to define which goods objects are connected to form real-life goods structures.** The references are to be given “flatly” at the XML level directly below the top level element of the goods structure being described – this top level may be a Consignment element, a GoodsItem element or a Package element. In this way we construct a goods structure which – indirectly through references to the basic object section of the message – contains all the information about the goods object being defined. An example: A manifest message may contain information about 15 consignments – we may then create 15 Consignment elements within the ConsignmentStructure element each of which will reference all the information pertaining to the respective consignment.

The use of the second ConsignmentStructure option will to a large extent reduce the use of the other structure types described below. One might, however, freely combine the two methods by letting some types of basic objects (f.ex. Party objects) be directly referenced in Consignment structures and placing the remaining information about the goods objects in other structure types.

If the TransportJob message is used in collaborations where the main focus is not on the separate goods structures, but on other types of basic objects (like Party or TransportLeg), the first ConsignmentStructure option would be the natural one to select.

Below we give an example of the use of option 2 of ConsignmentStructure – in this example 2 consignments are defined with all relevant references to basic objects:

```
<ConsignmentStructure>
  <Consignment href="#1">
    <GoodsItem href="#1">
      <Package href="#1"/>
      <Package href="#2"/>
      <Package href="#3"/>
    </GoodsItem>
    <GoodsItem href="#2">
      <Package href="#5"/>
      <Package href="#6"/>
    </GoodsItem>
    <Party subClass="Consignor" href="#1"/>
    <Party subClass="Consignee" href="#2"/>
    <Party subClass="DeliveryParty" href="#2"/>
  </Consignment>
</ConsignmentStructure>
```

```

    <Party subClass="FreightPayer" href="#2"/>
    <Party subClass="CashOnDeliveryPayee" href="#2"/>
    <Cod href="#1"/>
    <Charges href="#1"/>
    <TermsOfDelivery href="#1"/>
    <DateAndTimes subClass="ArrivalDate" href="#1"/>
  </Consignment>

  <Consignment href="#2">
    <GoodsItem href="#3"/>
    <Party subClass="Consignor" href="#3"/>
    <Party subClass="Consignee" href="#4"/>
    <Party subClass="DeliveryParty" href="#4"/>
    <TermsOfDelivery href="#1"/>
    <DateAndTimes subClass="ArrivalDate" href="#1"/>
    <Service href="#2"/>
  </Consignment>
</ConsignmentStructure>

```

A **TransportStructure** tells which goods (in the form of references to Consignment, GoodsItem and/or Package) are transported on which transport legs. These transport legs may be part of the same transport (f.ex. precarriage and main transport) or they may be independent of each other.

```

<TransportStructure>
  <TransportLeg href="#1">
    <Consignment href="#1"/>
    <Package href="#4"/>
  </TransportLeg>

  <TransportLeg href="#2">
    <Consignment href="#2"/>
    <Consignment href="#3"/>
  </TransportLeg>

  <TransportLeg href="#3">
    <GoodsItem href="#4"/>
    <GoodsItem href="#5"/>
  </TransportLeg>
</TransportStructure>

```

In **EquipmentStructure** the TransportJob message conveys information about which equipment is used – on which transport legs and for which consignments, goods items or packages.

When an Equipment object is related to a goods object (consignment, goods item or package) and a transport leg (like the example below where the equipment with id=2 is referenced), it means that the referenced equipment is used for the referenced goods object on the referenced transport leg.

```

<EquipmentStructure>
  <Equipment href="#1">
    <Consignment href="#2"/>

```

```

</Equipment>

<Equipment href="#2">
  <GoodsItem href="#1"/>
  <TransportLeg href="#2"/>
</Equipment>

<Equipment href="#3">
  <Package href="#25"/>
</Equipment>
</EquipmentStructure>

