Title: WP8 B2B Case Study Ontologies & Services

Version: 1.0
Date: 31/03/2004
Pages: 78

Responsible Authors:
Javier.Esplugas-Cuadrado@hp.com
Chris.Preist@hp.com

Hewlett Packard Labs, Bristol

Co-Author(s):
Stuart.Williams@hp.com

Hewlett Packard Labs, Bristol

SWWS – Semantic Web Enabled Web Services

Status:

[ ] Draft
[ ] To be reviewed
[ ] Proposal
[✓] Final / Released to CEC

Confidentiality:

[✓] Public - for public use
[ ] INT - for SWWS consortium (and Project Officer if requested)
[ ] Restricted - for SWWS consortium and Project Officer only

Project ID: IST-2002-37134

Deliverable ID: D8.2

Workpackage No: 8

Title: WP8 B2B Case Study Ontologies and Services

Summary / Contents:

This document includes the in-deep analysis of the case study as a set of web services and presents which are the ontologies that semantically support those services. This document follows the Deliverable 8.1 which included the Case Study requirements and the conceptual architecture of that domain. There are two main sections, Services and Ontologies. The first main section talks about the services that present the participants of the case study. We also introduce the concept of agreement template and how this item becomes an important player in the discovery and matchmaking phase. The second main section of the document refers to the ontologies that we use in the case study to support those web services and the interaction between them.

SWWS Consortium

This document is part of a research project funded by the IST Programme of the Commission of the European Communities as project number IST-2002-37134. The partners in this project are: Leopold-Franzens Universität Innsbruck (IFI, Austria)); National University of Ireland, Galway (NUI, Galway, Ireland); Forschungszentrum Informatik (FZI, Germany); Intelligent Software Components S.A. (iSOCO, Spain); OntoText Lab. - Sirma AI Ltd. (SAI, Bulgaria); Hewlett Packard (HP, UK), British Telecom (BT, UK)

Leopold-Franzens Universität Innsbruck (IFI)

Institut für Informatik
Technikerstrasse 13
A-6020 Innsbruck Austria

Tel: +43 512 507 6489
Fax: +43 512 507 9872

Contact person: Juan Miguel Gomez
E-mail: juan.miguel@uibk.ac.at

National University of Ireland, Galway (NUI)

National University of Ireland,
University Road
Galway, Ireland

Tel: +353 91 750414
Fax: +353 91 562894

Contact person: Liam Caffrey
E-mail: Liam.Caffrey@nuigalway.ie

FZI – Forschungszentrum Informatik

Haid-und-Neu-Str. 10-14
76131 Karlsruhe, Germany

Tel: +49 721 9654816
Fax: +49 721 9654817
Contact person: Adreas Abecker
E-mail: abecker@fzi.de

Intelligent Software Components S.A. (iSOCO)

Francisco Delgado 11, 2nd Flor
28108 Alcobendas, Madrid, Spain

Tel: +34 913 349797
Fax: +34 913 349799

Contact person: Richard Benjamins
E-mail: rbenjamins@isoco.com

OntoText Lab.- Sirma AI Ltd. (SAI)

OntoText Lab.
38A Chr. Botev Blvd.
Sofia 1000, Bulgaria

Tel: +35 92 9810018,
Fax: +35 92 9819058

Contact person: Atanas Kiryakov
E-mail: Atanas.Kiryakov@sirma.bg

Hewlett Packard (HP)

HP European Laboratories
Filton Road, Stoke Gifford
BS34 8QZ Bristol, UK

Tel: +44 117 3128631
Fax: +44 117 3129285

Contact person: Janet Bruten
E-mail: janet.bruten@hp.com

Associated Partner:
British Telecommunications plc. (BT)
Orion 5/12, Adastral Park
Ipswich ip5 3RE, UK
Tel: +44 1473 609583
Fax: +44 1473 609832
Contact person: John Davies
E-mail: john.nj.davies@bt.com

Change Log

Vers.	Date	Author	Description
0.1	05/03/04	J. Esplugas	Starting draft
1.0	31/03/04	J. Esplugas	Deliverable

List of Acronyms and Abbreviations

Acronym/ abbreviation	Resolution
SWWS	Semantic Web enabled Web Services
SWWS-AA	SWWS project Abstract Architecture
WS	Web Services
B2Bi	Business-to-Business Integration
WSMF	Web Service Modeling Framework
CRM	Customer Relationship Management
SCM	Supply Chain Management
X.12	ASC Standard for EDI
PIP	Partner Interface Process
ASN	Advanced Shipment Notice
SSM	Shipment Status Message
POD	Proof of Delivery
FI	Freight Invoice
EDI	Electronic Data Interchange

Table of Contents

Table of Figures

1 Executive Summary

This document includes the in-deep analysis of the case study as a set of web services and which are the ontologies that semantically support those services. This document follows the Deliverable D8.1[1] which includes the Case Study requirements. The introduction will also give the framing of the services and materials presented inside the SWWS-AA.

There are two main sections that are covered, Services and Ontologies. The first part talks about the services that present the participants of the case study. We introduce the concept of agreement template and how this item becomes an important player in the discovery and matchmaking phase. The agreement template will turn into an agreement contract during that process. The client of the service to be discovered will put some constraints and will fill-up some fields of this contract template will help the discovery service or search agent discern which service provider could enter into the negotiation phase.

The negotiation phase, also mentioned as the pre-contractual phase is the time when parties will conciliate a concrete class of service to be delivered, the service client could held conversations simultaneously with different partners to arrange the best deal possible for him. Once the provider is selected, there is the post-contractual or execution phase.

After the services are described, we present how we would formulate the descriptions using a description language like OWL-S[2]. At the end of that section we will present which problems we have encountered by the use of this concrete technology. Briefly, we could mention the coupled relationship between the grounding and the WSDL[3] technology which may not be suitable for a scenario like ours; given the fact that some protocols like RosettaNet are already web protocols that may not need a WSDL layer. Another negative aspect of the use of OWL-S is the lack of possibility to represent vital information that characterizes the service in the profile section.

The second part of the document refers to the ontologies that we use in the case study to support the web services and the interaction between them. The agreement template has its representation on those supporting ontologies. Those ontologies will be shared between the service provider and the service consumer.

At the end of the document, the reader could find the annexes that include the OWL files that correspond to the ontologies presented. Those ontologies are in constant change and become more complex as the project advances, therefore the reader should not take them as a definitive representation of the domain.

2 Introduction

This document follows the document D8.1, presented as the Case study requirements. After presenting the requirements that the scenario was proposing in order to create this case study, this document will cover the specification of the interactions between the partners in this scenario in terms of Web-services. The main players of this scenario are a multi-leg logistic provider and a freight forwarder.

The purpose of the multi-leg logistic provider (aka logistic provider) is to find a suitable freight forwarder to complete a supply-chain that has been broken. The freight forwarder offers his service publishing its capabilities using a semantic web enabled service.

In alignment with SWWS Abstract Architecture, the case study scenario holds the following conceptualization:

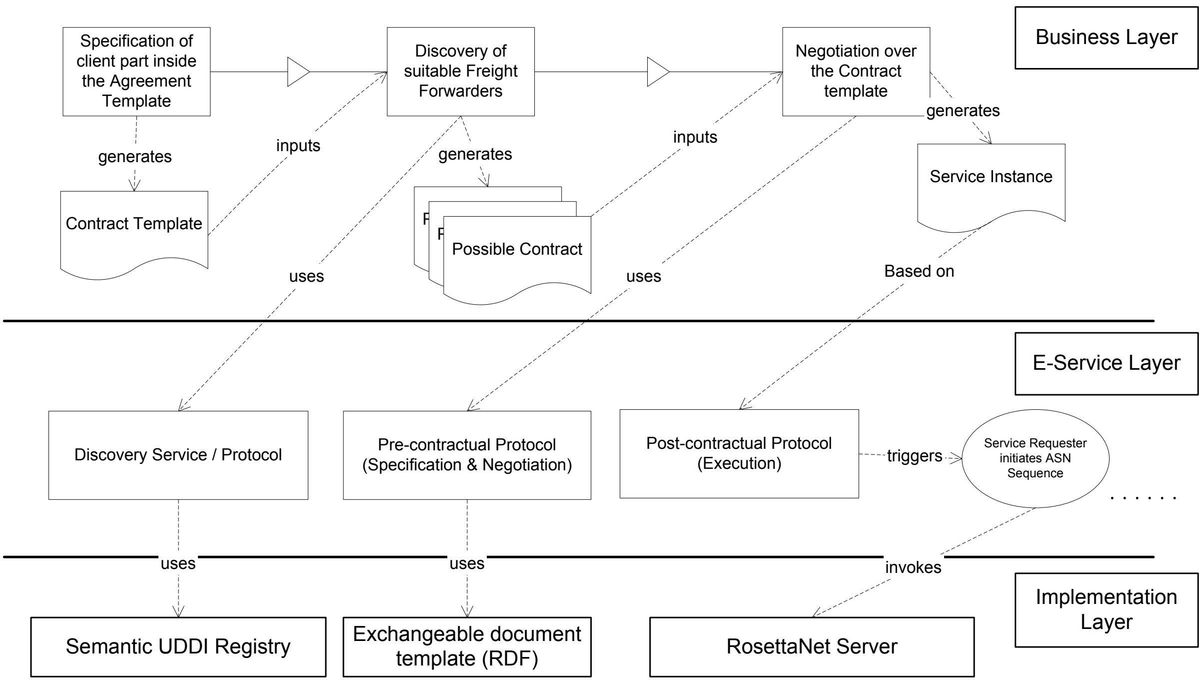

Figure 1 : Scenario view as a SWWS- AA

This document will give an in-depth analysis of the different stages related to the contract formation and the service execution. The second section of the document will refer to all the ontological support that this case study requires.

3 Services

This section will cover the specification of the services that will be deployed to support this case study. We will present the case study storyboard, for each of the steps; we will specify the impact that these stages have in the service and how it should be described.

3.1 Existing scenario storyboard

The reader could find a more detailed explanation about the scenario in the requirements document. Briefly, the situation is the following:

For an existing reason (optimization process, contract violation, etc) a logistic provider is about to be substituted. The multi-leg logistic provider stands as a communication broker between all partners. There is always 2 partner communications.

The Freight forwarder 2 is the component that is going to be substituted, and the following sections will present different aspects of the service substitution.

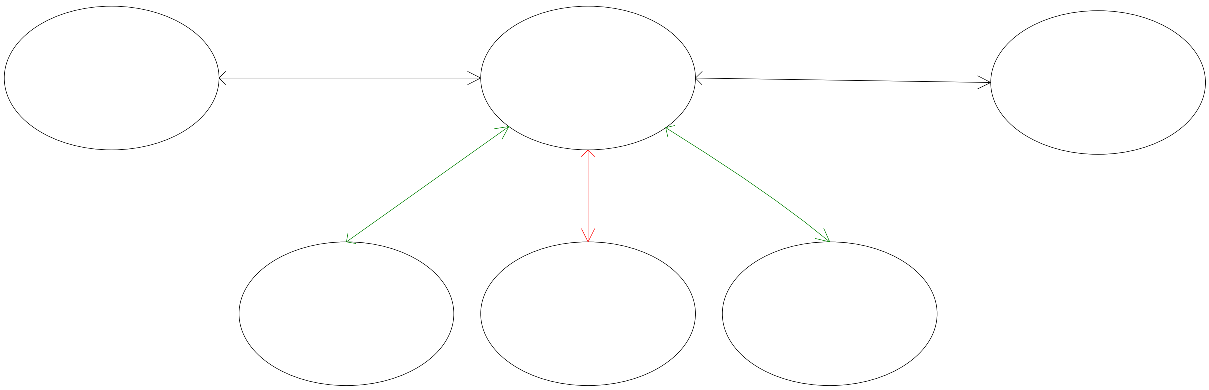

Figure 2: Actors involved in multi-leg logistic communications

1. Specification of the requirements of the
2. HP will use a discovery service to locate the possible candidates to that could enter into an agreement. (Discovery Phase)
3. This agreement or contract will be specified by HP and the service must acknowledge the exploration of a possible contract. (Contract Formation)
4. HP will examine the process model of the new logistic provider and will check if it could be integrated or not (Matchmaking). This process will be the following:
    a. Compare the process model with the existing partners in the communication chain
    b. Specify the Mediation guidelines
    c. Check the consistency of the ontology representation lifting and lowering messages.

Shipper Unit Multi-leg Logisti

The message sequence that the new logistic provider (with the help of a mediator) has to comply is presented in the case study requirements document as the following:

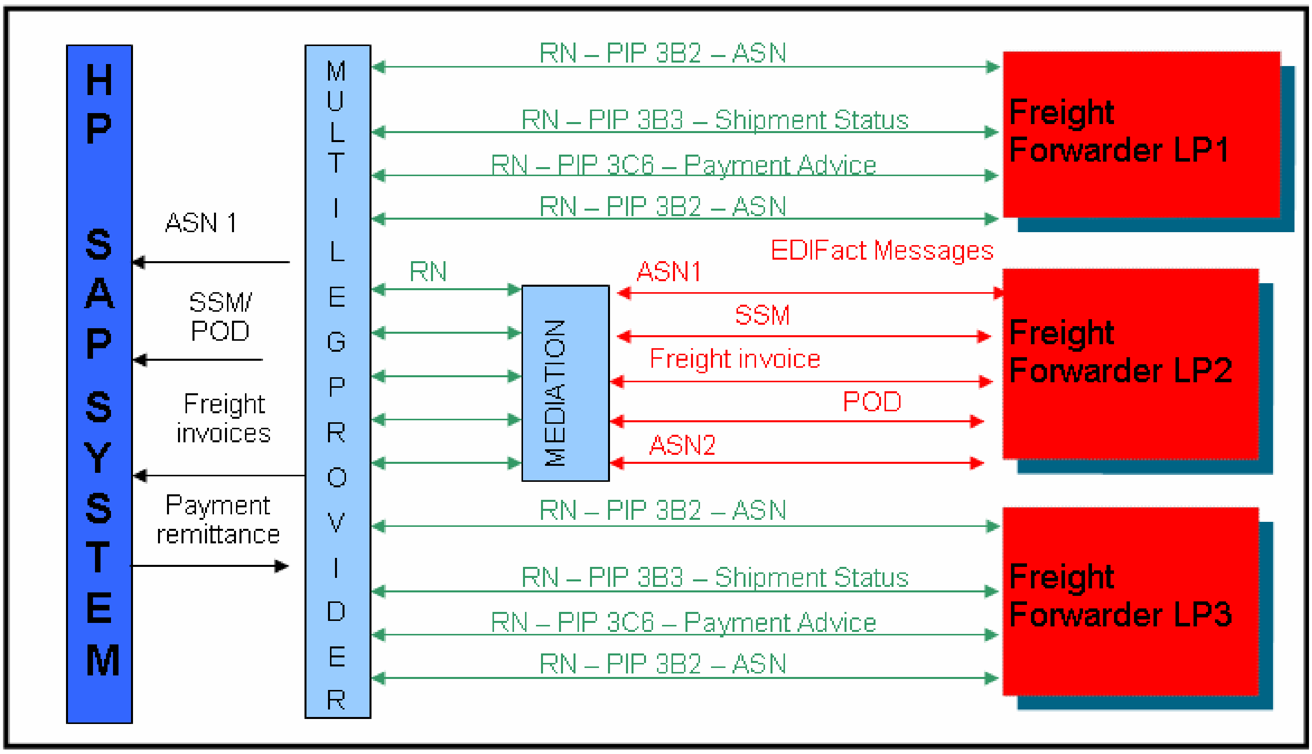

Figure 3 : Message exchange during logistic transactions

The substitution requirements will be specified in a single document, this document will be the Service Agreement. The service agreement will be built by both partners and will have certain degrees of freedom to allow specific flexibility.

3.2 The Service agreement template

The Service interaction always exists under an agreement of service, this form of contract may be explicit or implicit. In our case study, this agreement will be dynamically formed under a specific template. The template is defined as:

SERVICE AGREEMENT TEMPLATE	Terms or Service Details
	Payment
	Interaction

3.2.1 Terms or Service Details

The terms of transport will be all the information related to the transport of goods. It depends on the service offered; in this case, it is the description of the logistic terms.

Terms	Consumer	Provider	Supporting Ontologies	Dependencies	Consumer fixed?	Provider fixed?
Date and Time	Concrete Pair	Restricted List of possible	Date & Time	➢ Locations ➢ Pricing ➢ Window Delivery	YES	NO
Location	Concrete Pair	Restrictions on possible locations	Location Ontology	➢ Range ➢ Pricing	YES	NO
Package Size	List of Triple of Measures	Maximum measures	Domain Ontology	➢ Pricing	YES	NO
Package Weight	List of values	Maximum weight	Domain Ontology	➢ Pricing	YES	NO
Type of Service	List of values	List of values		➢ Pricing	NO	NO

3.2.2 Payment

The payment section relates to the economic valuation demanded to deliver the service.

Payment	Consumer	Provider	Supporting Ontologies	Dependencies	Consumer fixed?	Provider fixed?
Price	None or Request For Quotation.	Quotation	Payment Ontology		NO	YES
Means of Payment	List of accepted means	List of accepted means	Payment Ontology	➢ Preferences	NO	NO
Terms of Payment	List of accepted terms	List of accepted terms	Payment Ontology	➢ Preferences	NO	NO

3.2.3 Interaction

The terms of transport will be all the information related to the transport of goods. It depends on the service offered; in this case, it is the description of the logistic terms.

Interaction	Consumer	Provider	Supporting Ontologies	Dependencies	Consumer fixed?	Provider fixed?
General Sequence	Protocol FSM	Protocol FSM			YES	YES
ASN	List of accepted protocols	List of accepted Protocols	Domain Ontology	➢ Standard Protocol Overrides	YES	YES
SSM	List of accepted protocols	List of accepted Protocols	Domain Ontology	➢ Standard Protocol Overrides	YES	YES
FI	List of accepted protocols	List of accepted Protocols	Domain Ontology	➢ Standard Protocol Overrides	YES	YES
POD	List of accepted protocols	List of accepted Protocols	Domain Ontology	➢ Standard Protocol Overrides	YES	YES

3.3 Pre-contractual phase

The main purpose of this pre-contractual phase is the matchmaking exploration of the possible candidates in terms of service selection. The pre-contractual phase is where all the terms specified in the service agreement will be settled and approved. In this pre-contractual phase, the Service consumer will present the request of service in a form of a contract. The service consumer will initiate 1 to 1 communications to concrete the open clauses of the contract, once a satisfactory resolution is achieved, this concrete service provider will be elected and a concrete service will be instantiated. Generally, the pre-contractual phase could be divided into those three steps:

- Presentation by service consumer the agreement template with explicit preferences on the flexible options.
- The negotiation between the flexible elements of the parties regarding the agreement.
- Description of the post-contractual arena which includes the message patterns and the service instantiation related data.

3.3.1 Negotiation over the service agreement template

Prior to the agreement to provide a concrete service there are some elements that are not fixed by the participating partners, either because they have certain levels of freedom (i.e. Means of Payment) or either because of the nature of the parameter (i.e. Date & Time). The non-fixed terms of the service agreement are depending on the parties involved.

The negotiation over the service template will be done in three steps; the first one is the Service provider conciliation. The consumer of the service has some constraints about the service to be delivered (as dates, locations, etc), it will check with the restrictions imposed by the service provider that the service requested would be within the range of the services offered.

Once the service requested matches the services offered, the Service consumer should check and accept any imposition that the service provider would impose. This is the Service consumer conciliation and will include things like non-functional requirements.

When both parties are happy with the impositions of the counterpart, they start a negotiation process where they will agree on the values that both are willing to negotiate.

3.3.1.1 Service Provider conciliation

Those are the elements fixed by the Service consumer that are flexible in the terms of the service provider, in the case of our template the only existing example is the location.

Terms	Consumer	Provider	Supporting Ontologies	Dependencies	Consumer fixed?	Provider fixed?
Location	Concrete Pair	Restrictions on possible locations	Location Ontology	➢ Range ➢ Pricing	**YES**	**NO**
Package Size	List of Triple of Measures	Maximum measures	Domain Ontology	➢ Pricing	**YES**	**NO**
Package Weight	List of values	Maximum weight	Domain Ontology	➢ Pricing	**YES**	**NO**

The maximum values presented to the consumer by the provider are fixed values, the conciliation process on the service consumer side is basically checking if the presented values on the agreement are within the range of accepted values by the service provider.

For each of the existing elements to be shipped

Conciliation step 1

C: Package_Size ≡ <width, height, depth>

P: Max-measures ≡ <Max-width, Max-height, Max-depth>

IF Max-width ≥ width ⊓ Max-height ≥ height ⊓ Max-depth ≥ depth *THEN* step2
ELSE *raise violation*

Conciliation step 2

C: Package_Weight ≡ <weight>

P: Max-weight ≡ <Max-weight>

IF Max-weight ≥ weight *THEN* continue process

ELSE raise violation

next element

The consumer will present a concrete pair of values (Source and destination) and the service provider conciliation consist in checking the restrictions over the locations proposed to see if they are within the accepted values or not.

3.3.1.2 Service Consumer conciliation

Those are the elements fixed by the Service providers that are flexible in the terms of the service consumer, in our case it is only the price:

Payment	Consumer	Provider	Supporting Ontologies	Dependencies	Consumer fixed?	Provider fixed?
Price	None or Request For Quotation.	Quotation	Payment Ontology		**NO**	**YES**

In this case, the price will be a proposal of prices related to the open terms of the negotiation. The service provider offers at this stage a whole range of different products as a service, each different service will have a different price. The Service consumer conciliation does not exist in this scenario as the price acceptance is done in the following step in the negotiation.

3.3.1.3 Multi-party conciliation

Once there has been an agreement on the single party conciliation (both client and provider), the parties will explore the possibility of reaching an agreement on the double-opened issues. Those are the elements un-fixed by both parties, in the case of our agreement template these are those parameters.

Payment	Consumer	Provider	Supporting Ontologies	Dependencies	Consumer fixed?	Provider fixed?
Type of Service	List of values	List of values		➢ Pricing	**NO**	**NO**
Means of Payment	List of accepted means	List of accepted means	Payment Ontology	➢ Preferences	**NO**	**NO**
Terms of Payment	List of accepted terms	List of accepted terms	Payment Ontology	➢ Preferences	**NO**	**NO**

The “Type of service” is related to the *Incoterms* of service. The Incoterms are internationally accepted commercial terms defining the respective roles of the buyer and seller in the arrangement of transportation and other responsibilities and clarify when the ownership of the merchandise takes place. Those terms are:

- EXW - Ex Works -- Title and risk pass to buyer including payment of all transportation and insurance cost from the seller's door. Used for any mode of transportation.
- FCA - Free Carrier -- Title and risk pass to buyer including transportation and insurance cost when the seller delivers goods cleared for export to the carrier. Seller is obligated to load the goods on the Buyer's collecting vehicle; it is the Buyer's obligation to receive the Seller's arriving vehicle unloaded.
- FAS - Free Alongside Ship --Title and risk pass to buyer including payment of all transportation and insurance cost once delivered alongside ship by the seller. Used for sea or inland waterway transportation. The export clearance obligation rests with the seller.
- FOB - Free On Board and risk pass to buyer including payment of all transportation and insurance cost once delivered on board the ship by the seller. Used for sea or inland waterway transportation.
- CFR - Cost and Freight -- Title, risk and insurance cost pass to buyer when delivered on board the ship by seller who pays the transportation cost to the destination port. Used for sea or inland waterway transportation.
- CIF - Cost, Insurance and Freight -- Title and risk pass to buyer when delivered on board the ship by seller who pays transportation and insurance cost to destination port. Used for sea or inland waterway transportation.