```

**DangerousGoodsStructure** tells which consignments, goods items or packages contain dangerous goods. This structure may consequently reference Consignment objects, GoodsItem objects and Package objects.

```

<DangerousGoodsStructure>
  <DangerousGoods href="#1">
    <Consignment href="#1"/>
    <GoodsItem href="#2"/>
  </DangerousGoods>

  <DangerousGoods href="#2">
    <Package href="#1"/>
    <Package href="#3"/>
    <Package href="#5"/>
    <Package href="#12"/>
    <Package href="#14"/>
  </DangerousGoods>
</DangerousGoodsStructure>

```

**TermsOfDeliveryStructure** tells which consignments, goods items or packages have a certain terms of delivery. This structure may consequently reference Consignment objects, GoodsItem objects and Package objects. If we presuppose that the TermsOfDelivery basic object no. 1 defines EXW and that the TermsOfDelivery basic object no. 2 defines DDP, the example below tells us that consignments no. 1 and 2 plus package no. 4 are transported ex works, whereas consignment no. 3 is transported free delivered.

```

<TermsOfDeliveryStructure>
  <TermsOfDelivery href="#1">
    <Consignment href="#1"/>
    <Consignment href="#2"/>
    <Package href="#4"/>
  </TermsOfDelivery>

  <TermsOfDelivery href="#2">
    <Consignment href="#3"/>
  </TermsOfDelivery>
</TermsOfDeliveryStructure>

```

**Party** probably represents the type of basic object to be most frequently reused. In **PartyStructure** we describe ***which parties have which roles*** in relation to the various goods objects (goods structures). PartyStructure may reference Consignment, GoodsItem and Package objects.

For certain Party roles other basic objects may further be referenced:

- If role=FreightPayer, Charges may be referenced inside the reference to Consignment, GoodsItem or Package.
- If role=TransportCompany, Forwarder, DespatchingAgent or ReceivingAgent, the TransportLeg basic object may be referenced.
- If role=Importer or Exporter, CustomsInformation may be referenced.
- If role=CashOnDeliveryPayee, the Cod basic object may be referenced. Inside the Cod reference we may reference Consignment, GoodsItem or Package in order to denote which goods object/objects the delivery restriction pertain to.
- If role=InvitedTransporter, the Offer object may be referenced.

***The Party role is denoted using the subClass attribute in the Party element.*** We have defined the following roles:

- Consignee
- DeliveryParty
- OrderingParty
- FreightPayer
- Consignor
- DespatchParty
- CashOnDeliveryPayee
- BondedWarehouseParty
- Forwarder
- TransportCompany
- Exporter
- Importer
- DespatchingAgent
- ReceivingAgent
- MessageSender
- MessageRecipient
- InvitedTransportCompany

The various structures in which a Party instance may be referenced, are dependent on the specific role played by the party. There are many variants, and we do not present all of them in this run-through example. The examples given below should, however, illustrate the use.

***MessageSender/MessageRecipient:*** These two roles should in fact be unnecessary to include in the TransportJob itself – these are the sender and recipient of the message instance and will consequently be named in a message service protocol envelope in which the message is contained. Using Web Services or ebXML as the message service protocol these parties would be named in the SOAP header. We have included them as possible parties in the TransportJob message as a “security valve”.



```

<PartyStructure>
  <Party href="#1"
    subClass="Consignor">
    <Consignment href="#1"/>
    <GoodsItem href="#3"/>
    <Package href="#4"/>
    <Package href="#8"/>
  </Party>

  <Party href="#1"
    subClass="OrderingParty">
    <Package href="#4"/>
  </Party>

  <Party href="#2"
    subClass="DespatchParty">
    <Consignment href="#1"/>
    <Package href="#8"/>
  </Party>

```

<!--The examples above show that party no. 1 is the consignor of consignment no. 1, goods item no. 3 and packages no. 4 and 8. The same party is the ordering party for the transport of package no. 4. Consignment no. 1 and package no. 8 are to be picked up at party no. 2.
 →

```

  <Party href="#4"
    subClass="CashOnDeliveryPayee">
    <Cod href="#1">
      <Consignment href="#2"/>
    </Cod>
  </Party>

  <Party href="#1"
    subClass="FreightPayer">
    <Consignment href="#1">
      <Charges href="#1"/>
    </Consignment>
    <Package href="#4">
      <Charges href="#2"/>
    </Package>
  </Party>

  <Party href="#4"
    subClass="Exporter">
    <CustomsInformation href="#1"/>
  </Party>

  <Party href="#5"
    subClass="MessageRecipient"/>

  <Party href="#6"
    subClass="MessageSender"/>
</PartyStructure>

```

The structures **ServiceStructure** and **CustomsInformationStructure** may both relate the referenced basic object to consignments, goods items and/or packages. They may consequently both refer to Consignment, GoodsItem and Package objects.

As regards the **OfferStructure**, the Offer object will often be related to GoodsItem objects, since what is placed at the market place for quoting and auctions are cargo items (which are not yet consignments and for which at the time of the auction or quoting no packaging specification needs to be specified). We do, however, include the possibility to reference all three kinds of goods objects from the Offer element in the OfferStructure.

```
<ServiceStructure>
  <Service href="#1">
    <Consignment href="#1"/>
    <GoodsItem href="#9"/>
    <Package href="#4"/>
  </Service>

  <Service href="#2">
    <Consignment href="#2"/>
  </Service>
</ServiceStructure>
```

```
<CustomsInformationStructure>
  <CustomsInformation href="#1">
    <Consignment href="#1"/>
    <Package href="#4"/>
  </CustomsInformation>

  <CustomsInformation href="#2">
    <Consignment href="#2"/>
  </CustomsInformation>
</CustomsInformationStructure>
```

```
<OfferStructure>
  <Offer href="#1">
    <GoodsItem href="#4"/>
    <GoodsItem href="#7"/>
  </Offer>
</OfferStructure>
```

### **DateAndTimesStructure:**

A DateAndTimes object may play several roles, and the DateAndTimes element will therefore – as was the case with the Party object – contain a subClass attribute when it is used in the structure section of the message. The roles are:

- MessageDate
- DeliveryDate
- ArrivalDate
- LoadingDate
- PickupDate
- EarliestPickupDate
- LatestPickupDate
- EarliestDeliveryDate
- LatestDeliveryDate

- DeadLine

In **DateAndTimesStructure** we may reference the goods objects Consignment, GoodsItem and Package. We may further reference the TransportLeg object (relevant for most of the listed roles) and the Offer object (relevant for the Deadline role).

```
<DateAndTimesStructure>
  <DateAndTimes subClass="ArrivalDate" href="#3">
    <Consignment href="#11"/>
    <Consignment href="#12"/>
  </DateAndTimes>
  <DateAndTimes subClass="LoadingDate" href="#1">
    <TransportLeg href="#1"/>
  </DateAndTimes>
</DateAndTimesStructure>

</Structure>
</TransportJob>
```

This marks the end of our TransportJob message.

## 5.2 Comprehensive sequential run-through with example: TrackAndTrace

This chapter presents a complete sequential run-through of the second of the transportXML message types: TrackAndTrace. It contains as such most – if possibly not all – constructs which may be present in a message instance.

As with the TransportJob message in the preceding chapter the real values which have been applied in elements and attributes, are only examples (they are, though, relevant and possible example values). They are not supposed, however, to be necessarily connected.

In order to get a complete picture of the message type, all possible elements and attributes are included – in certain cases with empty values. The latter ones would naturally not be included in a real message instance.

We refer to **Semantic descriptions of classes and attributes** for a full definition and description of the objects which may be present in the message, including valid code values.

This sequential run-through is intended as a help for those who are to implement the message type. The formal definition of the message type is given by *the message model* and the corresponding *XSD schema*.