- CPT - Carriage Paid To -- Title, risk and insurance cost pass to buyer when delivered to carrier by seller who pays transportation cost to destination. Used for any mode of transportation.
- CIP - Carriage and Insurance Paid To --Title and risk pass to buyer when delivered to carrier by seller who pays transportation and insurance cost to destination. Used for any mode of transportation.
- DAF - Delivered at Frontier -- Title, risk and responsibility for import clearance pass to buyer when delivered to named border point by seller. Used for any mode of transportation.
- DES - Delivered Ex Ship -- Title, risk, responsibility for vessel discharge and import clearance pass to buyer when seller delivers goods on board the ship to destination port. Used for sea or inland waterway transportation.
- DEQ - Delivered Ex Quay (Duty Paid) -- Title and risk pass to buyer when delivered on board the ship at the destination point by the seller who delivers goods on dock at destination point cleared for import. Used for sea or inland waterway transportation.
- DDU - Delivered Duty Unpaid -- Title, risk and responsibility of import clearance pass to buyer when seller delivers goods to named destination point. Used for any mode of transportation. Buyer is obligated for import clearance.
- DDU - Delivered Duty Unpaid -- Seller fulfils his obligation when goods have been made available at the named place in the country of importation
- DDP - Delivered Duty Paid -- Title and risk pass to buyer when seller delivers goods to named destination point cleared for import. Used for any mode of transportation.

All those Incoterms will be defined in a standard ontology of Incoterms and therefore accepted from both parties. EXW, CPT, CIP, DAF, DDU and DDP are commonly used for any mode of transportation. FAS, FOB, CFR, CIF, DES, and DEQ are used for sea and inland waterway. Given the fact that those Incoterms are so widely accepted, they will be also part of the application layer above the semantic services, therefore, being self-explanatory, we consider no necessary to build an exclusive ontology to support the contract agreement.

The “Means of Payment” refer to the kind of payment could be executed; those concepts will exist in an ontology including:

- Bank transaction
- Cash
- Cheque
- Credit card
- …

Different means of payment will mean different prices; some of them may not be available depending on the type of service offered and also depending on the terms of the payment, and therefore all the items should be negotiated jointly.

The “Terms of Payment” refer about how the payment should be executed and it will include things like:

- Payment before collection
- Payment on collection
- Payment before delivery
- Payment on delivery
- Payment up to 30 days after delivery
- Payment up to 60 days after delivery
- …

There could be different ways of exploring how the matchmaking could be done in a optimal way using the minimum number of steps. This case study does not pretend to dig this possibility. We will assume that the Service provider will offer the complete range of available options including an associated price. The service consumer will evaluate and compare between providers the concrete “service / term of payment / means of payment / price” cluster.

Once a selection of a concrete service is done, the service consumer and the selected party; will enter the following stage, the contract execution description.

3.3.2 Description of contract execution

After both parties have reached an agreement concerning the terms of the service; both parties must agree in the pre-contractual stage how are they going to connect.

The sequencing of operations will be expressed in irrefutable form accepted from both parties. This agreement will include the constraints of the conversation, in conjunction with the service description of the services (in OWL-S or any other form of formalization). This information will be enough to assure the possibility and smoothness of the integration.

We will illustrate now how to build the description of the contract execution. This is roughly the sequencing of messages. It should be described in an indubitable way. The correct message invocation (or reception) order is guided by the events during the life-cycle:

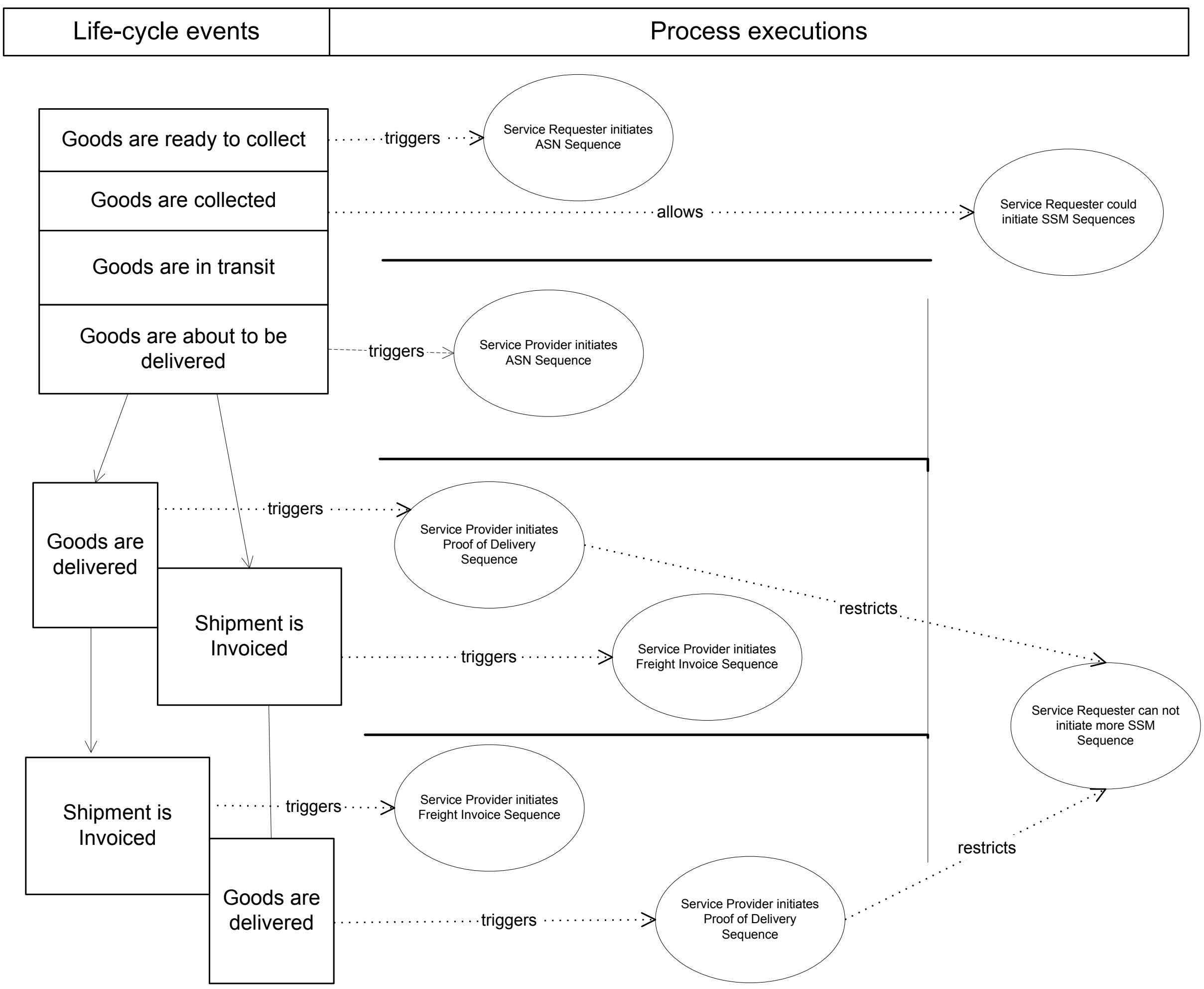

Figure 4 : Life-cycle events and its associated processes

Awaiting action 1: An ASN message will be initiated from the Service Requester. Once items are collected (no process captures this event), the Service Requester could initiate any SSM sequences.

Active phase 2: An ASN message will be initiated from the Service Provider once the conversation of the previous ASN is finished.

Active phase 3: Once the ASN sequence of the previous phase is finished, the service provider can initiate either a POD or a FI Sequence.

Active phase 4 (possibly concurrent with the previous one): The service provider initiates the second sequence (e.g. FI sequence if a POD was started).

Once POD Sequence is finished, the service requester can not initiate any more SSM sequences and the communication for this concrete transaction is finished.

Each of the message exchange sequence (ASN, POD, FI and SSM) is a two way messaging with no assumptions on security or delivery assumptions.

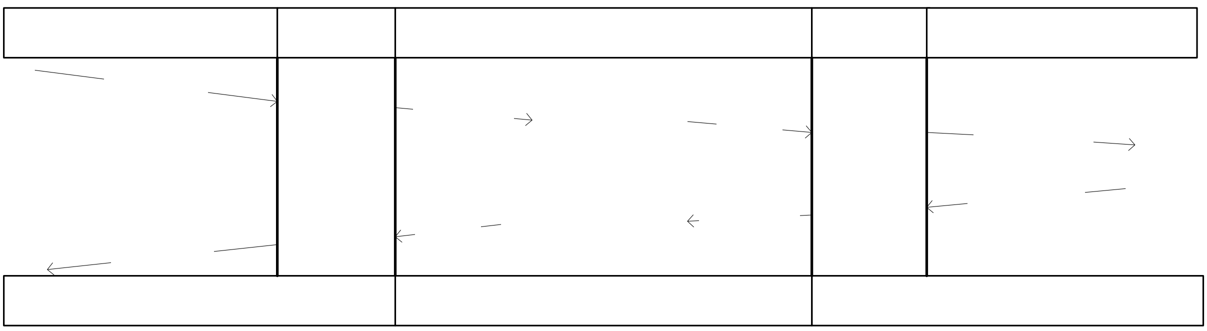

Figure 5 : ASN message exchange protocol

The ASN, SSM, POD and FI are E-Services message exchange processes. Those message exchange processes will be merged into an abstract state machine that will command the actions defining the conversation behaviour.

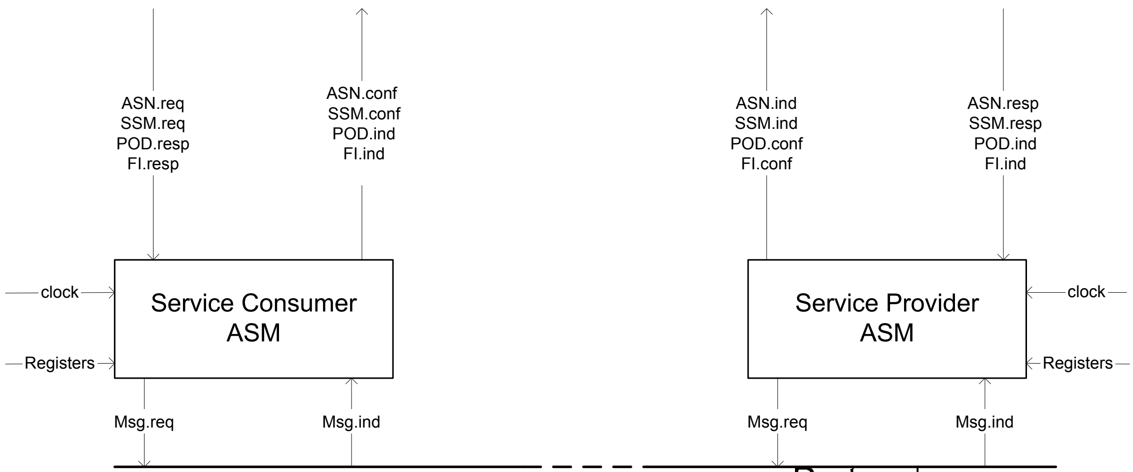

Figure 6 : Abstract States machine abstraction

Application Layer

Protocol Engine

Netw

ASN.request

msg.request

The conversation behaviour modelled as a state machine will be then transformed in process algebra. Once these conversations behaviours are in process algebra, the descriptions of the process could be represented using a semantic mark-up language as OWL-S or WSMO.

Apart from the conversation behaviour, there is also the need to specify a common environment to establish the communication. To define a common environment, partners will need to:

- Share a common vocabulary of concepts.

  This is the role taken by the ontology, both parties will share the semantics of concepts like ASN or POD, they will be understood in a unique form.

- Action vocabulary

  The action vocabulary will be specified between the semantic mark-up and the ontologies, actions like message request will have a concrete translation that will be the same regardless the partner that manipulates this expression,

- Compatibility in terms of sequence of actions

  The sequence of actions from one partner to the other has to be either directly compatible (no protocol mediation needed) or compatible by mediation. Each of the partners should know which scope of protocols could bring the use of mediators for each of the messages they have to interact with.

The union of the conversation behaviour and the common environment defines the protocol of interaction between both partners, also referred in the SWWS-AA as the post-contractual protocol. This protocol has to be satisfactory and accepted by both partners before the execution of the contract.

Now we will present the creation of the descriptions of the process, from the state machines to the algebras, its translation into OWL-S descriptions are presented in the section 3.5

The State machines get complicated extremely if we include inside the conversation behaviour also the SSM sequence message. Therefore the SSM sequence has been taken out from the main thread of communication (ASN + ASN+ (FI+POD)) in the state machine. The SSM sequence could be triggered from the end of the first ASN until the moment the POD is confirmed. Although being out of the state machine, the SSM has been represented in the process algebra, so the modelling of the processes is complete. The reader could find the SSM behaviour in terms of a state machine in the indexes as source for consultation.

Figure of the State machine presented by the Service Consumer

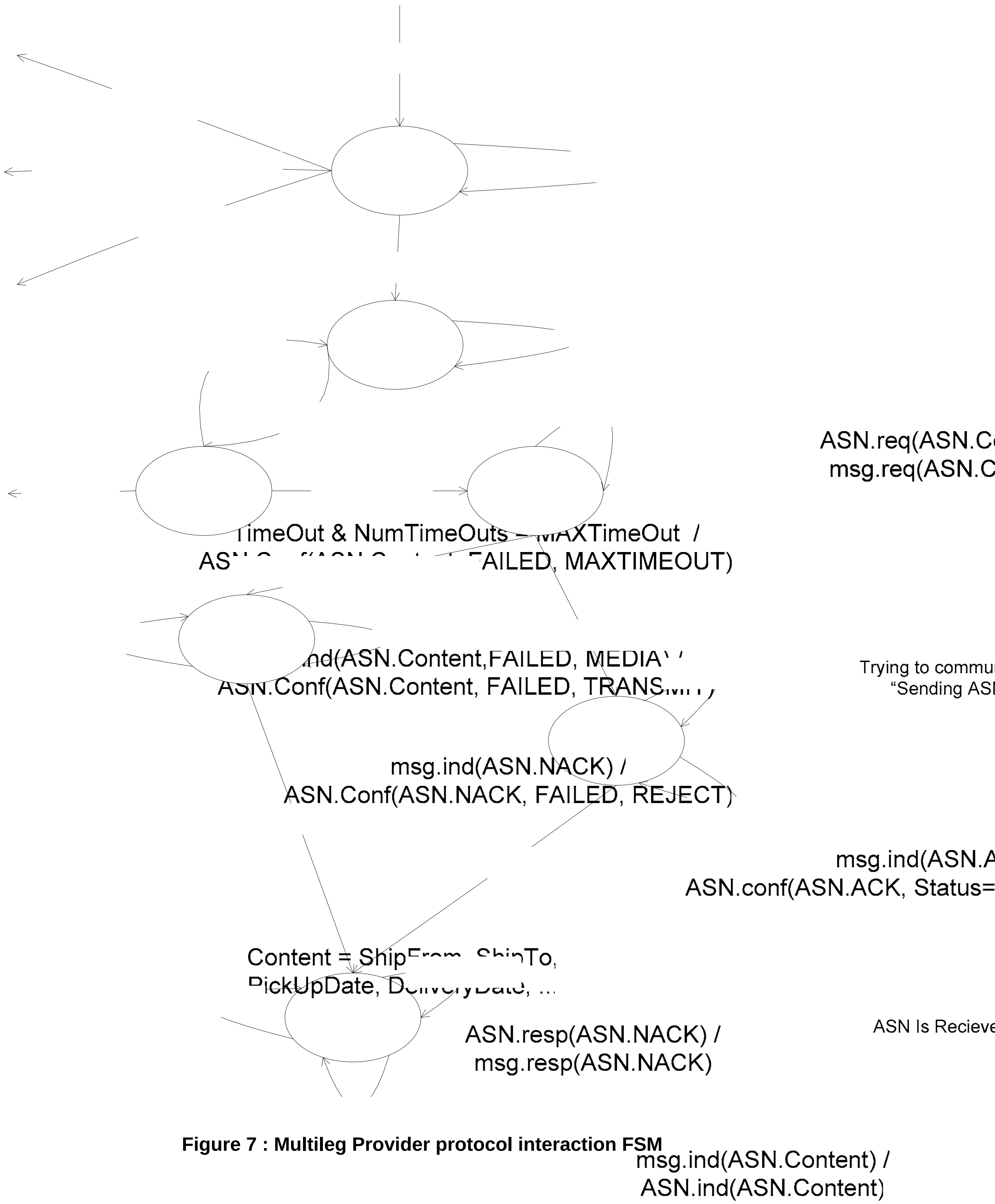

Figure 7 : Multileg Provider protocol interaction FSM

The formalization of the state machine in terms of CCS process algebra is as follows:

P0: send_asn.ASN_1_SENT

ASN_1_SENT: receive_asn.ASN_2_RECEIVED ¦ send_ssm.SSM_SENT

ASN_2_RECIEVED: ((receive_pod.POD_RECEIVED+
+receive_fi.FI_RECEIVED) ¦ send_ssm. SSM_SENT)

POD_RECEIVED: receive_fi.END

FI_RECEIVED: receive_pod.END ¦ send_ssm. SSM_SENT

SSM_SENT: send_ssm. SSM_SENT+0.

END: terminate_communication.0

This process algebra has the following UML representation:

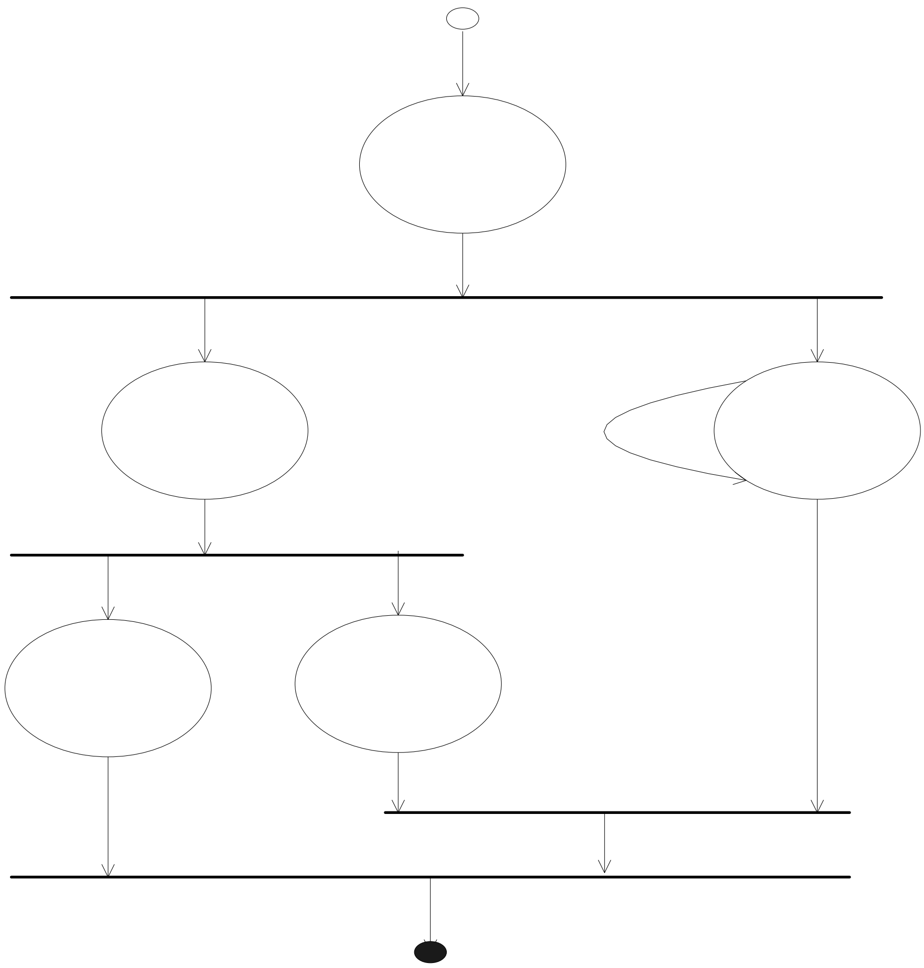

Figure 8 : UML Representation of Process invocation (Service Consumer)

Once we have this representation we could express this process model in any suitable mark-up language like OWL-S or WSMO, please refer to section 3.6 for the OWL-S description.

ASN_1

Figure of the State machine presented by the Service Provider

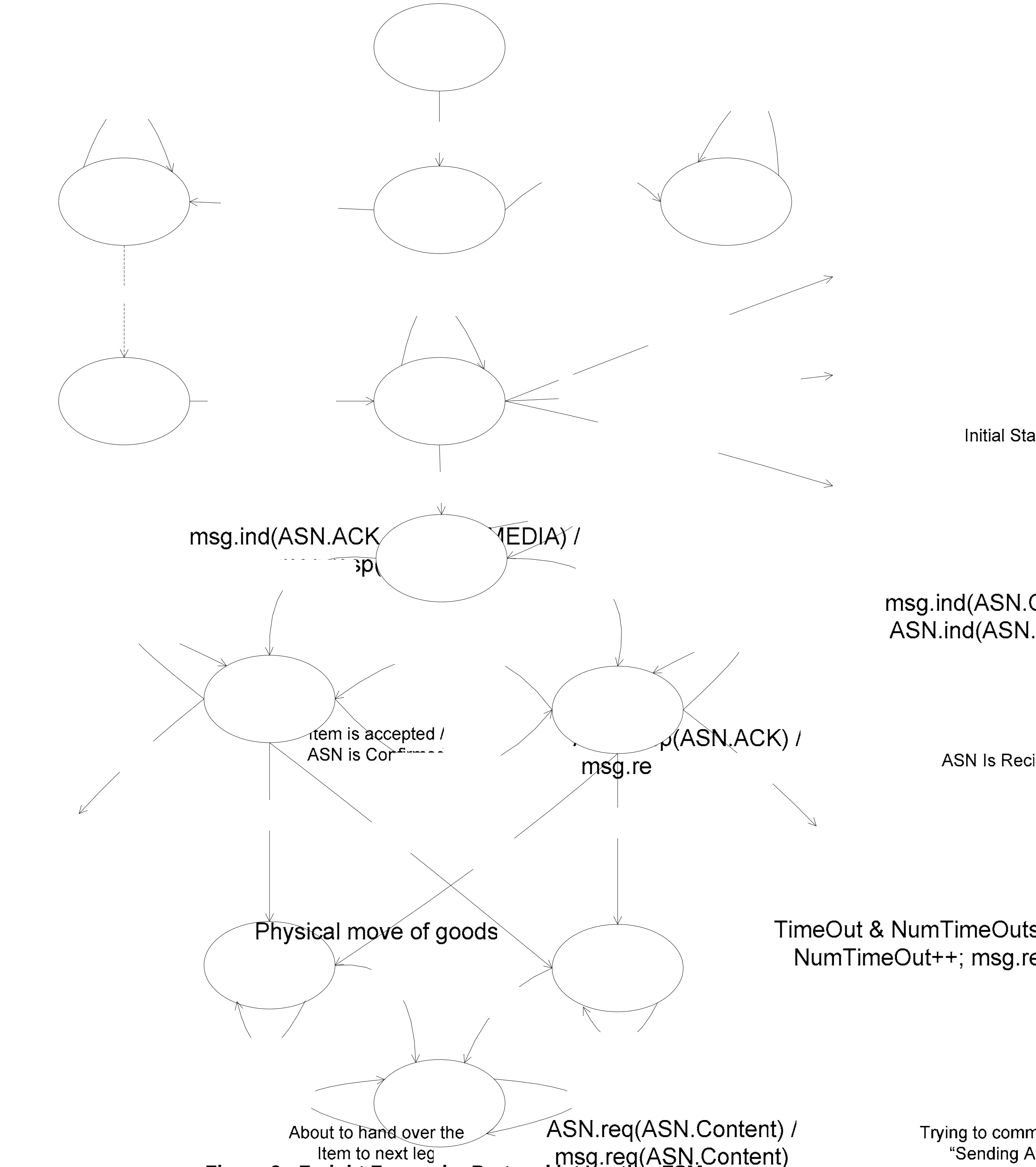

Figure 9 : Freight Forwarder Protocol interaction FSM

The formalization of the state machine in terms of CCS process algebra is as follows:

P0: receive_asn.ASN_1_RECEIVED

ASN_1_RECEIVED: send_asn.ASN_2_SENT ¦ received_ssm.SSM_RECEIVED

ASN_2_SENT: ((send_pod.POD_SENT+

+send_fi.FI_SENT) ¦ received_ssm. SSM_RECEIVED)

POD_SENT: send_fi.END

FI_SENT: send_pod.END ¦ received_ssm. SSM_RECEIVED

SSM_RECEIVED: received_ssm. SSM_RECEIVED +0

END: terminate_communication.0

This process algebra has the following UML representation:

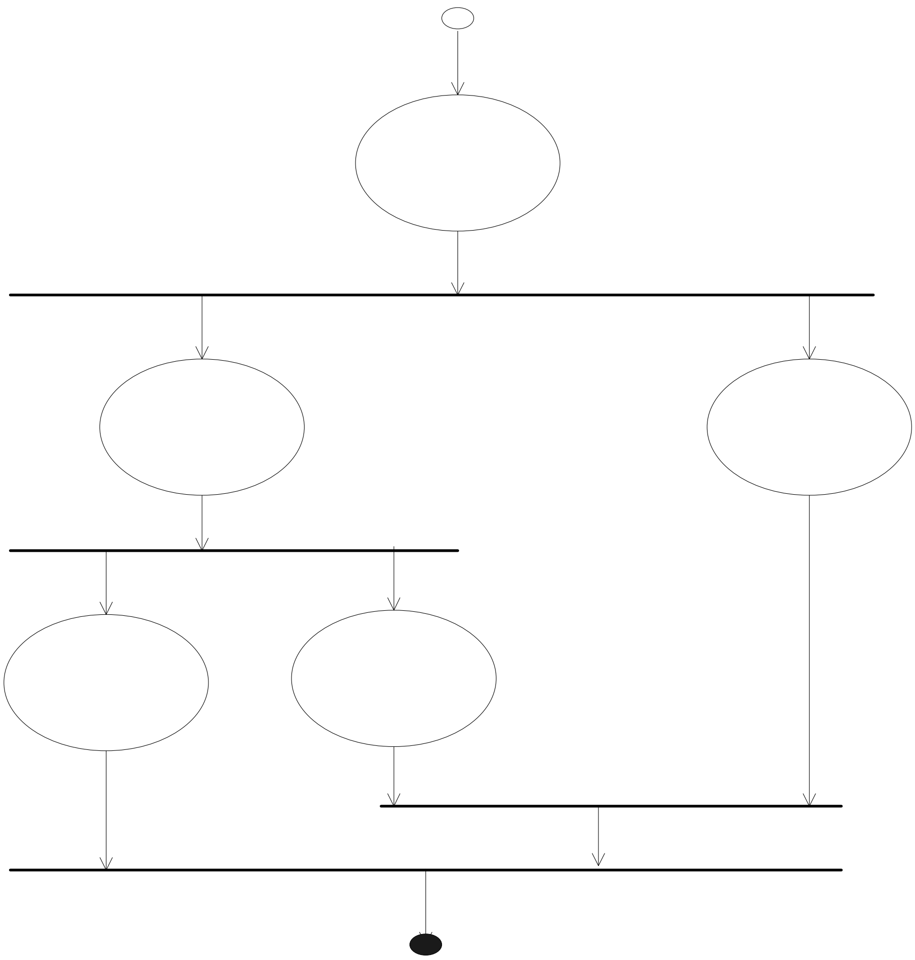

Figure 10 : UML representation of Process invocation (Service Provider)

As also stated in the process analysis of the provider, once we have this diagram we are in a position of use a mark-up language to create the semantic descriptions of those services.

ASN_1_R

3.4 Post-contractual phase

The post-contractual phase starts once the protocols of interaction have been defined and the services are instantiated. Once the service instance is created, if any of the partners has to implement a mediator, this entity should be specified and established. If we follow the case scenario picture presented in the storyboard description:

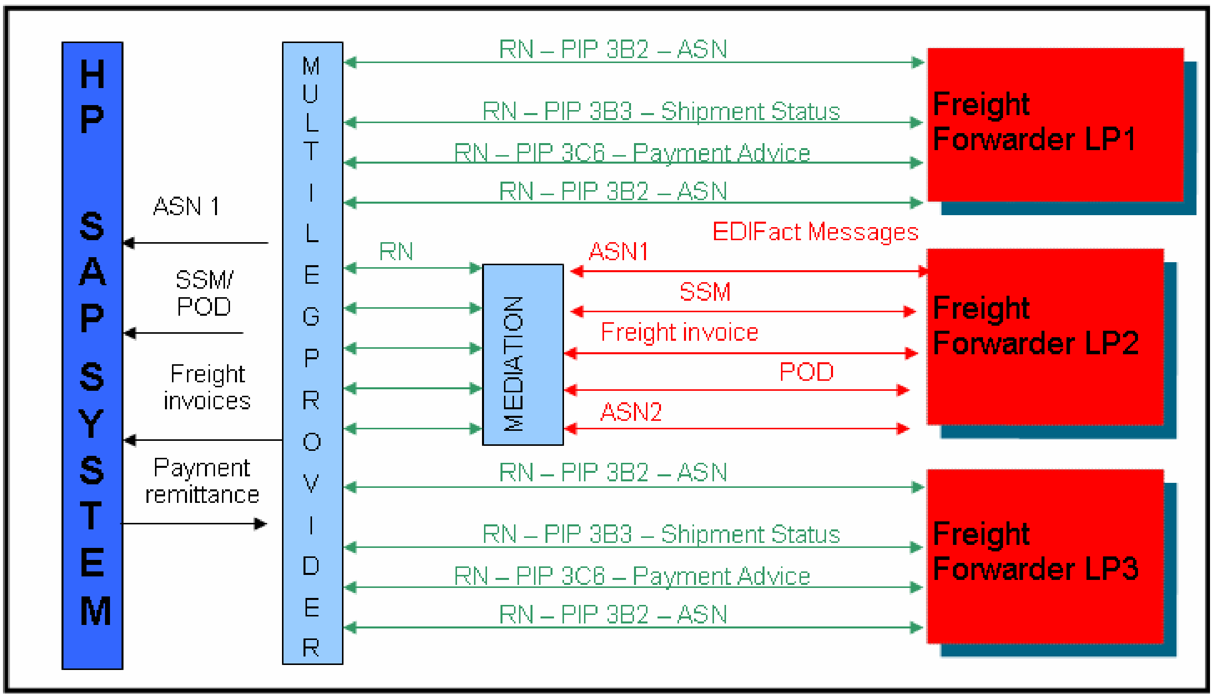

Figure 11 : Message exchange during logistic transactions

We clearly see that there is a need for mediation between the messages of EDIFact and RosettaNet. We will present here (as presented also in D8.1) how this mediation is solved in the case of the ASN.

In RosettaNet, this action is performed by the use of a concrete PIP, the PIP 3B2 is defined as:

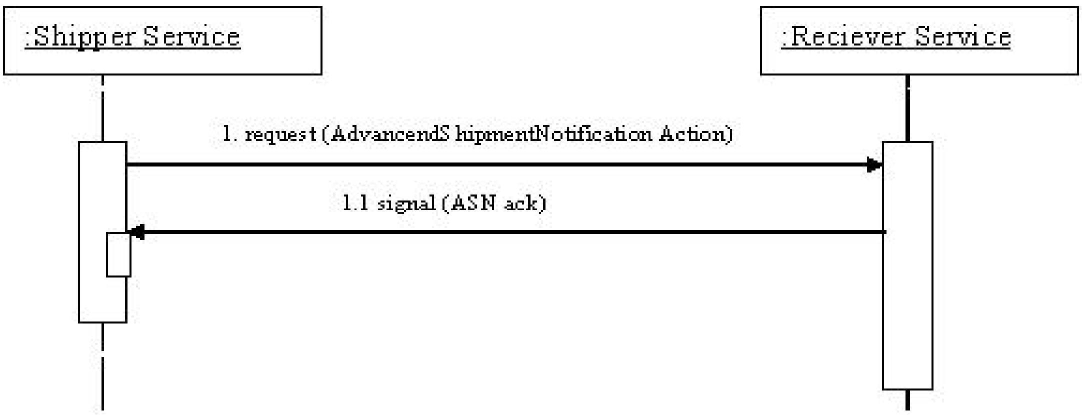

Figure 12 : Pip 3B2 - ASN in RosettaNet

In EDIFACT the action of sending a concrete notification of shipment is done using a concrete message called DESADV. We can imagine a company (Receiver Service) that requires a 3 hand-shaking confirmation.

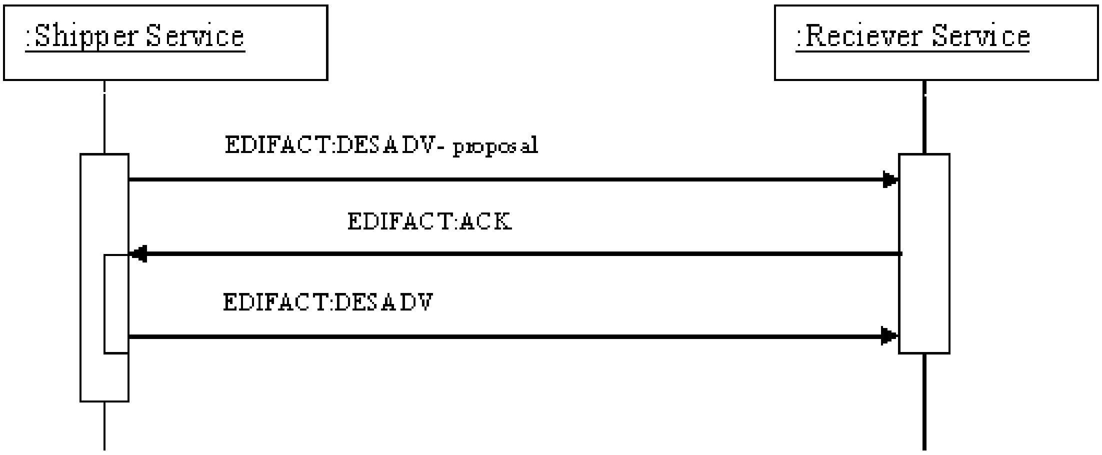

Figure 13 : Possible EDIFact implementation of an ASN

The Protocol mediation lays in the component that enables the communication between those two partners even if they do not match completely on the message exchange sequences. This is a scenario of protocol mediation as none of the partners had to modify their current interfaces, and the mediator is clearly a transparent component in the communication.

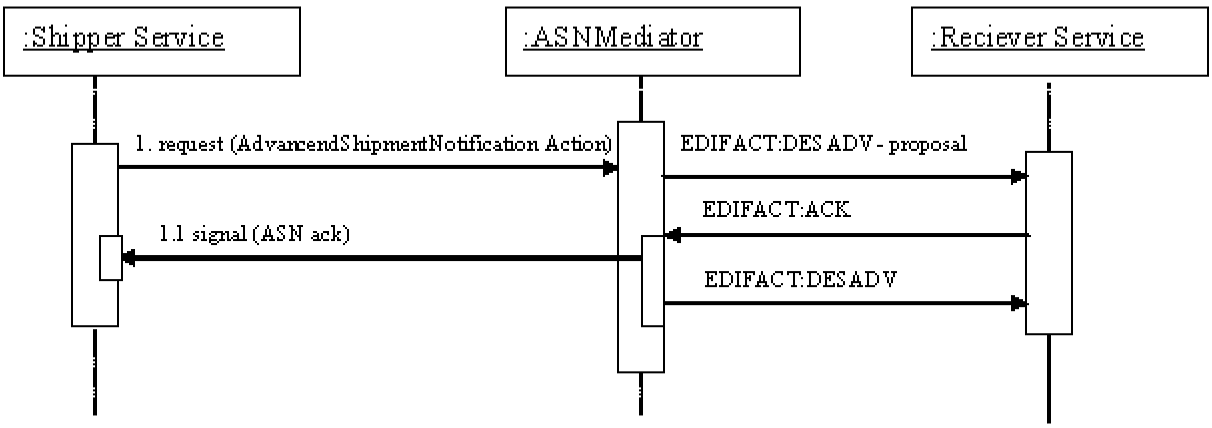

Figure 14 : Use of mediators in Post-Contract protocol

The sequence diagrams of the other transactions (SSM, POD, FI) could be found in the OWL-S annotation of the services.

3.5 OWL-S Descriptions for the services

The Owl-s Model of the Service has 4 main parts: The Service Model, The Service Profile, the Process Model and the Grounding. The first analysis is made under the assumption that a single service deals with a single protocol and multiple messages.

3.5.1 SERVICE MODEL

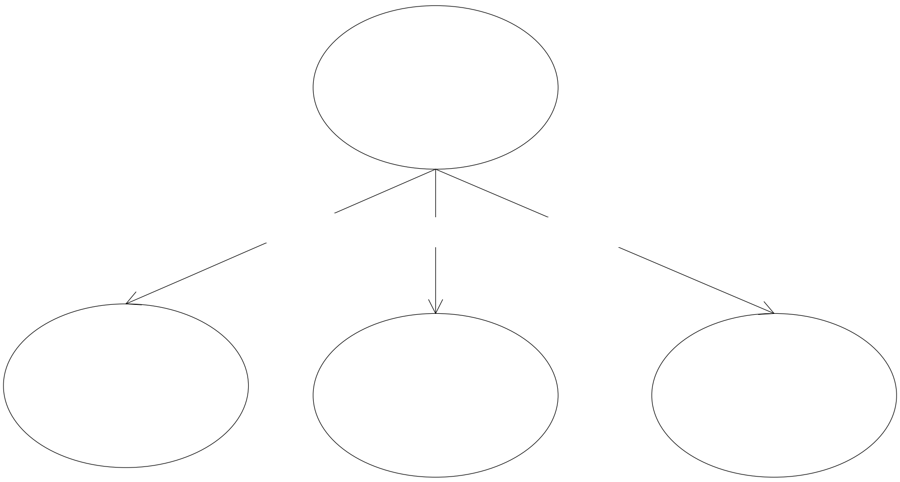

Figure 15 : OWL-S Service Model Diagram

```
  </ServiceTaxonomy:Logistic>

</rdf:RDF>

  <service:Service rdf:ID="Logistic">

    <!-- Reference to the Logistic Profile -->
    <service:presents rdf:resource="#LogisticProfile"/>

    <!-- Reference to the Logistic Process Model -->
    <service:describedBy rdf:resource="#LogisticProcess"/>

    <!-- Reference to the Logistic Grounding -->
    <service:supports rdf:resource="#LogisticGrounding"/>

 </service:Service>
```

The Service is nothing more than a formal definition of the Service, each of the components holds different responsibilities:

- Profile: The profile of the service presented will include all information needed in Discovery phase (Please refer to D.8.1 in the Conceptual Architecture Chapter).
- Process Model: The Process Model includes the modelling of the conversations and interaction
- Process Grounding: Presents which are the actual connections to the service.

3.5.2 Service Profile

The Service profile in OWL- S presents information about non-functional properties of the service like:

- Service name.
- Service Category.
- Contact information.
- Service provider ratings.
- Inputs, Outputs, Preconditions and Effects.

```
<!--  The logistic Service  -->

 <ServiceTaxonomy:Logistic rdf:ID="LogisticProfile">
   <!-- reference to the service specification -->
   <service:presentedBy rdf:resource="&logisticService"/>

   <profile:serviceName>Logistic_B2B_Service </profile:serviceName>

   <profile:textDescription>
   This service offers the "standard" Logistic communication (ASN1+SSM+ASN2+POD+FI)
   </profile:textDescription>

   <!-- specification of contact information.   -->
   <profile:contactInformation>
     <profile:Actor rdf:ID="Contact-information">
       <profile:name>Javier Esplugas</profile:name>
       <profile:title>R&D Engineer</profile:title>
       <profile:email>javesp@hpl.hp.com</profile:email>
       <profile:webURL>http://hpl.hp.com/</profile:webURL>
     </profile:Actor>
   </profile:contactInformation>
```

The Pre-conditions at this level are based on the Inputs and the Outputs of the Service in a high-level. One of the existing problems with these preconditions at the profile level is the fact that a concrete input could fulfil different roles in different parts of the process

E.g. the process model could contain the following statement:

If (name= void) then Input_value = DUNS Number

[…]

(* Some messages are identified by the DUNS number of the Sender, therefore the DUNS is a valid entry)

The process model could contain also the following statement:

If (name= void) then raise exception.

(* If a message is a Invoice, it has to contain always the name of the receiver for legal issues)

The Preconditions at the profile level over an input #name of type string can not provide us the right tools to model precisely the behaviour.

This would be the classical input definition for the ASN process.

```
<profile:input>
  <profile:ParameterDescription rdf:ID="ASN.RN.Request">
    <profile:parameterName> ASN1_Input </profile:parameterName>
    <profile:restrictedTo rdf:resource="&xsd;#string"/>
    <profile:refersTo rdf:resource="&logisticProcess;#ASN_1_RECEIVED"/>
  </profile:ParameterDescription>
</profile:input>
```

This would be the output definition for the ASN process.

```
<profile:output>
  <profile:ParameterDescription rdf:ID="ASN.RN.Response">
    <profile:parameterName> ASN1_Output </profile:parameterName>
    <profile:refersTo rdf:resource="&logisticProcess;# ASN_1_RECEIVED"/>
  </profile:ParameterDescription>
</profile:output>
```

A part from the information presented by the OWL-S Profile, a logistic Service has to provide also (to allow effective discovery) the following information included in the agreement template:

- Date & Time Information
- Geographic Information, but not only about the geographic radius of the service, it is needed a effective way of presenting the different Sources and destinations of the service.
- Volumes to be treated.
- Range of Services (Incoterms related activities)
- EDI Capabilities – not all the logistic providers offer all the ranges of protocols and messages of those protocols to interact with.

3.5.3 Process Model

The OWL-S process model will be developed taking into account the behaviour expected by the logistic protocols described in sections 3.4.1 and 3.4.2. As we are defining the services itself, we will take the UML diagram that corresponds to the Service Provider. This figure was:

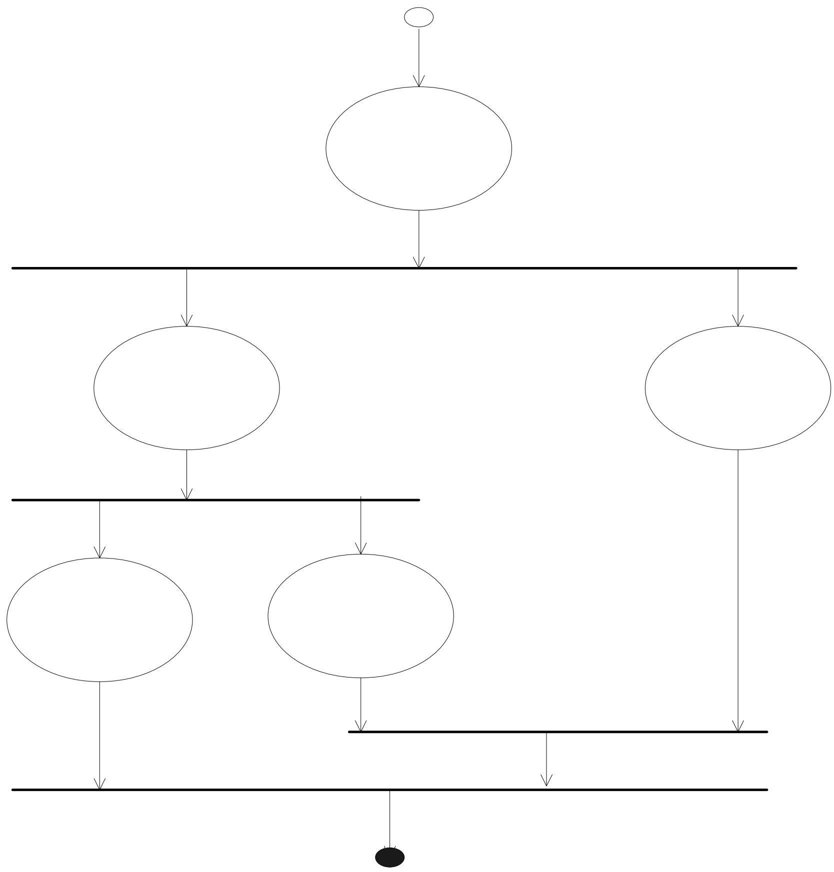

Figure 16 : UML Representation of Process Invocations (Service Provider)

ASN_1_R

The OWL-S process model has to be constructed to offer the behaviour expressed in this diagram.

If we analyse the diagram in terms of OWL-S constructs we will see that the main construct is a sequence. This sequence is composed by an composite process (ASN_1_RECEIVED) and a second composite process, this second composite process is a split between a sequence and another composite process (SSM_RECEIVED). That sequence is a sequence of a composite process (ASN_2_SENT) plus another composite process and this last composite process is a unordered execution of two composite processes (FI_SENT) & (POD_SENT).

ASN_2_SENT

In machine-readable OWL-S this is the Service Process:

```
<process:ProcessModel rdf:ID="LogisticProcess">
    <service:describes rdf:resource="&logisticService;#PurchaseService"/>
    <process:hasProcess rdf:resource="#State0"/>
</process:ProcessModel>

<!-- state 0 -->

<process:CompositeProcess rdf:ID="State0">
    <process:composedOf>
        <process:Sequence>
            <process:components rdf:parseType="Collection">
                <process:CompositeProcess rdf:ID="ASN_1_RECEIVED"/>
                <process:CompositeProcess rdf:ID="State1">
            </process:components>
        </process:Sequence>
    </process:composedOf>
</process:CompositeProcess>

<!-- state 1-->

<process:CompositeProcess rdf:ID="State1">
    <process:composedOf>
        <process:Split>
            <process:components rdf:parseType="Collection">
                    <process:AtomicProcess rdf:about="#SSM_RECEIVED"/>
                    <process:Sequence>
                        <process:components rdf:parseType="Collection">
                            <process:CompositeProcess rdf:ID="ASN_2_SENT"/>
                            <process:CompositeProcess rdf:ID="State2">
                        </process:components>
                    </process:Sequence>
            </process:components>
        </process:Split>
    </process:composedOf>
</process:CompositeProcess>

<!-- state 2-->

<process:CompositeProcess rdf:ID="State2">
    <process:composedOf>
        <process:Unordered>
            <process:components rdf:parseType="Collection">
                    <process:CompositeProcess rdf:ID="POD_SENT"/>
                    <process:CompositeProcess rdf:ID="FI_SENT"/>
            </process:components>
        </process:Unordered>
    </process:composedOf>
</process:CompositeProcess>
```

Check that the first Composite process main construct is a Split, so according to OWL-S 1.0 specification *"No further specification about waiting or synchronization is made at this level"*. This may generate a problem, as SSM messages are only valid until the POD is acknowledged, once the POD is acknowledged the Process control flow must not allow any further SSM to be treated. In this case, OWL-S is not enough to restrict the control flow in such way.