### The TrackAndTrace root element

The TransportJob root element is to contain the 7 attributes tabled below:

|                        |  |
|------------------------|--|
| <b>version</b>         | Version. As of October 2002="3.0".   |
| <b>definedBy</b>       | "Norsk EDIPRO".  |
| <b>profile</b>         | Identifies a subset of the message type, f.ex. "DomesticNO" or "None" (the message does not use any particular profile).   |
| <b>domain</b>          | Identifies the primary domain model which this message type is based on – value = "transportXML"   |
| <b>collaboration</b>   | Identifies one of the collaborations which have been defined by transportXML (in version 3.0 3 collaborations have been defined for the Track&Trace functional area – please refer to the activity models).  |
| <b>messageFunction</b> | The message function. This is to be one of the identifiers which are specified as names of transitions from one participant to another in the activity diagram defining the collaboration identified by the <i>collaboration</i> attribute – such a transition is often called a <i>flow object</i> in the activity model. |
| <b>messageId</b>       | A unique identifier for the message within the current message exchange setting.   |

In the root element there do additionally exist 2 optional attributes: updateIndicator and testIndicator:

|                        |   |
|------------------------|---|
| <b>updateIndicator</b> | Original=This is an original message.<br>Update=This is an update of the information in a previously sent message.<br>Deletion=This is an instruction to delete the information in a previously sent message. |
| <b>testIndicator</b>   | YES=This is a test.<br>NO=This is a production message.   |

Following the root element we have the basic objects of the message, and after the basic objects there is the structure part describing how the basic objects are related to each other. We therefore have the following overall message structure:

```
<?xml version="1.0" encoding="UTF-8"?>
<TrackAndTrace
  version="3.0"
  definedBy="Norsk EDIPRO"
  profile="None"
  domain="transportXML"
  collaboration="RequestCurrentStatus"
  messageFunction="StatusResponse"
  messageId="2"
  updateIndicator="Update"
  testIndicator="NO">

  <BasicObject>
    .....
  </BasicObject>
  <Structure>
    .....
  </Structure>
</TrackAndTrace>
```

## BasicObject

The basic objects which are to be included, are placed as *the first part* of the message. This part is contained in **one BasicObject element**. The structures of which these objects are part, are defined by **one Structure element** as *the second part* of the message.

In the BasicObject part of the message each basic object is allocated an internal reference which is *unique for the object type in question* (the **id** attribute). The present Party objects may f.ex. be numbered from 1 and upwards. This id will be used in the Structure part of the message in order to refer back to this object.

We have the following basic objects in TrackAndTrace:

- **Event**
- **Consignment**
- **Package**
- **Party**

**Event:**

Event is the most fundamental basic object of the TrackAndTrace message. This object contains information about one **event** or one **state** which is being requested or reported. In transportXML we use the Event object for events (something that happens) as well as states (a condition which exists) – in practical use it is often difficult to separate the two concepts. An attribute Type (element in the XML message) is used to distinguish between the various kinds of Event:

- PhysicalState = physical condition
- PhysicalEvent = physical or operational event
- AdministrativeState = administrative state
- AdministrativeEvent = administrative event
- Deviation = deviation / discrepancy / exception / irregularity

Typically a proof of delivery will be a physical event, whereas a report telling that a consignment has been customs cleared at a particular time will be an administrative event report.

The event itself is codified in the attribute **EventCode**. In transportXML two code lists have been established for use by this attribute:

- <http://www.edipro.no/transportXML/codes/eventCodes> (contains physical and administrative events/states)
- <http://www.edipro.no/transportXML/codes/reasonCodes> (contains exception codes and reason codes)

An event may be further described in a related **FreeText** object.

We may link a **Location** object with the subClass=EventLocation to an Event object. Similarly we may link to the Event object a **DateAndTimes** object with the subClass=EventDate. Very often an event will be reported related to a specific time and/or a specific geographical location.