3.5.3.1 Process ASN_1_RECEIVED

The Advance Shipment Notice Process PIP3B2 is defined by RosettaNet as:

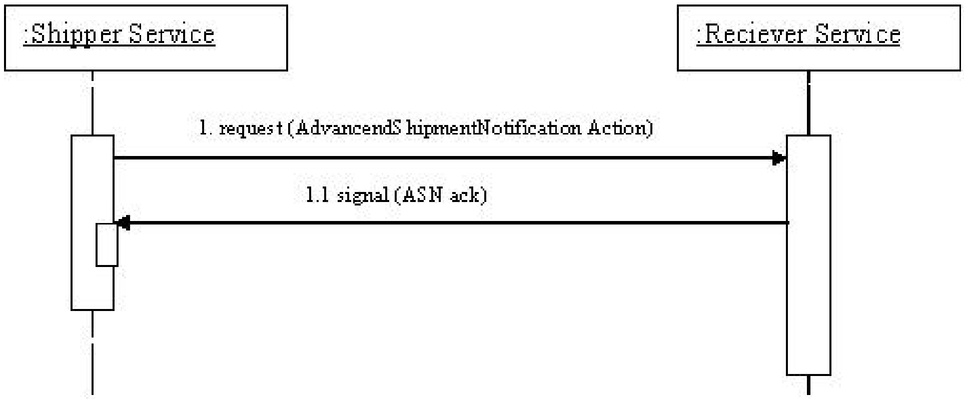

Figure 17 : Pip 3B2 - ASN in RosettaNet

In this case, the Service has to model the Receiver part of it; therefore the tasks of this process would be the reception of the request message, the treatment of the information and the generation of the ACK signal.

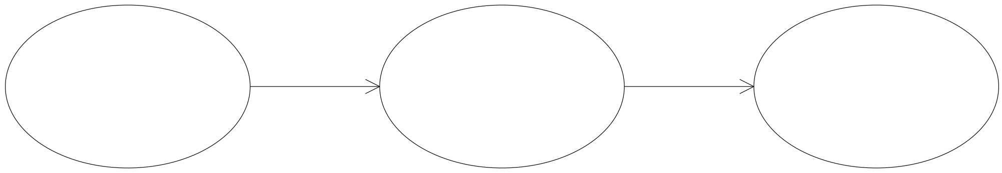

Figure 18 : ASN Receiver Composite process decomposition.

If we analyse the structure of the constructs we see that this composite process is a sequence of atomic ones. The OWL-S description for this process then it would be:

```
<process:CompositeProcess rdf:ID="ASN_1_RECEIVED">
<process:composedOf>
    <process:Sequence>
        <process:components rdf:parseType="Collection">
            <process:AtomicProcess rdf:about="#Process_Input_Message"/>
            <process:AtomicProcess rdf:about="#Treatment_of_Information"/>
            <process:AtomicProcess rdf:about="#Generate_Output_Message"/>
        </process:components>
    </process:Sequence>
</process:composedOf>
</process:CompositeProcess>
```

The Atomic Process "Process_Input_Message" will generate the ontological "lift" as defined in the Conceptual architecture in the Case study requirements is:

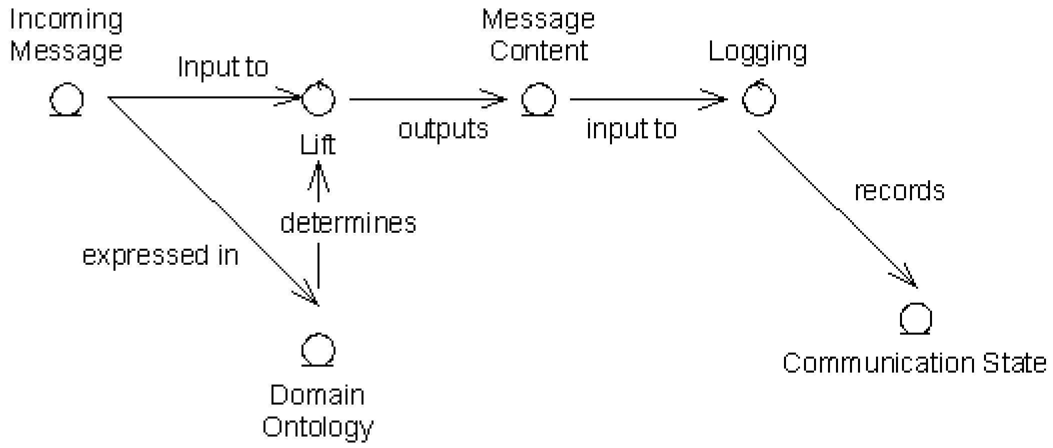

Figure 19 : Ontological Lift

This lift will allow the Logistic Service create the Message Content and trigger the next process – Composite Process "Treatment of the Information".

The Composite Process "Treatment of Information" will access the message content and a reasoner or other method will decide if the answer for the message is an acknowledgment (ACK) or a refusal (NACK). The decision will be transmitted to the next process "Generate Output Message"

The "Generate Output Message" will "lower" the information derived from the message content into a format understandable by the receiver of the message. As defined in the Conceptual architecture, this lower is:

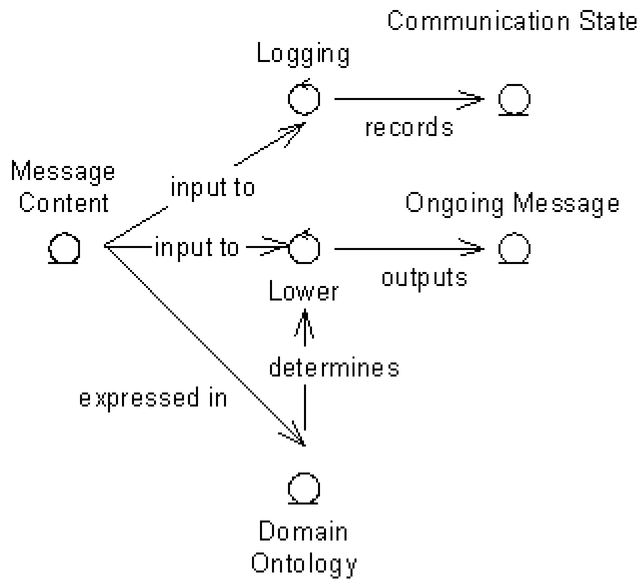

Figure 20 : Ontological Lower into message

3.5.3.2 Process SSM_RECEIVED

The Shipment Status Message Process PIP3B4 is defined by RosettaNet as:

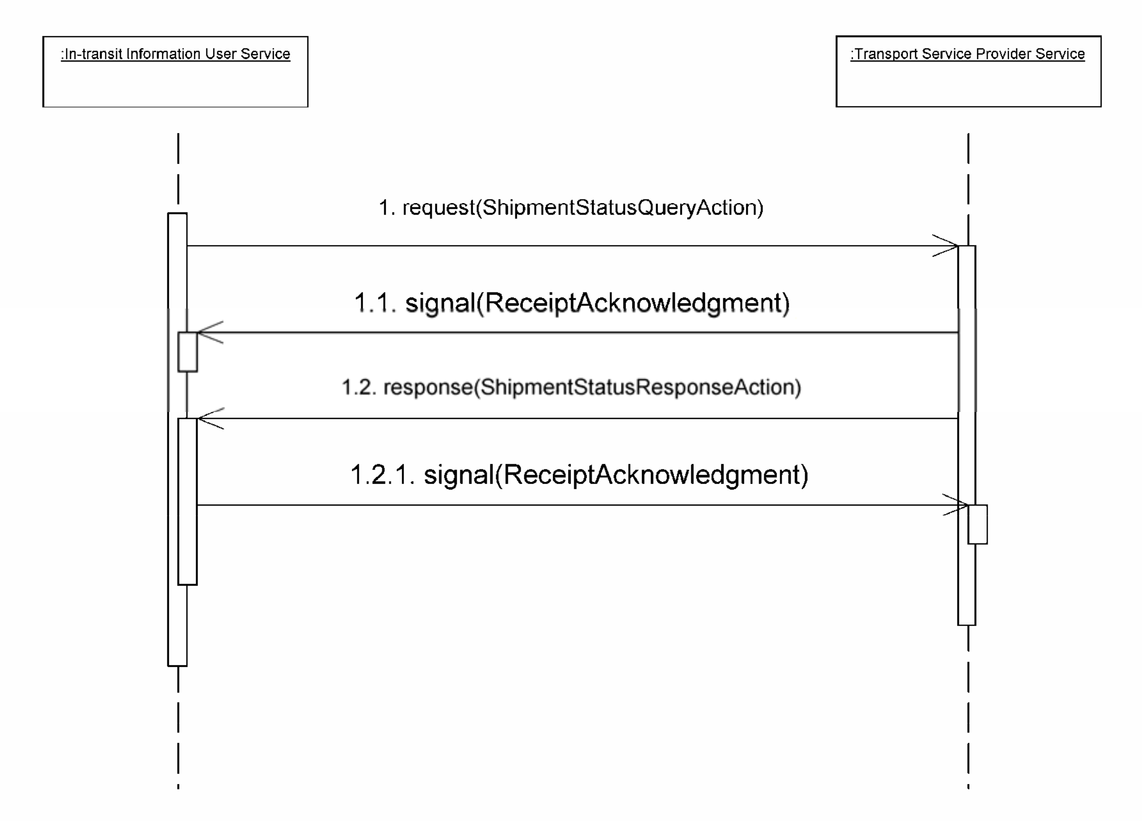

Figure 21 : PIP 3B4 - RosettaNet SSM

The Service will fulfil the communication expected from the Transport Service Provider; the Abstract State Machine that defines the behaviour of this process is as follows:

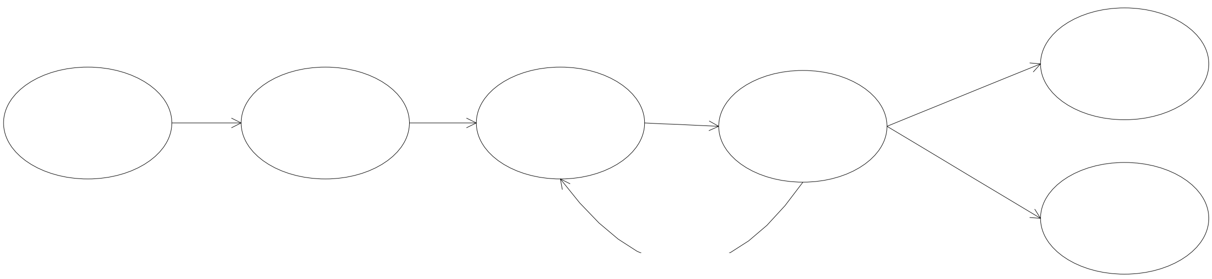

Figure 22 : SSM Composite process decomposition.

If we analyse the structure of the constructs we see that this composite process is a sequence of atomic ones plus an iteration that may end with a choice. The OWL-S description for this process then it would be:

```
<process:CompositeProcess rdf:ID="SSM_RECEIVED">
    <process:composedOf>
        <process:Sequence>
            <process:components rdf:parseType="Collection">
                <process:AtomicProcess rdf:about="#Process_Input_Message"/>
                <process:AtomicProcess rdf:about="#Generate_ACK"/>
                <process:Iterate>
                    <process:components rdf:parseType="Collection">
                        <process:AtomicProcess rdf:about="#Generate_Output_Message"/>
                        <process:AtomicProcess rdf:about="#Wait_for_ack"/>
                        <process:Choice>
                            <process:components rdf:parseType="Collection">
                                <process:AtomicProcess rdf:about="#Raise_Error"/>
                                <process:AtomicProcess rdf:about="#Process_Ack"/>
                            </process:components>
                        </process:Choice>
                    </process:components>
                </process:Iterate>
            </process:components>
        </process:Sequence>
    </process:composedOf>
</process:CompositeProcess>
```

The Process_Input_Message (explained in the previous section) lifts the content of the message into the ontology. The existence of a generation of the ack as a standalone process is due to an incompatibility that may arise in the grounding into WSDL, please refer to the grounding section for this issue. The process Wait-for-ACK-Signal is a blocking process that holds a time-out and also has access where the number of retries is stored. The process Raise-Error and Process-ACK-Signal are mere simple processes that inform to higher levels the new state of the communications.

3.5.3.3 Process ASN_2_SENT

The Advance Shipment Notice Process PIP3B2 is defined by RosettaNet as:

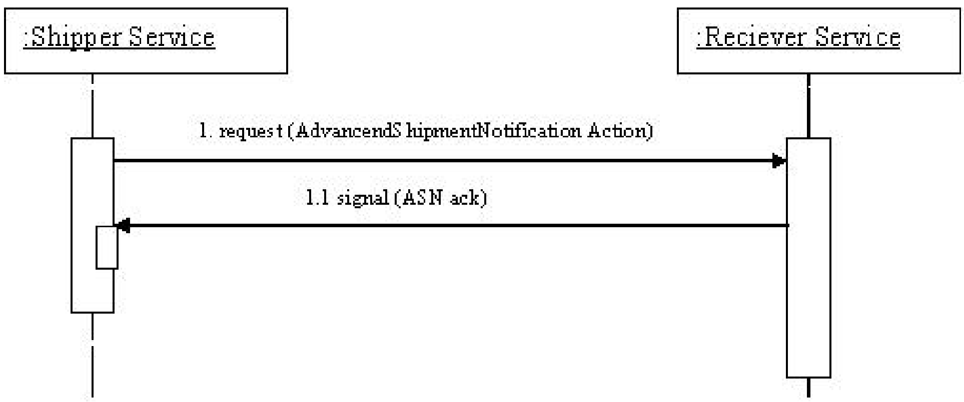

Figure 23 : Pip 3B2 - ASN in RosettaNet

In this case, the Service has to model the Sender part of it, therefore the tasks of this process would be the generation of the ASN message and then wait for the ACK signal.

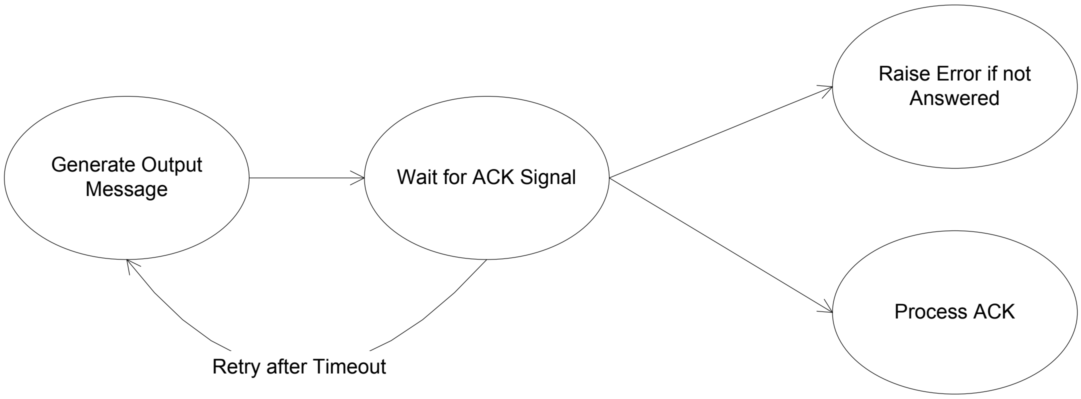

Figure 24 : ASN Sender Composite process decomposition.

If we analyse the structure of the constructs we see that this composite process is a iteration followed by a choice of atomic processes. The OWL-S description for this process then it would be:

```
<process:CompositeProcess rdf:ID="ASN_2_SENT">
    <process:composedOf>
        <process:Iterate>
            <process:components rdf:parseType="Collection">
                <process:AtomicProcess rdf:about="#Generate_Output_Message"/>
                <process:AtomicProcess rdf:about="#Wait_for_ack"/>
                <process:Choice>
                    <process:components rdf:parseType="Collection">
                        <process:AtomicProcess rdf:about="#Raise_Error"/>
                        <process:AtomicProcess rdf:about="#Process_Ack"/>
                    </process:components>
                </process:Choice>
            </process:components>
        </process:Iterate>
    </process:composedOf>
</process:CompositeProcess>
```

Please note that this OWL-S sequencing is identical as a sub-sequence of the SSM message. The process Generate-Output-Message has been also defined previously. The process Wait-for-ACK-Signal is a blocking process that holds a time-out and also has access where the number of retries is stored. The process Raise-Error and Process-ACK-Signal are mere simple processes that inform to higher levels the new state of the communications.

3.5.3.4 Process POD_SENT

The Proof of Delivery, implemented in RosettaNet by PIP 3B13 has the following structure:

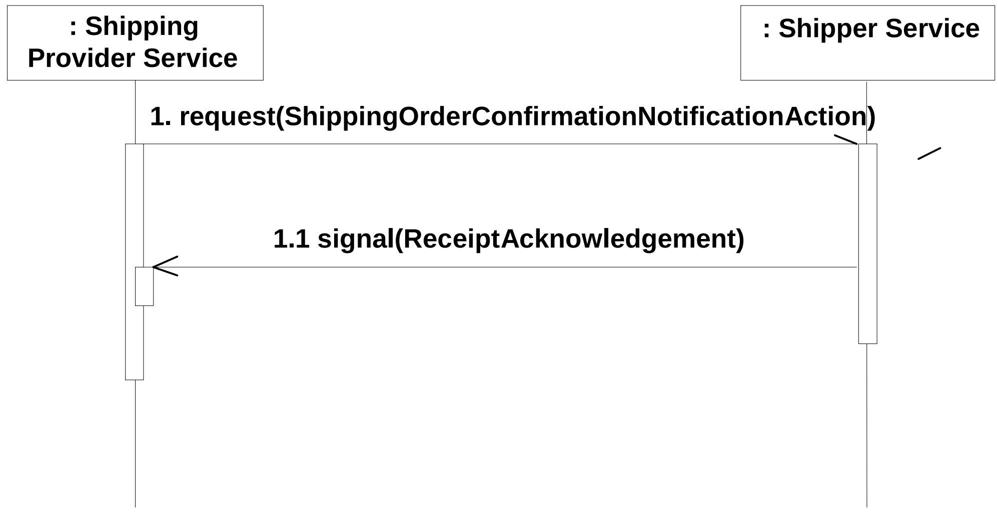

Figure 25 : PIP 3B13 - POD in RosettaNet

In this case, the Service provider fulfils the role of the Shipping Provider as a Freight Forwarder it will send the message once the item has been delivered. This is the Abstract State Machine that models the behaviour of this Service:

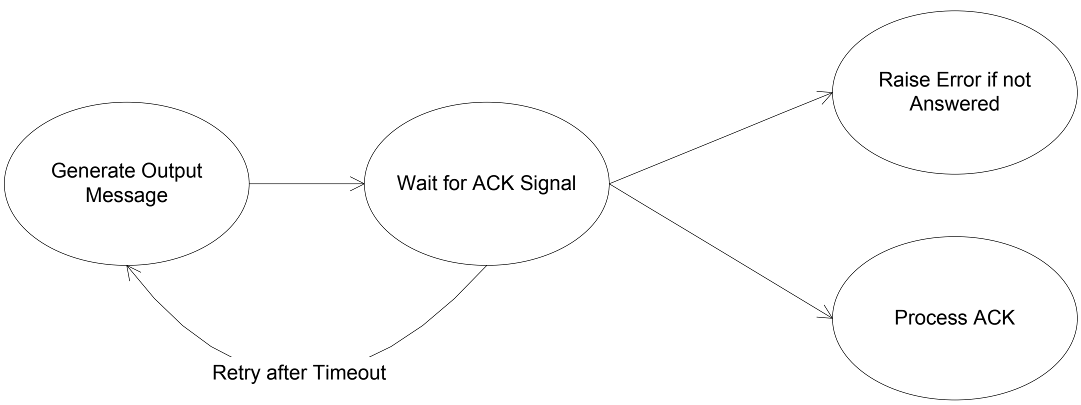

Figure 26 : POD Sender Composite process decomposition.

The decomposition of the POD Sender is identical as the ASN Sender, please refer to that section to explore deeper details.

3.5.3.5 Process FI_SENT

The Freight Invoice, implemented in RosettaNet by PIP 3C3 has the following structure:

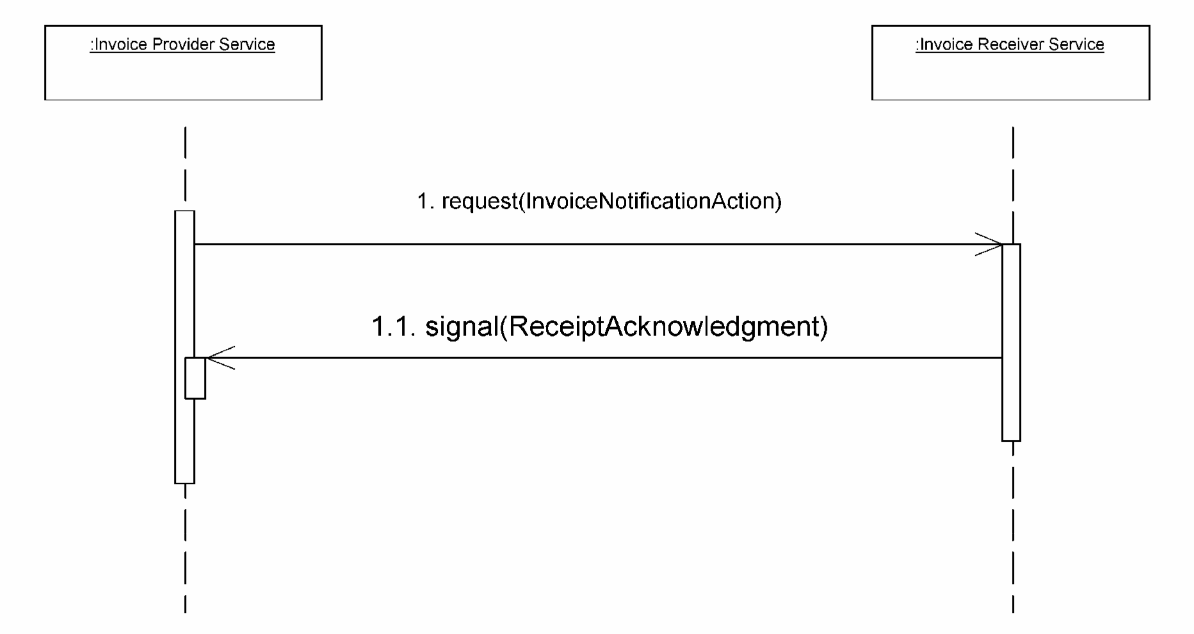

Figure 27 : PIP 3C3 – FI in in RosettaNet

Also this case, the Service provider it will send the message, not necessarily once the item has been delivered. This is the Abstract State Machine that models the behaviour of this Service:

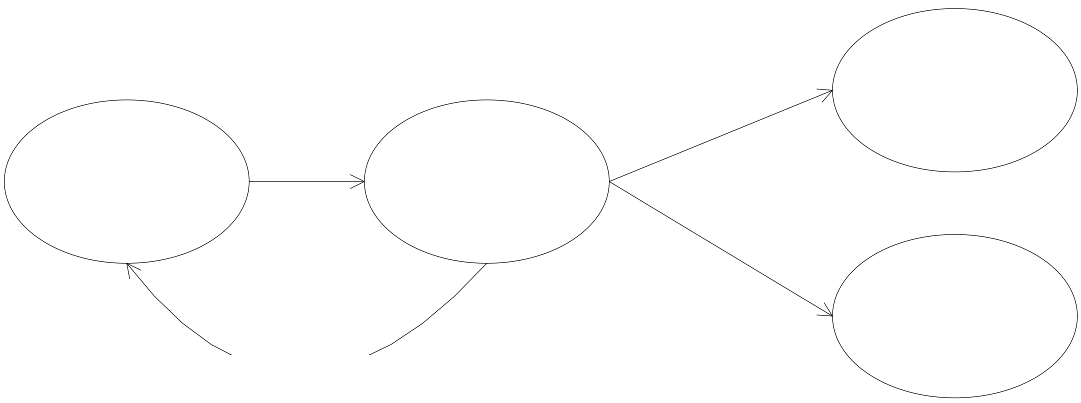

Figure 28 : FI Sender Composite process decomposition.

The decomposition of the FI Sender is identical as the previous POD and ASN Sender, please refer to the ASN section to explore deeper details.

3.5.4 Service Grounding

The Web services based on RosettaNet protocols could be built completely using the WSDL descriptions that OWL-S would provide for each of the processes. Nonetheless we also could find in the industry the following commercial products that will generate a solution or adapt an existing solution to offer RosettaNet compatibility:

- Microsoft BizTalk Server for RosettaNet
- Oracle 9i RosettaNet Driver
- SAP XML Connector + SAP RosettaNet guide
- BEA Systems WebLogic.
- Webmethods integration platform.
- …

On the same line, we could find from the same vendors products or connectors that would offer the connectivity for EDIFact or X.12. Some of those products could offer compatibility to WSDL, but some don't.

From a research point of view, it is interesting to explore how the use of OWL-S grounding would link the process model into WSDL descriptions. The designer could use those WSDL descriptions to create a RosettaNet compliant web services.

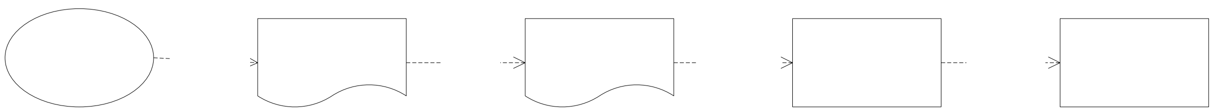

Figure 29 : Semantic Web creation of Web Services

Nonetheless, from a real point of view, in terms of technology adoption by a company; the process of construction of a solution will be from the bottom to the top. Existing web services will be annotated with semantic descriptions and would be registered in order to make it easier to be discovered.

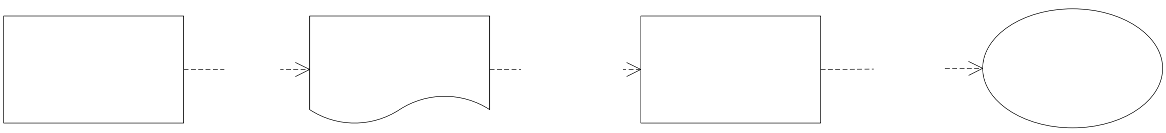

Figure 30 : Creation of Web Services to be in the Semantic Web

This second process leaves no place to WSDL descriptions as they become unnecessary. This is why we think the adoption of OWL-S would generate some redundant work to be done in the grounding. A good solution in this scenario would be the possibility of creation of OWL-S profiles and process models from existing integration protocols like EDIFact, Eb-XML or RosettaNet itself.

Nonetheless, if we want to go ahead with the OWL-S grounding into WSDL, first we should have a look at the summary of Processes and in which high level composite processes are:

- Process-Input-Message: Used in ASN1 and SSM
- Generation-of-the-ACK-Signal: Used in SSM
- Treatment-of-the-Information: Used in ASN1 and SSM
- Generate-Output-Message: Used in ASN1, SSM, ASN2, POD and FI
- Wait-for-ACK-Signal: Used in ASN2, POD and FI
- Raise-Error: Used in ASN2, POD and FI
- Process-ACK-Signal: Used in ASN2, POD and FI

As we could see from the list, all the processes are used in several High level composite processes. The designer would wish to have a single grounding for a process that is forming part on different composite processes.

This is the reason why Process-Input-Message & Generation-of-the-ACK-Signal have been modelled in two processes instead of a single process. If we would use a single process, the tie to the grounding into WSDL 1.1 would generate a predictable incompatibility:

In ASN1 there is a single request-response protocol, while in SSM there are two request response exchanges. If the ACK was generated in the first part of the ASM, to comply with the SSM protocol, that would not allow the use of the same instance of this process in the ASN. In the ASN, the ACK is the confirmation of carrying out the task, (there is a need of a form of reasoning behind). In the case of the SSM, the ACK is just a confirmation of the arrival of the message. This has to be done because the lack of possibility to override the operation.

3.6 Analysis of effectiveness of OWL-S in this concrete scenario

This section pretends to be a compilation of facts that have appeared on the previous sections by the use of OWL-S. The major issues that the use of OWL-S in this scenario are:

1. The lack of expressivity of the profile which offers no business information for classification of the service in terms:
    a. Date & Time Information
    b. Geographic Information, but not only about the geographic radius of the service, it is needed a effective way of presenting the different Sources and destinations of the service.
    c. …

2. The Pre-conditions at profile are based on the Inputs and the Outputs of the Service in a high-level. One of the existing problems with these preconditions at this level is the fact that a concrete input could fulfil different roles in different parts of the process therefore having to exist in different preconditions which could be exclusive between them.

3. One of the main construct is a Split, so according to OWL-S 1.0 specification *"No further specification about waiting or synchronization is made at this level".* This may generate a problem, as SSM messages are only valid until the POD is acknowledged, once the POD is acknowledged the Process control flow must not allow any further SSM to be treated. In this case, OWL-S is not enough to restrict the control flow in such way.

4. Existing web services will be annotated with semantic descriptions and would be registered in order to make it easier to be discovered. WSDL descriptions as they become unnecessary. This is why we think the adoption of OWL-S would generate some redundant work to be done in the grounding. A good solution in this scenario would be the possibility of creation of OWL-S profiles and process models from existing integration protocols like EDIFact, Eb-XML or RosettaNet itself.

5. The existence of an unnecessary process "generation of the ack" as a standalone process due to an incompatibility that happens when we ground the process into WSDL because the lack of possibility to override the operation.

4 Ontologies

As stated in section 3 of deliverable 8.1 "*Following the definition by T. R. Gruber*[4]*, Ontology is an explicit specification of a conceptualization. The term is borrowed from philosophy, where ontology is the systematic study of Existence. For knowledge-based systems, what "exists" is exactly that which can be represented. When the knowledge of a domain is represented in a declarative formalism, the set of objects that can be represented is called the universe of discourse. This set of objects, and the describable relationships among them, are reflected in the representational vocabulary with which a knowledge-based program represents knowledge. Thus, we can describe the ontology of a program by defining a set of representational terms. In such an ontology, definitions associate the names of entities in the universe of discourse (e.g., classes, relations, functions, or other objects) with human-readable text describing what the names are meant to denote, and formal axioms that constrain the interpretation and well-formed use of these terms."*

The case study has different areas where there is a need of ontologic support, following the conceptual architecture proposed in the Case study requirements :

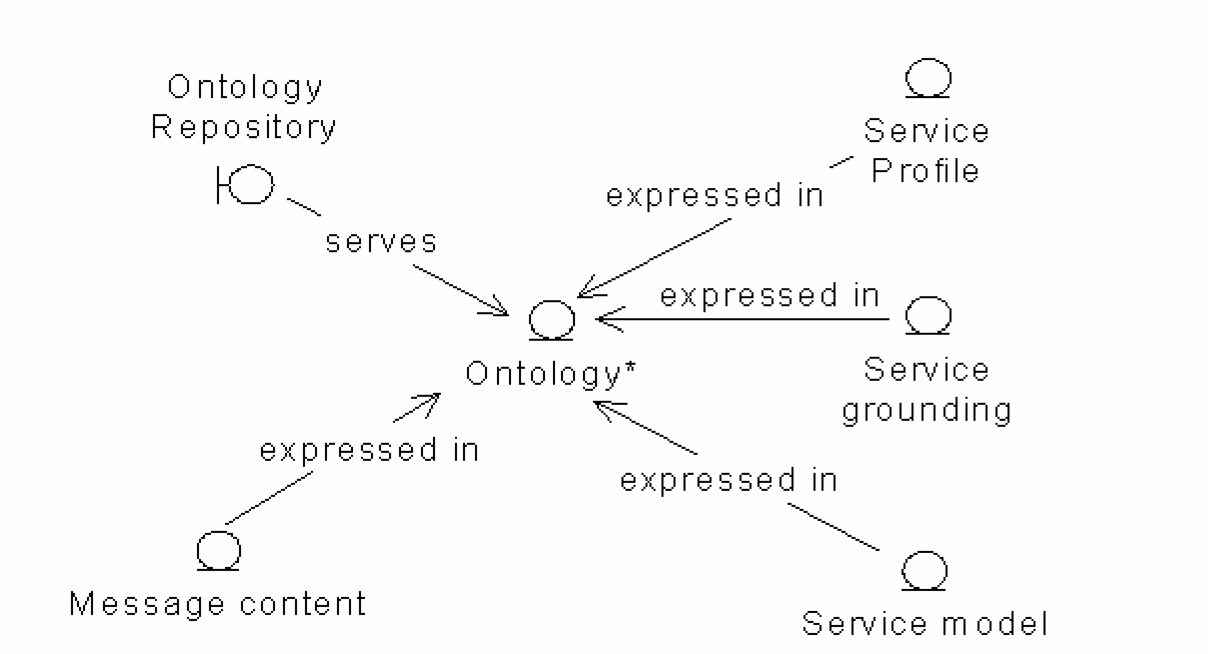

Figure 31 : Conceptual Architecture - Ontologies

The following sections will cover each of the concrete arrows giving the insight of those relationships and the models proposed to support them.

4.1 Message content expressed in Ontology

This section will cover the ontology analysis and proposal regarding the treatment of messages. For the concrete requirements that the case study arise on this concrete area please refer to section 4.2.2 of the document of requirements D8.1.

An incoming message that reaches the system will be encoded using RosettaNet, ANSI X.12 or EDIFact. Those three systems present different ways of encode the same kind of information.

The message content will refer to concepts that live in the range of the world modelled; the world modelled will be called the Domain Ontology.

The Domain Ontology is linked to several conceptual representations:

- World Model: There is all the set that represents the world as it is; there are trucks, there are containers, there are shippers, boxes, Person, Organization… All those concepts exist in all logistic transactions regardless the protocol or the role fulfilled by the message partner (Sender or receiver).
- HP View: There are some concepts that belong to the previous model (Container, boxes…) where HP takes its own interpretation how those concepts relate to each other, therefore add some information to the existing classes.
- HP Model: HP introduces on its information systems, its own concepts to represent information related to the above terms like orders, products). Those concepts are beyond the domain of logistics therefore do not belong to the World model as it and so they are not part of the HP view.

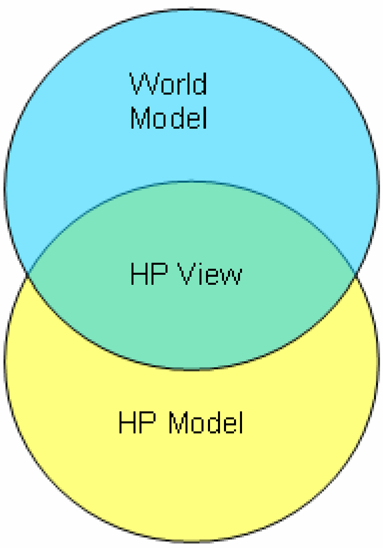

Figure 32 : Ontological overlap

The Domain Ontology as it is developed today is the HP Model.

4.1.1 The World Model Ontology

This ontology has been created by the team members of this case study. It reflects the different inputs received from logistic managers and other people with deep knowledge of the domain. This ontology covers the general concepts in common logistics.

4.1.1.1 Classes (& sub-classes)

Here is an enumerated list of the classes that the World model ontology represents:

- TransportDevice
  - Ship
  - Truck
  - Plane

The class TransportDevice represents the vehicle that actually moves the freight. It is sub-classed by type of transport. The logistic transport is made mainly by sea, road or air. Each kind of transport has its own characteristics and may change the way of communication. It has a relationship "*isCarrying*" with the FreightUnit concept.

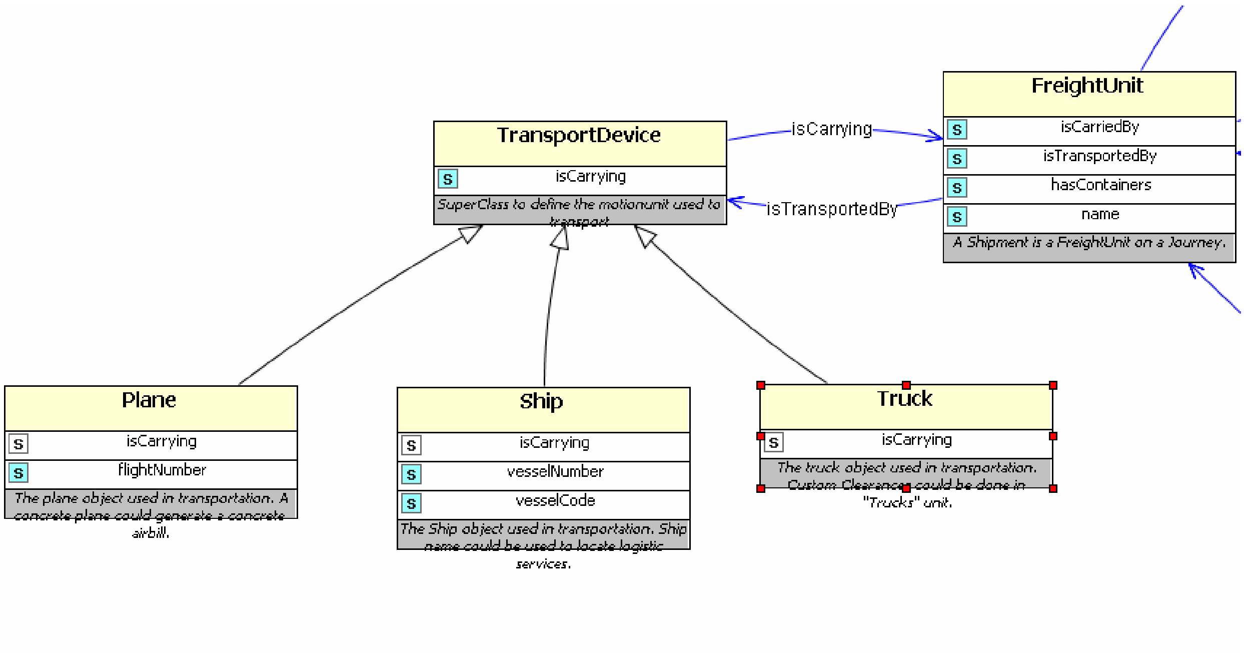

Figure 33 : Transport Device

- FreightUnit

The freight Unit represents the materialization of the fact that a concrete set of containers in a concrete Journeyleg being transported by one TransportDevice under de responsibility of a concrete carrier. The FreightUnit has relationships with each of the concepts mentioned before.

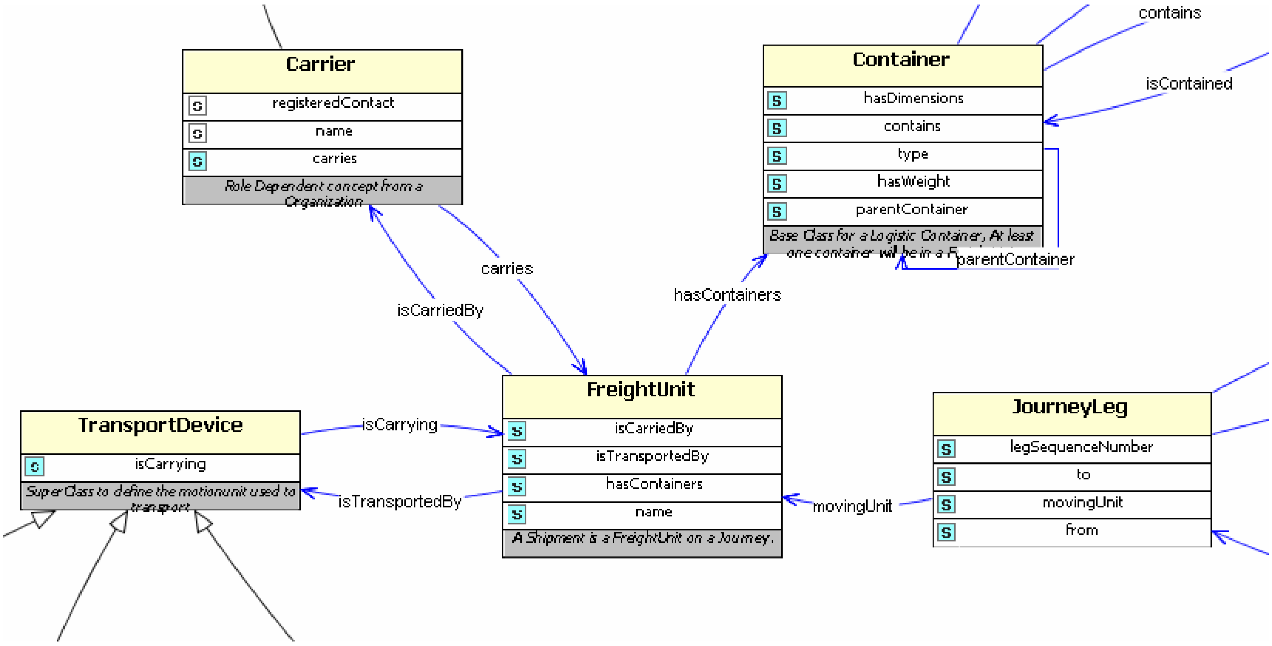

Figure 34 : Freight Unit

- Journeyleg

The journey leg is the class that represents the indivisible journey of a FreightUnit, this is the last level of visibility of the current status of the shipment. User will be able to see in which leg the journey currently is. Usually different legs will be settled with different logistic providers, being this option usually more cost-effective than single journey for a same shipment.

(Picture on the following page)

- Journey

The concept journey is declared to represent the move between the first source and the final destination points. There are two kinds of shipment in the logistic world: The single leg and the multi-leg shipment. A journey has relationship *"leg"* with its different legs. The representation states that any journey will have from one to n different legs.

(Picture on the following page)

- Location
  - ShipFrom
  - ShipTo

The class Location refers to the places where an item could be collected or dispatched. Not all the destination points could be also source points for a logistic provider. There is an initial sub-classing separation between the source point and destination to allow simpler reasoners deal with the ontology. We could also constraint the instances by controlling the domain of the relationships, and then no sub-classing will be needed.

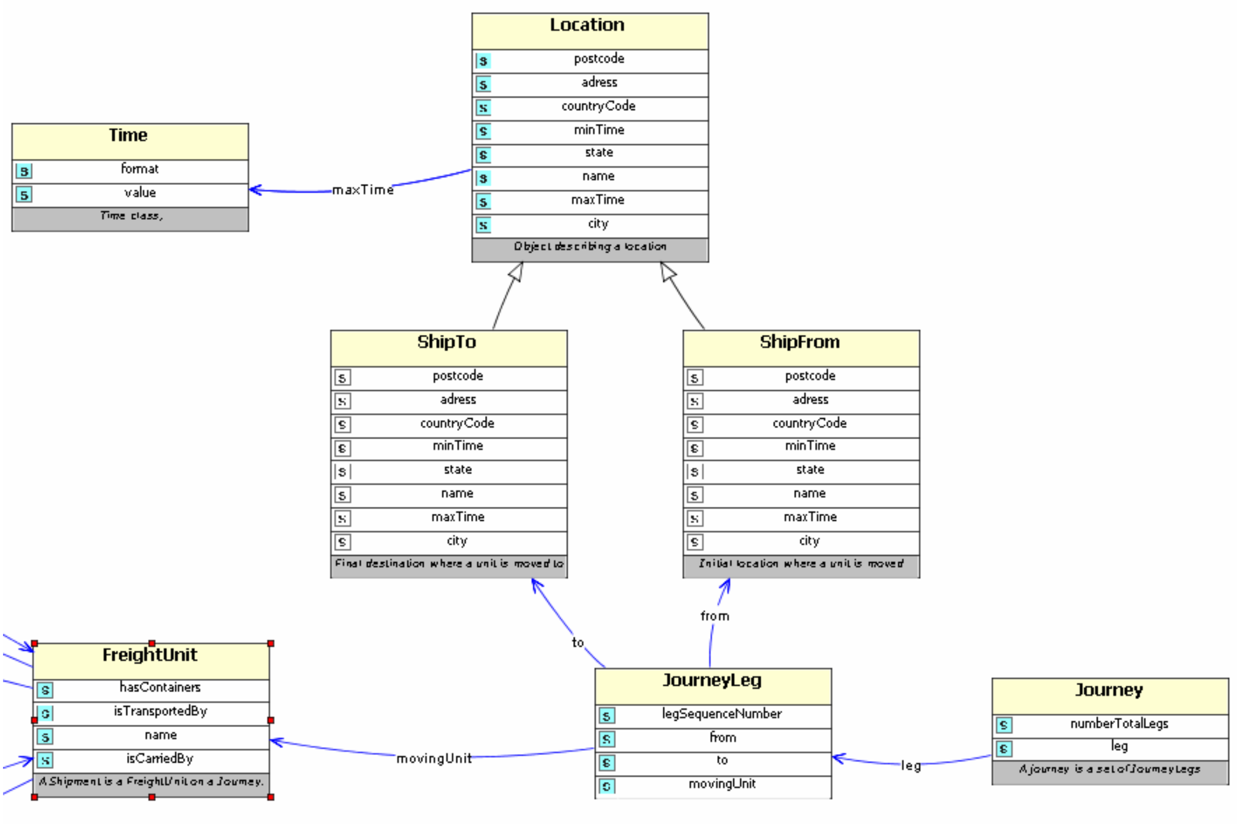

Figure 35 : Journey Legs and Locations

- Container

The container is the concept behind the item shipped by the freight forwarder. The freight forwarder will refer identify the goods into a concrete number of containers. The container will be either virtual or real. A virtual container is when the boxes provided by the source will be identified with a barcode sticker; a real container is when those boxes are introduced into an existing (and identifiable) container. During the logistic transactions a container could be place into another container (i.e. consolidation movements). A container has a concrete dimensions and a concrete weight.

- Box

The box is the concept that will be overridden by view of the parties (in our case by the HP View). We work under the assumption that any item to be shipped has to be self-contained (we will not consider the possibility of shipping liquids or gases). This self-contained item (box) has concrete dimensions and weight.

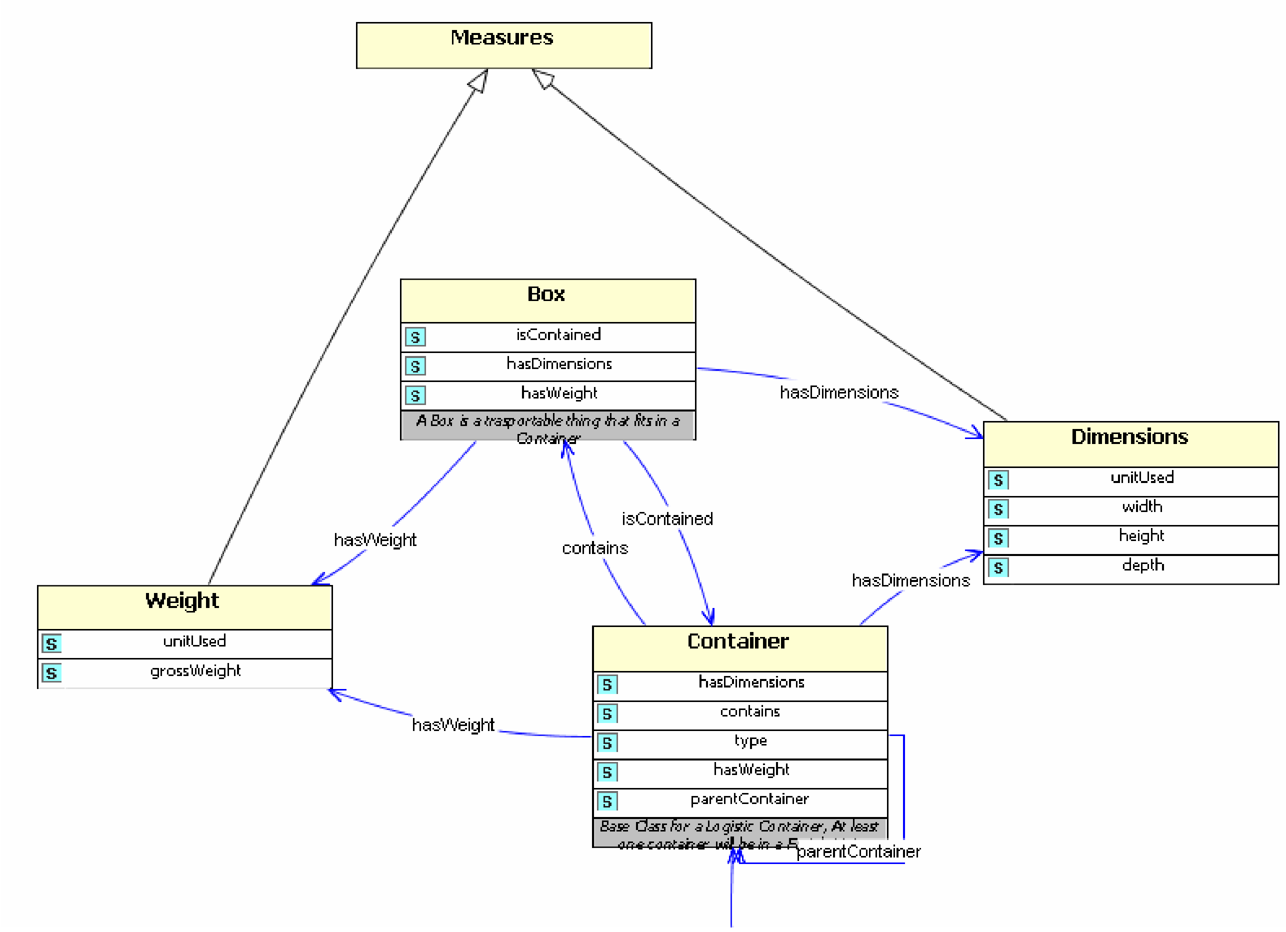

Figure 36 : Containers & Boxes

➢ Measures

We will work at a very basic level about the measures. They will be a natural number (or a triple, in the case of the dimensions) that will be qualified by the unit used.