An Event object may further be connected to a **Source** object. The latter object is used to specify which source has been employed for the data catch upon which the event report is based (this might f.ex. be a scanner or a PDA).

When reporting discrepancies/exceptions the reason for the irregularity should often be included. The Event object may therefore be connected to a **Reason** object. The cause of the exception is denoted by a **ReasonCode** and may be further detailed in an underlying **FreeText** object.

For **one** explicit event we only allow **one** reason to be given. The TrackAndTrace message does, however, allow the **nesting** of Reason object – one Reason element may contain another Reason element which may again contain a Reason element, etc. In this way we may create “causal chains” which may be relevant in exception reporting (example: a

consignment has not been delivered because it has not yet been customs cleared because required paper documentation from the consignor is lacking).

The codes for discrepancies/exceptions and reason are contained in the same code list: <http://www.edipro.no/transportXML/codes/reasonCodes>. The codes in this list may consequently be used as EventCode (in Event) as well as ReasonCode (in Reason).

For discrepancies/exceptions we may in addition to a reason specify an action to cope with the problem caused by the irregularity. This action is specified in the **Action** object. The action may be reported as *already taken by the message sender* or as *recommended to be taken by the message receiver*.

```
<BasicObject>

  <Event
    id="1">
    <Type>PhysicalEvent</Type>
    <CurrentState>NO</CurrentState>
    <EventCode
      codeListIdentifier="http://www.edipro.no/transportXML/codes/eventCodes">
        DeliveredAtDeliveryParty
      </EventCode>

    <DateAndTimes
      subClass="EventDate">
        <Year>2002</Year>
        <Month>10</Month>
        <Day>16</Day>
        <Hour>17</Hour>
        <Minute>42</Minute>
      </DateAndTimes>

    <Location
      subClass="EventLocation"
      locationId="2120"
      idType="16">
        <LocationType
          codeListIdentifier="http://www.edipro.no/transportXML/codes/locationTypes">
            Address
          </LocationType>
          <LocationName>Tiurvegen 34</LocationName>
          <CountryCode>NO</CountryCode>
        </Location>

    <!--Note: The following Reason and Action elements should not be assumed to be part of the
    same event report, since they are related to an exception: a consignment which has not been
    delivered due to an incomplete address. -->
    <Reason>
      <ReasonCode
        codeListIdentifier=http://www.edipro.no/transportXML/codes/reasonCodes>
          AddressIncomplete
        </ReasonCode>
      </Reason>

    <Action>
```

```

    <ActionCode
      codeListIdentifier="MutuallyAgreed">InformationNeeded</ActionCode>
    <ActionType>Requested</ActionType>
    <FreeText>
      <Usage>AAI</Usage>
      <Text>Please asap forward complete delivery address!!!</Text>
      <NoOfCharacters>48</NoOfCharacters>
    </FreeText>
  </Action>
<!-- End of special Reason/Action section -->

  <Source
    sourceId="Dolphin7200-34AA5">
      <UsedBy>Petter Hansen</UsedBy>
      <Type>Automatic</Type>
    </Source>
  </Event>

```

### Party:

The Party object is in the BasicObject section described *independent of the function or role of the party* – this role is defined by the *subClass* attribute where this Party is part of a structure in the second section of the message.

A Party may have a physical address and a postal address. The addresses are given as **Address** elements within the Party element – the kind of address (physical or postal) is defined by the **subClass** attribute of the Address element.