- Weight: A natural number that represents the weight in the units indicated by the unitUsed property.
- Dimensions: A triple (width, height, depth) of natural numbers representing the dimensions in the units indicated by the unitUsed property.

➢ Person

A person, whether natural or legal, is a responsible entity, it will be at least identified by an unique name.

- Organization: The organization will have always a natural person as a registered contact.
  - Carrier: The carries is a characterization of a concrete organization.
- Individual: When the person is natural.

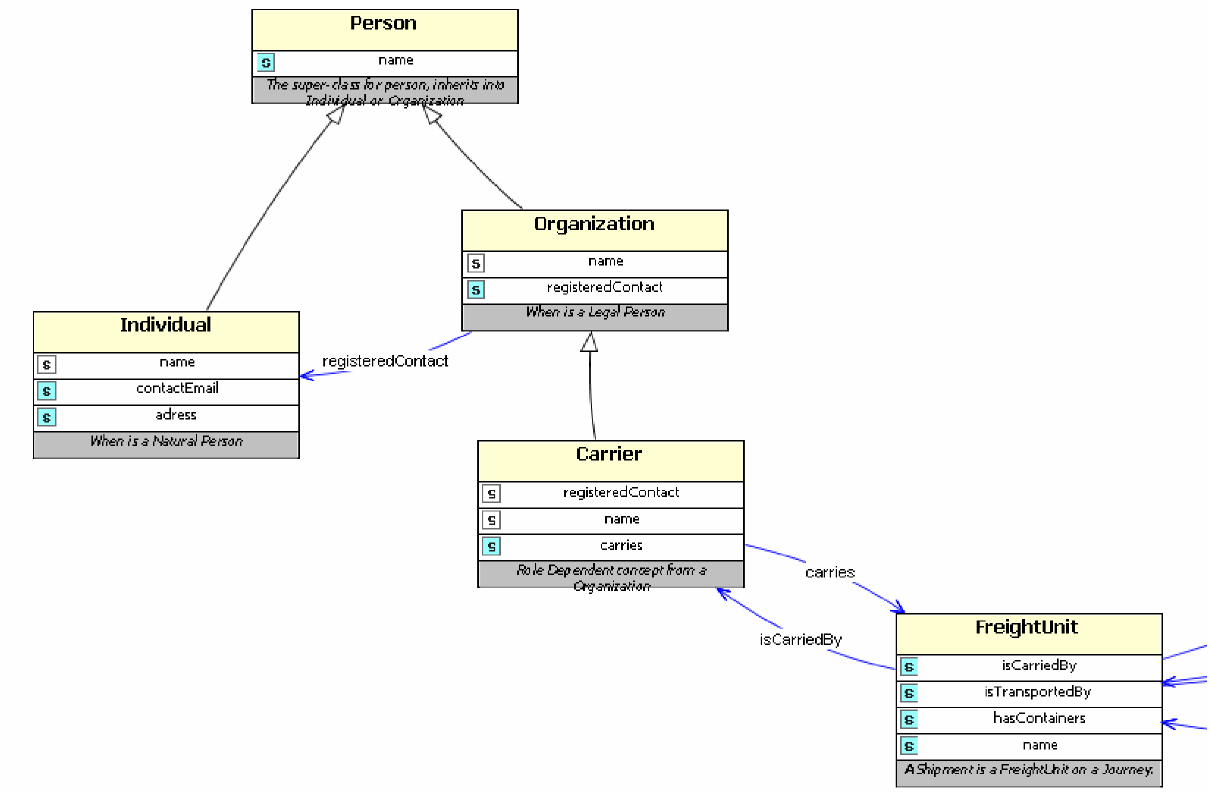

Figure 37 : Person (Individual & Organization)

4.1.2 The HP View Ontology

As the previous ontology, this ontology has been also created from the input of people with deep knowledge of the HP way of doing logistic interactions.

- Product & SubProduct: Those are concepts representing a design, not a physical object. One product design is made up of several design components - the subproducts. These aren't subclasses, they are subcomponents. A (physical object) particular productItem is made up of (physical) SubProductItems. Again, subcomponents.
- Product Item & SubProduct Item: Those are the materialization of the concepts presented above, in this case those are the physical objects that would be self-contained in a Box
- Ordered Item: An ordered item will be either a full product or a sub-product (replacement part)
- LineItem: Each order placed by a costumer will have one or several lines, each of those lines will relate to a concrete product (even if it is a replacement part), therefore the relationship is established with the product and not with subproducts.

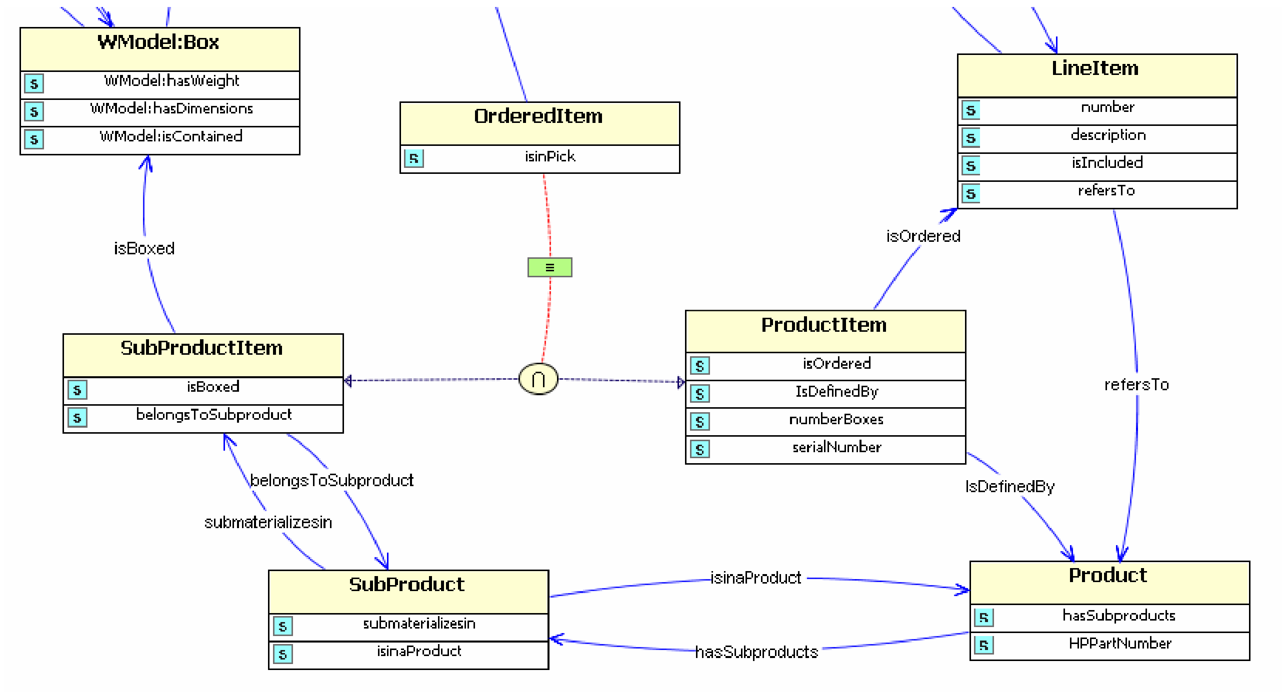

Figure 38 : Products & Subproducts

- Order: The order will have an order ID, this identifier will be unique and persist during the life of the order.
- Supplier: Any order will be related to a concrete supplier, this supplier is a person (whether legal or natural).
- Customer: Any order will have a concrete customer. Customer could be also any kind of person.
- Consignee: The consignee is an actor that may play a role in the case that there is a need for a third party in order to allow mediation. It is still to be explored.

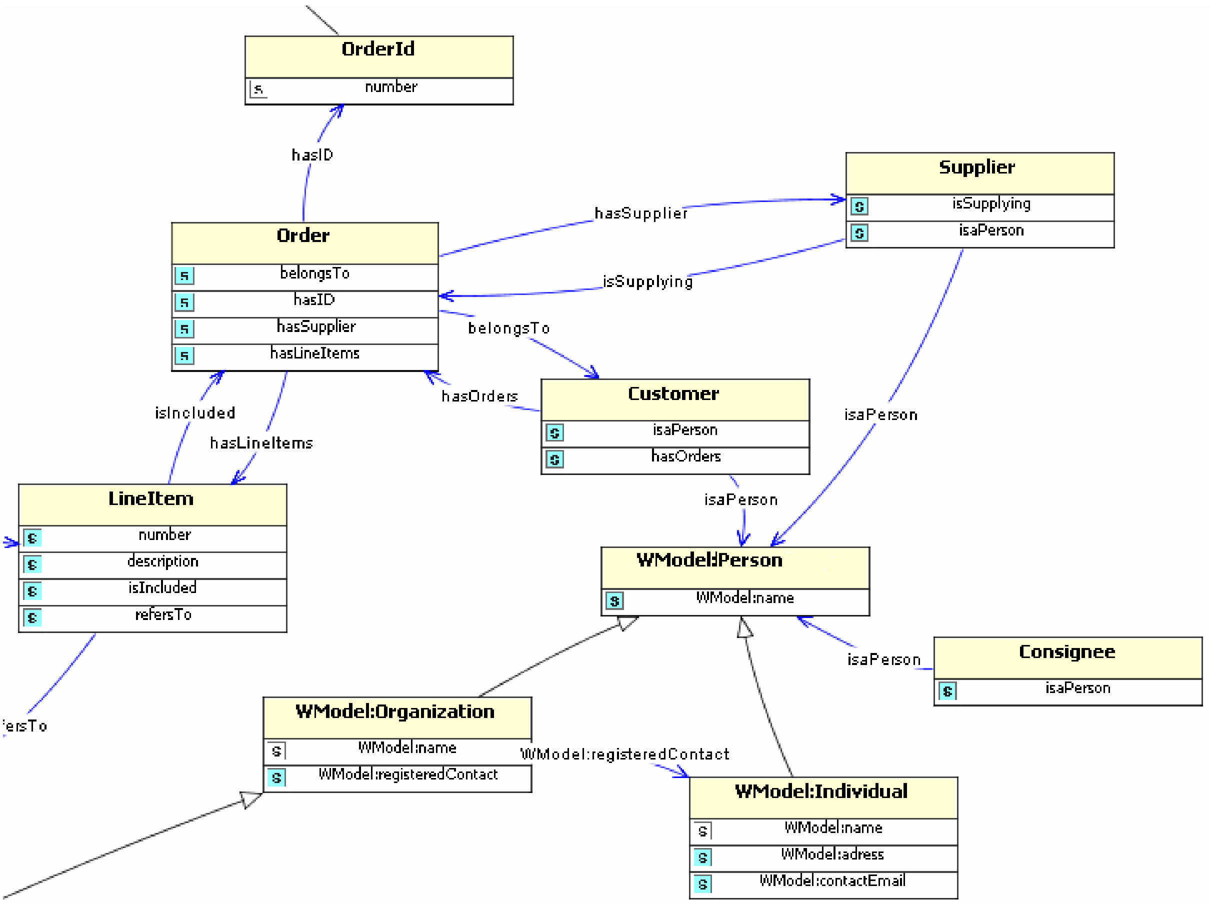

Figure 39 : Orders, Customers & Suppliers

In section 4.2.2 of the requirement document, we conclude as appropriate the use of OWL as the language to specify these ontologies above specified. Please refer to the Appendix sections to see the OWL source code of the ontologies referred in this section.

4.2 Service Profile expressed in Ontology

This section will cover the other ontologies requested by the profile other than the domain ontology (the one used to represent the exchanged messages). In section 3.2 we presented the Service agreement template; one of the parts of this template (column 4) is supporting ontologies. This section will present a detailed view of those ontologies and how they support the Service agreement template. Apart from the domain ontology (presented in the previous chapter), other ontologies that will support our case study are:

- Date & Time Ontology
- Location Ontology
- Payment Ontology

In the following sections we will present the most relevant aspects of those ontologies and how they support the terms of the service template agreement.

4.2.1 Date & Time Ontology

The date and time ontology of the service agreement has to be a shared ontology. To reach a common ontology to be used, either one partner adopts the ontology of the other or both parties use a public ontology. There are some public ontologies that are starting to be developed in among different groups, academic and industrial. We will suggest at this first stage, the use of the Kestrel Time ontology that belongs to the Kestrel Scheduling Ontology[5], an available ontology from the DAML repository (We have use a Owl converter to transform this DAML ontology into a OWL ontology)

```
<Ontology about="">
    <versionInfo>$Id$</versionInfo>
    <comment>Kestrel Scheduling Ontology
    This ontology is based on the time ontology available from Stanford's ontoserver.
    Other groups also defined time ontologies in DAML:
        http://www.ai.sri.com/daml/ontologies/sri-basic/1-0/Time.daml
        http://www.cs.cmu.edu/~softagents/atlas/ontologies/atlas-date.daml
    But they do not provide all the entities needed in a scheduling domain.
</comment>
    <imports resource="http://www.daml.org/2000/10/daml-ont"/>
</Ontology>
```

This ontology has interesting conceptualizations like the time-points, those time points are defined as:

"A time-point is a point in real, historical time (on earth). It is independent of observer and context. A time-point is not a measurement of time, nor is it a specification of time. It is the point in time. The time-points at which events occur can be known with various degrees of precision and approximation, but conceptually time-points are point-like and not interval-like."

The reader could find the complete ontology in the appendixes section.

4.2.2 Location Ontology

The location ontology, as any ontology that supports the Service agreement template should be a shared ontology, and preferably a public ontology. In this case we could use perfectly the “Geographic Feature Name” Ontology[6]. This ontology allows specifying with enough precision the locations where the items have to be shipped.

```
<owl:Ontology rdf:about="">
    <owl:versionInfo>$Id: geonames-ont.daml,v 1.5 2002/09/18 22:34:48 mdean Exp $</owl:versionInfo>
    <rdfs:comment>Geographic Feature Names Ontology, based on NIMA Geonet Names Server</rdfs:comment>
    <rdfs:comment>see http://164.214.2.59/gns/html/index.html</rdfs:comment>
</owl:Ontology>
```

This ontology also has some interesting features like the possibility of specification a secondary country, feature that is necessary to deal when the warehouses are in the customs area.

```
<rdfs:subClassOf>
    <owl:Restriction>
        <owl:onProperty rdf:resource="#secondaryCountry"/>
        <owl:allValuesFrom rdf:resource="&countries-ont;Country"/>
    </owl:Restriction>
  </rdfs:subClassOf>
```

4.2.3 Payment Ontology

The payment ontology is used to agree on the terms and means of payment of the service. In this case has not been possible nowadays to find a public ontology that covers those two areas. The service provider could implement the following ontology and force the service consumer to use it also.

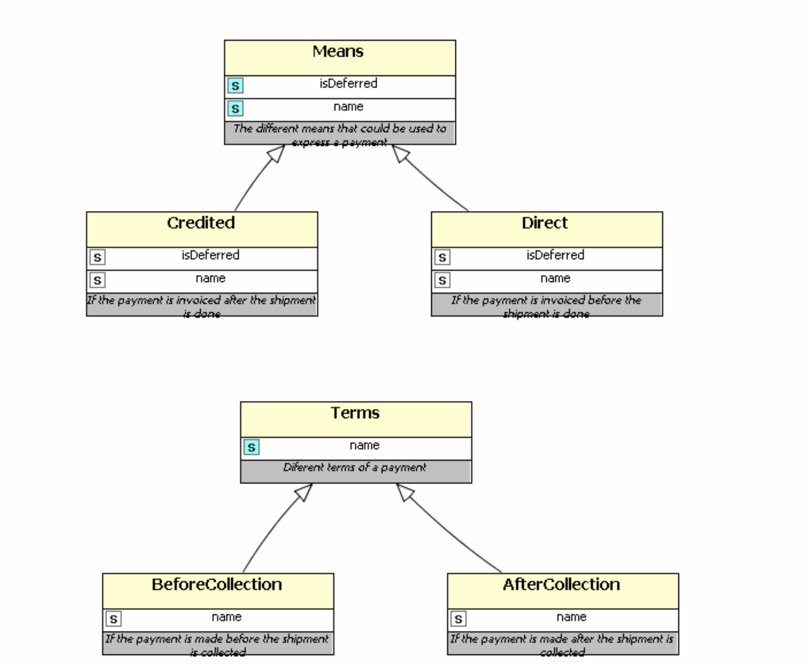

Figure 40 : Payment Ontology

4.3 Service Model expressed in Ontology

The Service model has been expressed using OWL-S. Its formal definition is:

OWL-S is a OWL-based Web service ontology, which supplies Web service providers with a core set of markup language constructs for describing the properties and capabilities of their Web services in unambiguous, computer-intepretable form. OWL-S markup of Web services will facilitate the automation of Web service tasks, including automated Web service discovery, execution, composition and interoperation. Following the layered approach to markup language development, the current version of OWL-S builds on the Ontology Web Language (OWL)

This is the OWL definition of the OWL-S ontology.

```
- <owl:Ontology rdf:about="">
    <owl:versionInfo>$Id: Process.owl,v 1.52 2003/12/08 05:54:07 martin Exp $</owl:versionInfo>
    <rdfs:comment>Upper-level OWL ontology for Processes. Part of the DAML-S/OWL-S effort; see
                 http://www.daml.org/services/.</rdfs:comment>
    <owl:imports rdf:resource="http://www.isi.edu/~pan/damltime/time-entry.owl" />
    <owl:imports rdf:resource="http://www.daml.org/services/owl-s/1.0/Service.owl" />
  </owl:Ontology>
```

The use of owl-s provides a series of advantages but also has some drawbacks, until there is no new Semantic Mark-up language we can not compare them to reach a conclusion about which of those Ontologies is more suitable for express the service model of the services of our scenario.

4.4 Service Grounding expressed in Ontology

The service grounding could be presented in OWL-S as we have presented in the services part of this document. We also have given concrete reasons why we think we should not proceed with WSDL grounding. If we are going to use the grounding into the concrete RosettaNet Invocation, this is the specification of the RosettaNet Implementation Framework.