Linked to a Party there may be a **Contact** object. For proof of delivery messages – one of the most important event reports – the Contact object is used to convey the signature (name) of the person who signs the delivery receipt.

```

<Party
  id="1"
  partyId="929219666"
  idType="82">

  <Name>Oles Reklameartikler AS
  </Name>
  <VATNo/>

  <Address
    subClass="PhysicalAddress">
      <Street>Tiurvegen</Street>
      <StreetNo>34</StreetNo>
      <AddressLine/
      <PostalCode>2120</PostalCode>
      <City>SAGSTUA</City>
      <ZIPCode/>
      <StateOrRegion/>
    </Address>
  </Party>

```



```

    <CountryCode>NO</CountryCode>
  </Address>

  <Address
    subClass="PostalAddress">
    <POBox>12</POBox>
    <PostOffice>Sagstua</PostOffice>
    <PostalCode>2120</PostalCode>
    <City>Sagstua</City>
    <ZIPCode/>
    <StateOrRegion/>
    <CountryCode/
  </Address>

  <Contact>
    <ContactName>Gunda Brattbakkollen</ContactName>
    <ContactFunction>Receipt</ContactFunction>
  </Contact>

</Party>

```

### Goods basic objects (Consignment and Package):

TrackAndTrace may contain 2 types of goods basic objects: Consignment and Package. These represent the goods units for which it is relevant to request or report events or states.

A TrackAndTrace message should not be used to convey general consignment or package information – this is the job of the TransportJob message. Therefore not all of the attributes of the Consignment or Package objects are relevant for a TrackAndTrace message. Basically two kinds of information items apply:

- identification of the consignment or package (a unique identifier must be specified – for Norwegian domestic transport we prescribe the Serial Shipping Container Code as it is defined in Norsk EDIPRO's "Dokumentasjon i innenlandstransport")
- corrected measurements and weights (possibly being the result of a scanning event).

Below please find examples of Consignment and Package:

```

<Consignment
  id="1"
  consignmentId="70703202210020004" idType="EANSSCC">
  <TotalGrossWeight unitCode="KGM">15.8</TotalGrossWeight>
  <TotalVolume unitCode="MTQ">3.8</TotalVolume>
  <LoadingMetres/>
</Consignment>

```

```

<Package
  id="1"
  packageId="370703200135950018"

```

```
idType="EANSSCC">
  <Height unitCode="CMT">10</Height>
  <Width unitCode="CMT">100</Width>
  <Length unitCode="CMT">50</Length>
  <GrossWeight unitCode="KGM">17.5</GrossWeight>
  <NetWeight unitCode="KGM"/>
  <Volume unitCode="MTQ">0.05</Volume>
  <LoadingMetres/>
</Package>
```

```
</BasicObject>
```

Here we end the BasicObject section of the message.

## Structures (the Structure element).

After all the basic objects have been described in the BasicObject section of the message, the message proceeds with the structures in which these objects may be contained. ***This means that we have types of structures corresponding to the types of basic objects.*** These structures are:

- **EventStructure**
- **ConsignmentStructure**
- **PartyStructure**

The structure part of the message expresses the relationships between the basic objects which have been defined. In the TrackAndTrace message the choice of structure in the structure section depends on the *focus* (or point of view) of the status message (in other words: what is the *primary subject* of the message). We may give the following guidelines:

- If we report an event/state and are to specify which goods objects and/or which parties this event/state pertains to, the **EventStructure** should be used.
- If the focus of attention is a specific goods object (f.ex. a consignment), and we are to report (or request) all relevant events for this goods object, the **ConsignmentStructure** should be used.
- If we report or request several (or all) of the events pertaining to a specific party, the **PartyStructure** should be used. An example: We report to our agent abroad about all consignments delivered too late to consignee Peter Haines Ltd during week 38.

In the structures the references to basic objects are expressed using the **href** attribute. The value of **href** should be prefixed by the **#** character, upon which the value of the **id** attribute of the referenced basic object follows.

**Comments on the various structures** (note that in order to give complete examples we have used some references which are not part of the BasicObject element above – in a real message instance all **href** references must naturally exist as **id** values in the BasicObject section):

### **EventStructure:**

An EventStructure has the Event element as its focus. The event may be related to one or more consignments and/or one or more packages. It is further possible to report events for consignments/packages per party (the structure Event -> Party-> goods object).

Below please find two examples of the use of EventStructure conveying exactly the same information: An event (Event no. 1 – let's say it is a proof of delivery) is reported for packages no. 2 and 3 plus for consignment no. 1 related to party no. 1 (being the delivery party – as we can see the party's role is specified in the subClass attribute):

<Structure>
-------------

```

<EventStructure>
  <Event href="#1">
    <Package href="#2">
      <Party subClass="DeliveryParty" href="#1"/>
    </Package>
    <Package href="#3">
      <Party subClass="DeliveryParty" href="#1"/>
    </Package>
    <Consignment href="#1">
      <Party subClass="DeliveryParty" href="#1"/>
    </Consignment>
  </Event>
</EventStructure>

<EventStructure>
  <Event href="#1">
    <Party subClass="DeliveryParty" href="#1"/>
    <Package href="#2"/>
    <Package href="#3"/>
    <Consignment href="#1"/>
  </Party>
</Event>
</EventStructure>

```

### PartyStructure:

A PartyStructure reports or requests events pertaining to a specific party. Below the Party level in the message we can choose whether to report ***goods objects per event*** or ***events per goods object***.

The Party element must be specified with its role in the subClass attribute. All the roles defined in our transport domain model are potentially applicable.

Below please find two examples of the use of PartyStructure. The first one shows goods objects reported per event for a DeliveryParty, whereas the second one shows events reported per goods object for a DeliveryParty.

```

<PartyStructure>
  <Party subClass="DeliveryParty" href="#1">
    <Event href="#1">
      <Consignment href="#1"/>
      <Package href="#2"/>
      <Package href="#3"/>
    </Event>
    <Event href="#5">
      <Consignment href="#2">
        <Package href="#11"/>
        <Package href="#12"/>
        <Package href="#13"/>
      </Consignment>
    </Event>
  </Party>
</PartyStructure>

```

```

<PartyStructure>
  <Party subClass="DeliveryParty" href="#1">
    <Consignment href="#1">

```

```

    <Event href="#1"/>
    <Event href="#2"/>
    <Event href="#14"/>
  </Consignment>
  <Package href="#31">
    <Event href="#5"/>
  </Package>
</Party>
</PartyStructure>

```

### ConsignmentStructure:

We use ConsignmentStructure to request or report about several (or all) events related to a Consignment or a Package. At the level below the goods object in the message we may choose whether to report **events per party** or **events in which the party possibly may be connected to each reported event**.

The below examples show the use of ConsignmentStructure.

First we depict a consignment given two events related to a DeliveryParty, of which event no. 5 is further linked to two identified packages (this might be a situation in which consignment no. 1 is reported as delivered as event no. 1, but when delivered a damage was registered for packages no. 8 and 9 in this consignment – the “damage exception event” will then be event no. 5). For the same consignment we further report event no. 12 for an importer (this might be a customs clearance administrative event).

Subsequently we show a ConsignmentStructure in which events are reported directly (with no party in between) per goods object – Consignment no. 2 and Package no. 8 respectively. As can be seen a party may further be linked to each event which is being reported.

```

<ConsignmentStructure>
  <Consignment href="#1">
    <Party subClass="DeliveryParty" href="#1">
      <Event href="#1"/>
      <Event href="#5">
        <Package href="#8"/>
        <Package href="#9"/>
      </Event>
    </Party>
    <Party subClass="Importer" href="#1">
      <Event href="#12"/>
    </Party>
  </Consignment>
</ConsignmentStructure>

```

```

<ConsignmentStructure>
  <Consignment href="#2">
    <Event href="#1">
      <Party subClass="DeliveryParty" href="#1"/>
    </Event>
  </Consignment>

  <Package href="#8">
    <Event href="#5">

```

```
<Party subClass="DeliveryParty" href="#1"/>
</Event>
<Event href="#6"/>
<Event href=""7"/>
</Package>
</ConsignmentStructure>
```

```
</Structure>
</TrackAndTrace>
```

And then we have reached the end of our TrackAndTrace message.