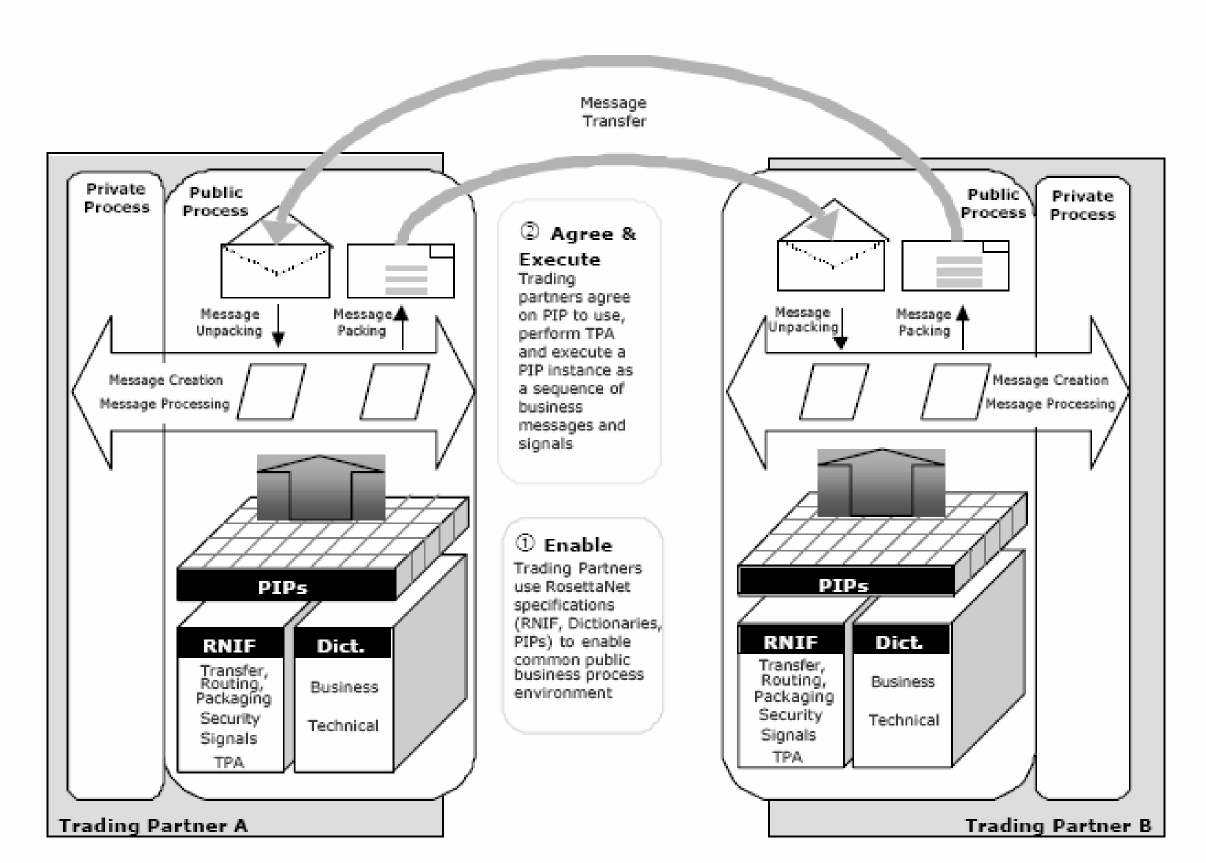

Figure 41 : RosettaNet Specifications in a Trading Partner Implementation

5 Appendixes

5.1 SSM behaviour of the Service Consumer

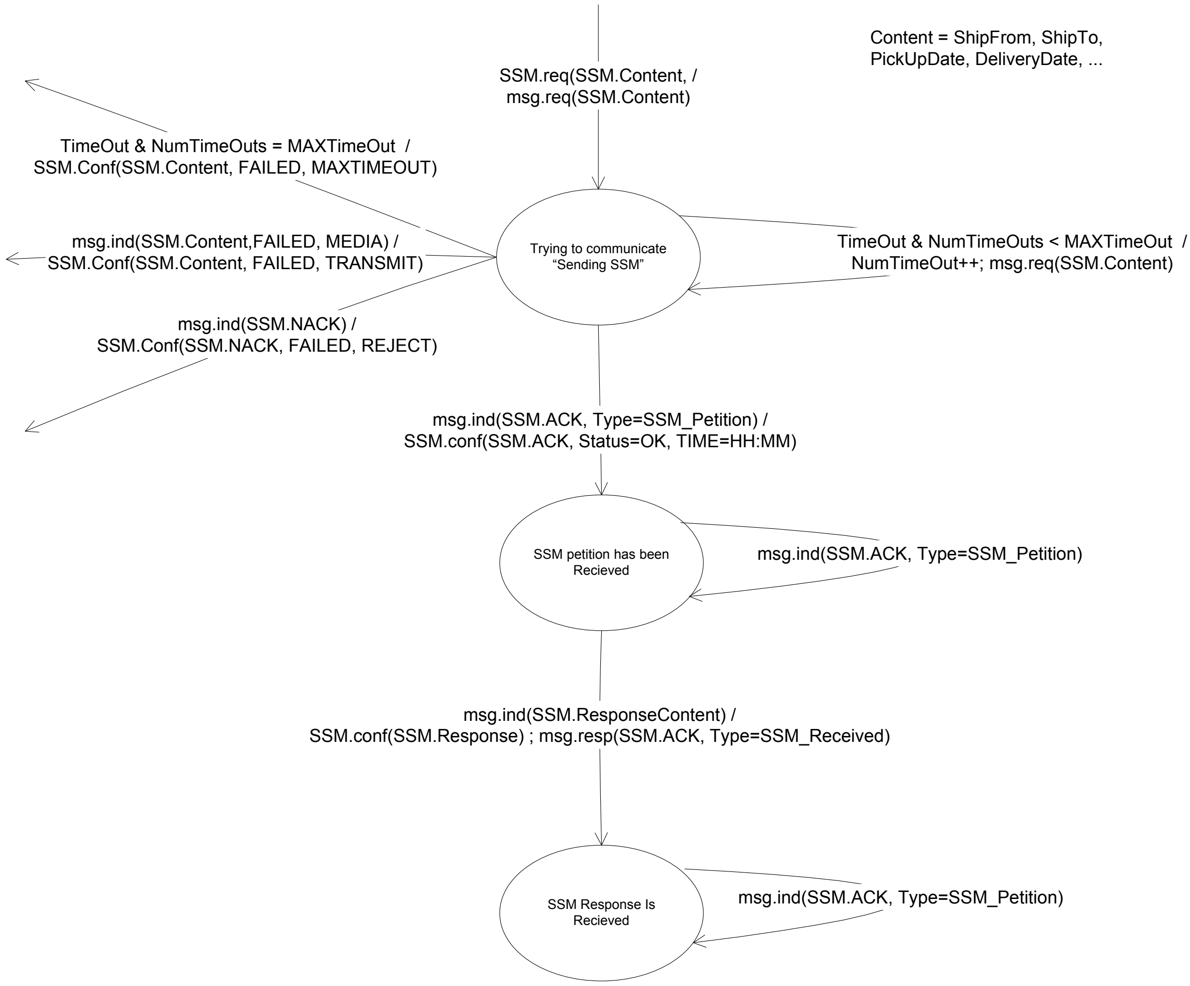

Figure 42 : SSM Behaviour of the Service Consumer

5.2 SSM behaviour of the Service Provider

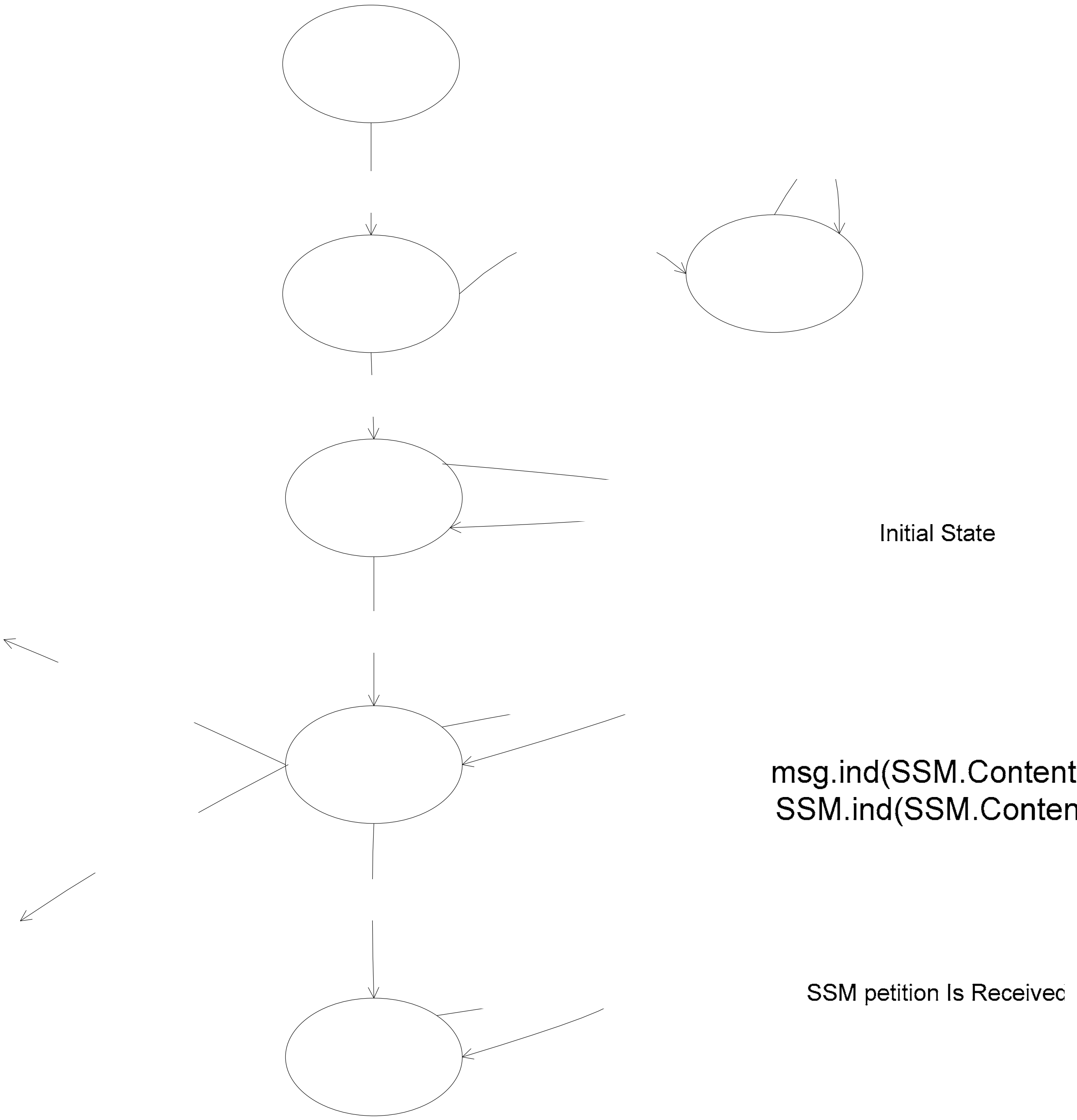

Figure 43 : SSM Behaviour of the Service Provider

5.3 World Model Ontology

```
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#" xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#" xmlns:owl="http://www.w3.org/2002/07/owl#" xmlns="http://ontology.hpl.hp.com/Logistics/WM.owl#">
    <owl:Ontology rdf:about="http://ontology.hpl.hp.com/Logistics/WM.owl">
    </owl:Ontology>
    <owl:Class rdf:ID="Weight">
        <rdfs:subClassOf>
            <owl:Class rdf:ID="Measures"/>
        </rdfs:subClassOf>
    </owl:Class>
    <owl:Class rdf:ID="Plane">
        <rdfs:subClassOf>
            <owl:Class rdf:about="http://ontology.hpl.hp.com/Logistics/WM.owl#TransportDevice"/>
        </rdfs:subClassOf>
        <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string">The plane object used in transportation. A concrete plane could generate a concrete airbill.</rdfs:comment>
    </owl:Class>
    <owl:Class rdf:ID="Individual">
        <rdfs:subClassOf>
            <owl:Class rdf:about="http://ontology.hpl.hp.com/Logistics/WM.owl#Person"/>
        </rdfs:subClassOf>
        <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string">When is a Natural Person</rdfs:comment>
    </owl:Class>
    <owl:Class rdf:ID="TransportDevice">
        <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string">SuperClass to define the motionunit used to transport</rdfs:comment>
    </owl:Class>
    <owl:Class rdf:ID="Carrier">
        <rdfs:subClassOf>
            <owl:Class rdf:about="http://ontology.hpl.hp.com/Logistics/WM.owl#Organization"/>
        </rdfs:subClassOf>
        <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string">Role Dependent concept from a Organization</rdfs:comment>
    </owl:Class>
    <owl:Class rdf:ID="FreightUnit">
        <rdfs:comment>A Shipment is a FreightUnit on a Journey.</rdfs:comment>
    </owl:Class>
    <owl:Class rdf:ID="ShipFrom">
        <rdfs:subClassOf>
            <owl:Class rdf:about="http://ontology.hpl.hp.com/Logistics/WM.owl#Location"/>
        </rdfs:subClassOf>
        <rdfs:comment>Initial location where a unit is moved</rdfs:comment>
    </owl:Class>
    <owl:Class rdf:ID="Ship">
        <rdfs:subClassOf rdf:resource="http://ontology.hpl.hp.com/Logistics/WM.owl#TransportDevice"/>
        <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string">The Ship object used in transportation. Ship name could be used to locate logistic services.</rdfs:comment>
    </owl:Class>
    <owl:Class rdf:ID="Box">
        <rdfs:comment>A Box is a trasportable thing that fits in a Container</rdfs:comment>
    </owl:Class>
    <owl:Class rdf:ID="Location">
        <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string">Object describing a location</rdfs:comment>
    </owl:Class>
    <owl:Class rdf:ID="Dimensions">
```

```xml
        <rdfs:subClassOf rdf:resource="http://ontology.hpl.hp.com/Logistics/WM.owl#Measures"/>
    </owl:Class>
    <owl:Class rdf:ID="Journey">
        <rdfs:comment>A journey is a set of JourneyLegs</rdfs:comment>
    </owl:Class>
    <owl:Class rdf:ID="Organization">
        <rdfs:subClassOf>
            <owl:Class rdf:about="http://ontology.hpl.hp.com/Logistics/WM.owl#Person"/>
        </rdfs:subClassOf>
        <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string">When is a Legal
Person</rdfs:comment>
    </owl:Class>
    <owl:Class rdf:ID="Time">
        <rdfs:comment>Time class,</rdfs:comment>
    </owl:Class>
    <owl:Class rdf:ID="ShipTo">
        <rdfs:subClassOf rdf:resource="http://ontology.hpl.hp.com/Logistics/WM.owl#Location"/>
        <rdfs:comment>Final destination where a unit is moved to</rdfs:comment>
    </owl:Class>
    <owl:Class rdf:ID="JourneyLeg"/>
    <owl:Class rdf:ID="Truck">
        <rdfs:subClassOf rdf:resource="http://ontology.hpl.hp.com/Logistics/WM.owl#TransportDevice"/>
        <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string">The truck object used in
transportation. Custom Clearances could be done in "Trucks" unit.</rdfs:comment>
    </owl:Class>
    <owl:Class rdf:ID="Container">
        <rdfs:comment>Base Class for a Logistic Container, At least one container will be in a
FreightUnit</rdfs:comment>
    </owl:Class>
    <owl:Class rdf:ID="Person">
        <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string">The super-class for person, inherits
into Individual or Organization</rdfs:comment>
    </owl:Class>
    <owl:ObjectProperty rdf:ID="hasContainers">
        <rdfs:domain rdf:resource="http://ontology.hpl.hp.com/Logistics/WM.owl#FreightUnit"/>
        <rdfs:comment>a FreightUnit "hasContainers" Containers</rdfs:comment>
        <owl:inverseOf>
            <owl:ObjectProperty rdf:about="http://ontology.hpl.hp.com/Logistics/WM.owl#isContained"/>
        </owl:inverseOf>
        <rdfs:range rdf:resource="http://ontology.hpl.hp.com/Logistics/WM.owl#Container"/>
    </owl:ObjectProperty>
    <owl:ObjectProperty rdf:ID="isCarrying">
        <owl:inverseOf>
            <owl:ObjectProperty rdf:about="http://ontology.hpl.hp.com/Logistics/WM.owl#isCarriedBy"/>
        </owl:inverseOf>
        <rdfs:comment>A TransportDevice isCarrying multiple FreightUnit</rdfs:comment>
        <rdfs:domain rdf:resource="http://ontology.hpl.hp.com/Logistics/WM.owl#TransportDevice"/>
        <rdfs:range rdf:resource="http://ontology.hpl.hp.com/Logistics/WM.owl#FreightUnit"/>
    </owl:ObjectProperty>
    <owl:ObjectProperty rdf:ID="countryCode">
        <rdfs:comment>A Contry code ... have a look at daml ontology for country codes...</rdfs:comment>
        <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#FunctionalProperty"/>
        <rdfs:domain rdf:resource="http://ontology.hpl.hp.com/Logistics/WM.owl#Location"/>
    </owl:ObjectProperty>
    <owl:ObjectProperty rdf:ID="carries">
        <rdfs:domain rdf:resource="http://ontology.hpl.hp.com/Logistics/WM.owl#Carrier"/>
        <rdfs:range rdf:resource="http://ontology.hpl.hp.com/Logistics/WM.owl#FreightUnit"/>
        <rdfs:comment>A Carrier "carries" a FreightUnit</rdfs:comment>
    </owl:ObjectProperty>
    <owl:ObjectProperty rdf:ID="hasWeight">
        <rdfs:comment>a Concrete box has weight</rdfs:comment>
        <rdfs:range rdf:resource="http://ontology.hpl.hp.com/Logistics/WM.owl#Weight"/>
        <rdfs:domain>
            <owl:Class>
```

```xml
                <owl:unionOf rdf:parseType="Collection">
                    <owl:Class rdf:about="http://ontology.hpl.hp.com/Logistics/WM.owl#Box"/>
                    <owl:Class rdf:about="http://ontology.hpl.hp.com/Logistics/WM.owl#Container"/>
                </owl:unionOf>
            </owl:Class>
        </rdfs:domain>
    </owl:ObjectProperty>
    <owl:ObjectProperty rdf:ID="contains">
        <rdfs:domain rdf:resource="http://ontology.hpl.hp.com/Logistics/WM.owl#Container"/>
        <rdfs:range rdf:resource="http://ontology.hpl.hp.com/Logistics/WM.owl#Box"/>
        <rdfs:comment>The Container contains boxes</rdfs:comment>
        <owl:inverseOf>
            <owl:ObjectProperty rdf:about="http://ontology.hpl.hp.com/Logistics/WM.owl#isContained"/>
        </owl:inverseOf>
    </owl:ObjectProperty>
    <owl:ObjectProperty rdf:ID="registeredContact">
        <rdfs:comment>Individual who is the registered contact for an Organization</rdfs:comment>
        <rdfs:domain rdf:resource="http://ontology.hpl.hp.com/Logistics/WM.owl#Organization"/>
        <rdfs:range rdf:resource="http://ontology.hpl.hp.com/Logistics/WM.owl#Individual"/>
    </owl:ObjectProperty>
    <owl:ObjectProperty rdf:ID="leg">
        <rdfs:domain rdf:resource="http://ontology.hpl.hp.com/Logistics/WM.owl#Journey"/>
        <rdfs:range rdf:resource="http://ontology.hpl.hp.com/Logistics/WM.owl#JourneyLeg"/>
    </owl:ObjectProperty>
    <owl:ObjectProperty rdf:ID="to">
        <rdfs:range rdf:resource="http://ontology.hpl.hp.com/Logistics/WM.owl#ShipTo"/>
        <rdfs:comment>Ending point of the leg</rdfs:comment>
        <rdfs:domain rdf:resource="http://ontology.hpl.hp.com/Logistics/WM.owl#JourneyLeg"/>
        <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#FunctionalProperty"/>
    </owl:ObjectProperty>
    <owl:ObjectProperty rdf:ID="isContained">
        <rdfs:domain rdf:resource="http://ontology.hpl.hp.com/Logistics/WM.owl#Box"/>
        <owl:inverseOf rdf:resource="http://ontology.hpl.hp.com/Logistics/WM.owl#contains"/>
        <rdfs:range rdf:resource="http://ontology.hpl.hp.com/Logistics/WM.owl#Container"/>
        <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#FunctionalProperty"/>
        <rdfs:comment>a Box is Contained in a Container</rdfs:comment>
    </owl:ObjectProperty>
    <owl:ObjectProperty rdf:ID="from">
        <rdfs:range rdf:resource="http://ontology.hpl.hp.com/Logistics/WM.owl#ShipFrom"/>
        <rdfs:comment>The Starting point of the leg</rdfs:comment>
        <rdfs:domain rdf:resource="http://ontology.hpl.hp.com/Logistics/WM.owl#JourneyLeg"/>
        <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#FunctionalProperty"/>
    </owl:ObjectProperty>
    <owl:ObjectProperty rdf:ID="isCarriedBy">
        <rdfs:comment>a FreightUnit is Carriedby a Carrier</rdfs:comment>
        <owl:inverseOf rdf:resource="http://ontology.hpl.hp.com/Logistics/WM.owl#isCarrying"/>
        <rdfs:domain rdf:resource="http://ontology.hpl.hp.com/Logistics/WM.owl#FreightUnit"/>
        <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#FunctionalProperty"/>
        <rdfs:range rdf:resource="http://ontology.hpl.hp.com/Logistics/WM.owl#Carrier"/>
    </owl:ObjectProperty>
    <owl:DatatypeProperty rdf:ID="name">
        <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
        <rdfs:comment>String containing the name</rdfs:comment>
        <rdfs:domain>
            <owl:Class>
                <owl:unionOf rdf:parseType="Collection">
                    <owl:Class rdf:about="http://ontology.hpl.hp.com/Logistics/WM.owl#Location"/>
                    <owl:Class rdf:about="http://ontology.hpl.hp.com/Logistics/WM.owl#Person"/>
                    <owl:Class rdf:about="http://ontology.hpl.hp.com/Logistics/WM.owl#FreightUnit"/>
                </owl:unionOf>
            </owl:Class>
        </rdfs:domain>
    </owl:DatatypeProperty>
    <owl:DatatypeProperty rdf:ID="value">
```

```xml
        <rdfs:domain rdf:resource="http://ontology.hpl.hp.com/Logistics/WM.owl#Time"/>
        <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
        <rdfs:comment>String that contains a value</rdfs:comment>
    </owl:DatatypeProperty>
    <owl:DatatypeProperty rdf:ID="contactEmail">
        <rdfs:comment>String containing the email of the individual</rdfs:comment>
        <rdfs:domain rdf:resource="http://ontology.hpl.hp.com/Logistics/WM.owl#Individual"/>
        <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
    </owl:DatatypeProperty>
    <owl:DatatypeProperty rdf:ID="state">
        <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
        <rdfs:domain rdf:resource="http://ontology.hpl.hp.com/Logistics/WM.owl#Location"/>
        <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#FunctionalProperty"/>
        <rdfs:comment>String containing the name of the State</rdfs:comment>
    </owl:DatatypeProperty>
    <owl:DatatypeProperty rdf:ID="format">
        <rdfs:domain rdf:resource="http://ontology.hpl.hp.com/Logistics/WM.owl#Time"/>
        <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
    </owl:DatatypeProperty>
    <owl:DatatypeProperty rdf:ID="numberTotalLegs">
        <rdfs:comment>Number of total legs</rdfs:comment>
        <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#int"/>
        <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#FunctionalProperty"/>
        <rdfs:domain rdf:resource="http://ontology.hpl.hp.com/Logistics/WM.owl#Journey"/>
    </owl:DatatypeProperty>
    <owl:DatatypeProperty rdf:ID="legSequenceNumber">
        <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#FunctionalProperty"/>
        <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#int"/>
        <rdfs:domain rdf:resource="http://ontology.hpl.hp.com/Logistics/WM.owl#JourneyLeg"/>
        <rdfs:comment>Number of this leg in the journey sequence</rdfs:comment>
    </owl:DatatypeProperty>
    <owl:DatatypeProperty rdf:ID="vesselCode">
        <rdfs:comment>Vessel code (ship’s radio call sign) assigned at the time the vessel is built and stays with it
throughout its life regardless of any vessel name changes.   Code registered and assigned by Lloyd’s Registry of
Shipping.</rdfs:comment>
        <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
        <rdfs:domain rdf:resource="http://ontology.hpl.hp.com/Logistics/WM.owl#Ship"/>
        <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#FunctionalProperty"/>
    </owl:DatatypeProperty>
    <owl:DatatypeProperty rdf:ID="adress">
        <rdfs:comment>String containing a physical Adress</rdfs:comment>
        <rdfs:domain>
            <owl:Class>
                <owl:unionOf rdf:parseType="Collection">
                    <owl:Class rdf:about="http://ontology.hpl.hp.com/Logistics/WM.owl#Location"/>
                    <owl:Class rdf:about="http://ontology.hpl.hp.com/Logistics/WM.owl#Individual"/>
                </owl:unionOf>
            </owl:Class>
        </rdfs:domain>
        <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
        <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#FunctionalProperty"/>
    </owl:DatatypeProperty>
    <owl:FunctionalProperty rdf:ID="maxTime">
        <rdfs:comment>maximum time to deliver or collect the item</rdfs:comment>
        <rdfs:range rdf:resource="http://ontology.hpl.hp.com/Logistics/WM.owl#Time"/>
        <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#ObjectProperty"/>
        <rdfs:domain rdf:resource="http://ontology.hpl.hp.com/Logistics/WM.owl#Location"/>
    </owl:FunctionalProperty>
    <owl:FunctionalProperty rdf:ID="type">
        <rdfs:comment>Code identifying the type and size of the steel container - ISO equipment code.</rdfs:comment>
        <rdfs:domain rdf:resource="http://ontology.hpl.hp.com/Logistics/WM.owl#Container"/>
        <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#DatatypeProperty"/>
        <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
    </owl:FunctionalProperty>
```

```xml
<owl:FunctionalProperty rdf:ID="isTransportedBy">
    <rdfs:range rdf:resource="http://ontology.hpl.hp.com/Logistics/WM.owl#TransportDevice"/>
    <rdfs:comment>a FreightUnit has a TransportDevice</rdfs:comment>
    <rdfs:domain rdf:resource="http://ontology.hpl.hp.com/Logistics/WM.owl#FreightUnit"/>
    <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#ObjectProperty"/>
</owl:FunctionalProperty>
<owl:FunctionalProperty rdf:ID="flightNumber">
    <rdfs:domain rdf:resource="http://ontology.hpl.hp.com/Logistics/WM.owl#Plane"/>
    <rdfs:comment>Flight number of the air carrier.</rdfs:comment>
    <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#DatatypeProperty"/>
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
</owl:FunctionalProperty>
<owl:FunctionalProperty rdf:ID="hasDimensions">
    <rdfs:range rdf:resource="http://ontology.hpl.hp.com/Logistics/WM.owl#Dimensions"/>
    <rdfs:comment>A (box, container) has dimensions</rdfs:comment>
    <rdfs:domain>
        <owl:Class>
            <owl:unionOf rdf:parseType="Collection">
                <owl:Class rdf:about="http://ontology.hpl.hp.com/Logistics/WM.owl#Box"/>
                <owl:Class rdf:about="http://ontology.hpl.hp.com/Logistics/WM.owl#Container"/>
            </owl:unionOf>
        </owl:Class>
    </rdfs:domain>
    <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#ObjectProperty"/>
</owl:FunctionalProperty>
<owl:FunctionalProperty rdf:ID="unitUsed">
    <rdfs:domain>
        <owl:Class>
            <owl:unionOf rdf:parseType="Collection">
                <owl:Class rdf:about="http://ontology.hpl.hp.com/Logistics/WM.owl#Weight"/>
                <owl:Class rdf:about="http://ontology.hpl.hp.com/Logistics/WM.owl#Dimensions"/>
            </owl:unionOf>
        </owl:Class>
    </rdfs:domain>
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
    <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#DatatypeProperty"/>
</owl:FunctionalProperty>
<owl:FunctionalProperty rdf:ID="depth">
    <rdfs:domain rdf:resource="http://ontology.hpl.hp.com/Logistics/WM.owl#Dimensions"/>
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#int"/>
    <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#DatatypeProperty"/>
    <rdfs:comment>The Depth of an object</rdfs:comment>
</owl:FunctionalProperty>
<owl:FunctionalProperty rdf:ID="city">
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
    <rdfs:domain rdf:resource="http://ontology.hpl.hp.com/Logistics/WM.owl#Location"/>
    <rdfs:comment>String Containing a name of a City</rdfs:comment>
    <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#DatatypeProperty"/>
</owl:FunctionalProperty>
<owl:FunctionalProperty rdf:ID="minTime">
    <rdfs:comment>Minimum time to either collect or deliver</rdfs:comment>
    <rdfs:domain rdf:resource="http://ontology.hpl.hp.com/Logistics/WM.owl#Location"/>
    <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#ObjectProperty"/>
    <rdfs:range rdf:resource="http://ontology.hpl.hp.com/Logistics/WM.owl#Time"/>
</owl:FunctionalProperty>
<owl:FunctionalProperty rdf:ID="movingUnit">
    <rdfs:range rdf:resource="http://ontology.hpl.hp.com/Logistics/WM.owl#FreightUnit"/>
    <rdfs:domain rdf:resource="http://ontology.hpl.hp.com/Logistics/WM.owl#JourneyLeg"/>
    <rdfs:comment>The FreightUnit that is going to be moved</rdfs:comment>
    <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#ObjectProperty"/>
</owl:FunctionalProperty>
<owl:FunctionalProperty rdf:ID="parentContainer">
    <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#ObjectProperty"/>
    <rdfs:comment>a Container (could) fit in another container</rdfs:comment>
```

```
        <rdfs:range rdf:resource="http://ontology.hpl.hp.com/Logistics/WM.owl#Container"/>
        <rdfs:domain rdf:resource="http://ontology.hpl.hp.com/Logistics/WM.owl#Container"/>
    </owl:FunctionalProperty>
    <owl:FunctionalProperty rdf:ID="grossWeight">
        <rdfs:domain rdf:resource="http://ontology.hpl.hp.com/Logistics/WM.owl#Weight"/>
        <rdfs:comment>Natural number containing the Total Gross Weight</rdfs:comment>
        <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#int"/>
        <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#DatatypeProperty"/>
    </owl:FunctionalProperty>
    <owl:FunctionalProperty rdf:ID="height">
        <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#int"/>
        <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#DatatypeProperty"/>
        <rdfs:comment>Value Containing the height</rdfs:comment>
        <rdfs:domain rdf:resource="http://ontology.hpl.hp.com/Logistics/WM.owl#Dimensions"/>
    </owl:FunctionalProperty>
    <owl:FunctionalProperty rdf:ID="width">
        <rdfs:comment>The Width of an object</rdfs:comment>
        <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#int"/>
        <rdfs:domain rdf:resource="http://ontology.hpl.hp.com/Logistics/WM.owl#Dimensions"/>
        <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#DatatypeProperty"/>
    </owl:FunctionalProperty>
    <owl:FunctionalProperty rdf:ID="postcode">
        <rdfs:domain rdf:resource="http://ontology.hpl.hp.com/Logistics/WM.owl#Location"/>
        <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
        <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#DatatypeProperty"/>
        <rdfs:comment>String containing the PostCode</rdfs:comment>
    </owl:FunctionalProperty>
    <owl:FunctionalProperty rdf:ID="vesselNumber">
        <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#DatatypeProperty"/>
        <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
        <rdfs:comment>Identifier to uniquely identify which vessel the shipment is on (Ocean only). It can also be a
vessel name.</rdfs:comment>
        <rdfs:domain rdf:resource="http://ontology.hpl.hp.com/Logistics/WM.owl#Ship"/>
    </owl:FunctionalProperty>
</rdf:RDF>
```

5.4 HPModel Ontology


```
<rdf:RDF xmlns="http://ontology.hpl.hp.com/Logistics/HPModel.owl#" xmlns:rdf="http://www.w3.org/1999/02/22-rdf-
syntax-ns#" xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#" xmlns:owl="http://www.w3.org/2002/07/owl#"
xmlns:WModel="http://ontology.hpl.hp.com/Logistics/WM.owl#">
    <owl:Ontology rdf:about="http://ontology.hpl.hp.com/Logistics/HPModel.owl">
        <owl:imports rdf:resource="http://ontology.hpl.hp.com/Logistics/WM.owl"/>
    </owl:Ontology>
    <owl:Class rdf:ID="Pack">
        <rdfs:comment>A Pack is composed of Different Picks has an identifier and stands in a
Container.</rdfs:comment>
    </owl:Class>
    <owl:Class rdf:ID="Time"/>
    <owl:Class rdf:ID="Product">
        <rdfs:comment>The "abstract" concept of a Product of HP. e.g. Laserjet 5000</rdfs:comment>
    </owl:Class>
    <owl:Class rdf:ID="Supplier">
        <rdfs:comment>a Supplier is a Person that Supplies Orders</rdfs:comment>
    </owl:Class>
    <owl:Class rdf:ID="ProductItem">
        <rdfs:comment>The actual Instance of the abstract concept Product. It refers to a contrete
Article</rdfs:comment>
    </owl:Class>
    <owl:Class rdf:ID="Order">
        <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string">The concept Order</rdfs:comment>
    </owl:Class>
    <owl:Class rdf:ID="Date"/>
    <owl:Class rdf:ID="Customer">
        <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string">Role Dependent concept from a
Person</rdfs:comment>
    </owl:Class>
    <owl:Class rdf:ID="OrderedItem">
        <owl:equivalentClass>
            <owl:Class>
                <owl:intersectionOf rdf:parseType="Collection">
                    <owl:Class rdf:about="http://ontology.hpl.hp.com/Logistics/HPModel.owl#SubProductItem"/>
                    <owl:Class rdf:about="http://ontology.hpl.hp.com/Logistics/HPModel.owl#ProductItem"/>
                </owl:intersectionOf>
            </owl:Class>
        </owl:equivalentClass>
        <rdfs:comment>OrderedItem is a SubproductItem or a ProductItem</rdfs:comment>
    </owl:Class>
    <owl:Class rdf:ID="Identifier">
        <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string">Any kind ot label that identifies an
object /concept</rdfs:comment>
    </owl:Class>
    <owl:Class rdf:ID="SubProduct">
        <rdfs:comment>A Product could be composed of several subproducts</rdfs:comment>
    </owl:Class>
    <owl:Class rdf:ID="SubProductItem">
        <rdfs:comment>a Product Item will be composed of different Subproduct items, each of them will be in a
concrete box.</rdfs:comment>
    </owl:Class>
    <owl:Class rdf:ID="OrderId">
        <rdfs:subClassOf rdf:resource="http://ontology.hpl.hp.com/Logistics/HPModel.owl#Identifier"/>
        <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string">Identifier for an
Order</rdfs:comment>
    </owl:Class>
    <owl:Class rdf:ID="PickId">
        <rdfs:subClassOf rdf:resource="http://ontology.hpl.hp.com/Logistics/HPModel.owl#Identifier"/>
        <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string">Identifier for a Pick</rdfs:comment>
    </owl:Class>
```

```
<owl:Class rdf:ID="Consignee">
    <rdfs:comment>RoleDependent Class of Person</rdfs:comment>
</owl:Class>
<owl:Class rdf:ID="PackId">
    <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string">Identifier for a Pack</rdfs:comment>
    <rdfs:subClassOf rdf:resource="http://ontology.hpl.hp.com/Logistics/HPModel.owl#Identifier"/>
</owl:Class>
<owl:Class rdf:ID="Pick">
    <rdfs:comment>A pick is a unit of different products collected from a near location</rdfs:comment>
</owl:Class>
<owl:Class rdf:ID="LineItem">
    <rdfs:comment>Lines that compose an Order</rdfs:comment>
</owl:Class>
<owl:ObjectProperty rdf:ID="isIncluded">
    <rdfs:comment>a LineItem l "is included" in an order O</rdfs:comment>
    <rdfs:domain rdf:resource="http://ontology.hpl.hp.com/Logistics/HPModel.owl#LineItem"/>
    <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#FunctionalProperty"/>
    <rdfs:range rdf:resource="http://ontology.hpl.hp.com/Logistics/HPModel.owl#Order"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="submaterializesin">
    <rdfs:domain rdf:resource="http://ontology.hpl.hp.com/Logistics/HPModel.owl#SubProduct"/>
    <rdfs:comment>a SubProduct Materialices in a subproduct item</rdfs:comment>
    <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#FunctionalProperty"/>
    <rdfs:range rdf:resource="http://ontology.hpl.hp.com/Logistics/HPModel.owl#SubProductItem"/>
    <owl:inverseOf>
        <owl:ObjectProperty rdf:about="http://ontology.hpl.hp.com/Logistics/HPModel.owl#belongsToSubproduct"/>
    </owl:inverseOf>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="hasAssociatedOrders">
    <rdfs:range rdf:resource="http://ontology.hpl.hp.com/Logistics/HPModel.owl#Order"/>
    <rdfs:comment>A Container could have a number of associated Orders</rdfs:comment>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="containsPicks">
    <rdfs:range rdf:resource="http://ontology.hpl.hp.com/Logistics/HPModel.owl#Pick"/>
    <rdfs:domain rdf:resource="http://ontology.hpl.hp.com/Logistics/HPModel.owl#Pack"/>
    <rdfs:comment>a Pack contain several Picks</rdfs:comment>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="hasOrders">
    <rdfs:range rdf:resource="http://ontology.hpl.hp.com/Logistics/HPModel.owl#Order"/>
    <rdfs:comment>a Costumer places Orders</rdfs:comment>
    <owl:inverseOf>
        <owl:ObjectProperty rdf:about="http://ontology.hpl.hp.com/Logistics/HPModel.owl#belongsTo"/>
    </owl:inverseOf>
    <rdfs:domain rdf:resource="http://ontology.hpl.hp.com/Logistics/HPModel.owl#Customer"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="belongsTo">
    <owl:inverseOf rdf:resource="http://ontology.hpl.hp.com/Logistics/HPModel.owl#hasOrders"/>
    <rdfs:range rdf:resource="http://ontology.hpl.hp.com/Logistics/HPModel.owl#Customer"/>
    <rdfs:comment>A Concrete Order belongs to a Customer</rdfs:comment>
    <rdfs:domain rdf:resource="http://ontology.hpl.hp.com/Logistics/HPModel.owl#Order"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="IsDefinedBy">
    <owl:inverseOf>
        <owl:ObjectProperty rdf:about="http://ontology.hpl.hp.com/Logistics/HPModel.owl#hasUnits"/>
    </owl:inverseOf>
    <rdfs:comment>a concrete Product Item "IsDefinedBy" a concrete Product</rdfs:comment>
    <rdfs:range rdf:resource="http://ontology.hpl.hp.com/Logistics/HPModel.owl#Product"/>
    <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#FunctionalProperty"/>
    <rdfs:domain rdf:resource="http://ontology.hpl.hp.com/Logistics/HPModel.owl#ProductItem"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="isinContainer">
    <rdfs:comment>a Pack is in Container</rdfs:comment>
    <rdfs:domain rdf:resource="http://ontology.hpl.hp.com/Logistics/HPModel.owl#Pack"/>
    <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#FunctionalProperty"/>
```

```
    <rdfs:range rdf:resource="http://ontology.hpl.hp.com/Logistics/WM.owl#Container"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="containsPacks">
    <rdfs:domain rdf:resource="http://ontology.hpl.hp.com/Logistics/WM.owl#Container"/>
    <rdfs:comment>A Container Contains several Packs</rdfs:comment>
    <rdfs:range rdf:resource="http://ontology.hpl.hp.com/Logistics/HPModel.owl#Pack"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="belongsToSubproduct">
    <rdfs:comment>a SubproductItem Belongs to a subproduct</rdfs:comment>
    <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#FunctionalProperty"/>
    <owl:inverseOf rdf:resource="http://ontology.hpl.hp.com/Logistics/HPModel.owl#submaterializesin"/>
    <rdfs:domain rdf:resource="http://ontology.hpl.hp.com/Logistics/HPModel.owl#SubProductItem"/>
    <rdfs:range rdf:resource="http://ontology.hpl.hp.com/Logistics/HPModel.owl#SubProduct"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="isinPick">
    <rdfs:comment>an OrderedItem is in Pick</rdfs:comment>
    <rdfs:range rdf:resource="http://ontology.hpl.hp.com/Logistics/HPModel.owl#Pick"/>
    <rdfs:domain rdf:resource="http://ontology.hpl.hp.com/Logistics/HPModel.owl#OrderedItem"/>
    <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#FunctionalProperty"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="isSupplying">
    <rdfs:comment>A person P "isSupplying" multiple orders O</rdfs:comment>
    <owl:inverseOf>
        <owl:FunctionalProperty rdf:about="http://ontology.hpl.hp.com/Logistics/HPModel.owl#hasSupplier"/>
    </owl:inverseOf>
    <rdfs:domain rdf:resource="http://ontology.hpl.hp.com/Logistics/HPModel.owl#Supplier"/>
    <rdfs:range rdf:resource="http://ontology.hpl.hp.com/Logistics/HPModel.owl#Order"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="hasLineItems">
    <rdfs:range rdf:resource="http://ontology.hpl.hp.com/Logistics/HPModel.owl#LineItem"/>
    <rdfs:domain rdf:resource="http://ontology.hpl.hp.com/Logistics/HPModel.owl#Order"/>
    <rdfs:comment>an Order "hasLineItems"</rdfs:comment>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="hasSubproducts">
    <owl:inverseOf>
        <owl:FunctionalProperty rdf:about="http://ontology.hpl.hp.com/Logistics/HPModel.owl#isinaProduct"/>
    </owl:inverseOf>
    <rdfs:range rdf:resource="http://ontology.hpl.hp.com/Logistics/HPModel.owl#SubProduct"/>
    <rdfs:comment>a Product hasSubproducts</rdfs:comment>
    <rdfs:domain rdf:resource="http://ontology.hpl.hp.com/Logistics/HPModel.owl#Product"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="hasID">
    <rdfs:domain rdf:resource="http://ontology.hpl.hp.com/Logistics/HPModel.owl#Order"/>
    <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#FunctionalProperty"/>
    <rdfs:comment>An Order O "hasID" an Identifier I</rdfs:comment>
    <rdfs:range rdf:resource="http://ontology.hpl.hp.com/Logistics/HPModel.owl#OrderId"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="hasUnits">
    <rdfs:range rdf:resource="http://ontology.hpl.hp.com/Logistics/HPModel.owl#ProductItem"/>
    <owl:inverseOf rdf:resource="http://ontology.hpl.hp.com/Logistics/HPModel.owl#IsDefinedBy"/>
    <rdfs:comment>An "abstract" Product "hasUnits" Product Units (concrete materializations)</rdfs:comment>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="hasCustomer">
    <rdfs:range rdf:resource="http://ontology.hpl.hp.com/Logistics/WM.owl#LegalPerson"/>
    <rdfs:comment>An Order O "hasCustomer" a Person P</rdfs:comment>
    <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#FunctionalProperty"/>
</owl:ObjectProperty>
<owl:DatatypeProperty rdf:ID="serialNumber">
    <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#FunctionalProperty"/>
    <rdfs:comment>The Concrete Product will have a concrete Serial Number</rdfs:comment>
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
    <rdfs:domain rdf:resource="http://ontology.hpl.hp.com/Logistics/HPModel.owl#ProductItem"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="description">
```

```xml
        <rdfs:domain rdf:resource="http://ontology.hpl.hp.com/Logistics/HPModel.owl#LineItem"/>
        <rdfs:comment>String used for describe the Line Item</rdfs:comment>
        <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
    </owl:DatatypeProperty>
    <owl:DatatypeProperty rdf:ID="HPPartNumber">
        <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#FunctionalProperty"/>
        <rdfs:domain rdf:resource="http://ontology.hpl.hp.com/Logistics/HPModel.owl#Product"/>
        <rdfs:comment>Internal HP Identification of a Product</rdfs:comment>
        <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
    </owl:DatatypeProperty>
    <owl:DatatypeProperty rdf:ID="quantity">
        <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#int"/>
        <rdfs:comment>Integer containing the quantity</rdfs:comment>
    </owl:DatatypeProperty>
    <owl:DatatypeProperty rdf:ID="numberBoxes">
        <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#FunctionalProperty"/>
        <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#int"/>
        <rdfs:comment>Defines how many number of boxes a product is composed</rdfs:comment>
        <rdfs:domain rdf:resource="http://ontology.hpl.hp.com/Logistics/HPModel.owl#ProductItem"/>
    </owl:DatatypeProperty>
    <owl:FunctionalProperty rdf:ID="isinaProduct">
        <rdfs:range rdf:resource="http://ontology.hpl.hp.com/Logistics/HPModel.owl#Product"/>
        <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#ObjectProperty"/>
        <rdfs:domain rdf:resource="http://ontology.hpl.hp.com/Logistics/HPModel.owl#SubProduct"/>
        <owl:inverseOf rdf:resource="http://ontology.hpl.hp.com/Logistics/HPModel.owl#hasSubproducts"/>
        <rdfs:comment>a Subproduct is in a Product</rdfs:comment>
    </owl:FunctionalProperty>
    <owl:FunctionalProperty rdf:ID="hasWeight">
        <rdfs:range rdf:resource="http://ontology.hpl.hp.com/Logistics/WM.owl#Weight"/>
        <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#ObjectProperty"/>
        <rdfs:comment>a Product has a concrete Weight</rdfs:comment>
    </owl:FunctionalProperty>
    <owl:FunctionalProperty rdf:ID="hasPackId">
        <rdfs:comment>A Pack has a PackID, extremely important, is the ID that HP traces back from Multi-leg
Shipments</rdfs:comment>
        <rdfs:range rdf:resource="http://ontology.hpl.hp.com/Logistics/HPModel.owl#PackId"/>
        <rdfs:domain rdf:resource="http://ontology.hpl.hp.com/Logistics/HPModel.owl#Pack"/>
        <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#ObjectProperty"/>
    </owl:FunctionalProperty>
    <owl:FunctionalProperty rdf:ID="isaPerson">
        <rdfs:range rdf:resource="http://ontology.hpl.hp.com/Logistics/WM.owl#LegalPerson"/>
        <rdfs:domain>
            <owl:Class>
                <owl:unionOf rdf:parseType="Collection">
                    <owl:Class rdf:about="http://ontology.hpl.hp.com/Logistics/HPModel.owl#Consignee"/>
                    <owl:Class rdf:about="http://ontology.hpl.hp.com/Logistics/HPModel.owl#Customer"/>
                    <owl:Class rdf:about="http://ontology.hpl.hp.com/Logistics/HPModel.owl#Supplier"/>
                </owl:unionOf>
            </owl:Class>
        </rdfs:domain>
        <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#ObjectProperty"/>
        <rdfs:comment>a Concept is a Person</rdfs:comment>
    </owl:FunctionalProperty>
    <owl:FunctionalProperty rdf:ID="hasPickID">
        <rdfs:range rdf:resource="http://ontology.hpl.hp.com/Logistics/HPModel.owl#PickId"/>
        <rdfs:domain rdf:resource="http://ontology.hpl.hp.com/Logistics/HPModel.owl#Pick"/>
        <rdfs:comment>a Pick "has a" PickID</rdfs:comment>
        <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#ObjectProperty"/>
    </owl:FunctionalProperty>
    <owl:FunctionalProperty rdf:ID="hasSupplier">
        <owl:inverseOf rdf:resource="http://ontology.hpl.hp.com/Logistics/HPModel.owl#isSupplying"/>
        <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#ObjectProperty"/>
        <rdfs:range rdf:resource="http://ontology.hpl.hp.com/Logistics/HPModel.owl#Supplier"/>
        <rdfs:domain rdf:resource="http://ontology.hpl.hp.com/Logistics/HPModel.owl#Order"/>
```

```
        <rdfs:comment>An order O "hasSupplier" person P</rdfs:comment>
    </owl:FunctionalProperty>
    <owl:FunctionalProperty rdf:ID="isOrdered">
        <rdfs:comment>a concrete product Item is ordered in a concrete LineOrder of an Order. Any Product Item with
no order related still belogs to HP Stock and could have a Warehouse transfer.</rdfs:comment>
        <rdfs:domain rdf:resource="http://ontology.hpl.hp.com/Logistics/HPModel.owl#ProductItem"/>
        <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#ObjectProperty"/>
        <rdfs:range rdf:resource="http://ontology.hpl.hp.com/Logistics/HPModel.owl#LineItem"/>
    </owl:FunctionalProperty>
    <owl:FunctionalProperty rdf:ID="refersTo">
        <rdfs:comment>a Line Item will refer to a single "Product"</rdfs:comment>
        <rdfs:range rdf:resource="http://ontology.hpl.hp.com/Logistics/HPModel.owl#Product"/>
        <rdfs:domain rdf:resource="http://ontology.hpl.hp.com/Logistics/HPModel.owl#LineItem"/>
        <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#ObjectProperty"/>
    </owl:FunctionalProperty>
    <owl:FunctionalProperty rdf:ID="isBoxed">
        <rdfs:domain rdf:resource="http://ontology.hpl.hp.com/Logistics/HPModel.owl#SubProductItem"/>
        <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#ObjectProperty"/>
        <rdfs:comment>A Concrete Product Item "isBoxed" in N boxes.</rdfs:comment>
        <rdfs:range rdf:resource="http://ontology.hpl.hp.com/Logistics/WM.owl#Box"/>
    </owl:FunctionalProperty>
    <owl:FunctionalProperty rdf:ID="isIdentified">
        <rdfs:comment>a COntainer will be identified by a concrete ID in HP to track purpouses</rdfs:comment>
        <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#ObjectProperty"/>
        <rdfs:range rdf:resource="http://ontology.hpl.hp.com/Logistics/HPModel.owl#Identifier"/>
    </owl:FunctionalProperty>
    <owl:FunctionalProperty rdf:ID="number">
        <rdfs:domain>
            <owl:Class>
                <owl:unionOf rdf:parseType="Collection">
                    <owl:Class rdf:about="http://ontology.hpl.hp.com/Logistics/HPModel.owl#Identifier"/>
                    <owl:Class rdf:about="http://ontology.hpl.hp.com/Logistics/HPModel.owl#LineItem"/>
                </owl:unionOf>
            </owl:Class>
        </rdfs:domain>
        <rdfs:comment>Slot to store the Value of the Identifier</rdfs:comment>
        <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
        <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#DatatypeProperty"/>
    </owl:FunctionalProperty>
    <rdf:Description>
        <rdf:rest rdf:parseType="Collection">
            <owl:Class rdf:about="http://ontology.hpl.hp.com/Logistics/HPModel.owl#Product"/>
        </rdf:rest>
    </rdf:Description>
</rdf:RDF>
```

5.5 KSO – Date & Time for scheduling ontology

```
<rdf:RDF xmlns="newDefaultNamespace" xmlns:daml="http://www.daml.org/2000/10/daml-ont#"
xmlns:a="https://www.daml.org/actionitems/actionitems-20000905.rdfs#"
xmlns:num="http://www.daml.org/2000/10/daml-num.daml#" xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#">
    <!-- mark DAML program action item 12 as complete -->
    <rdf:Description about="https://www.daml.org/actionitems/12.rdf">
        <a:state>closed</a:state>
        <a:Action parseType="Resource">
            <a:status>available at http://www.kestrel.edu/DAML/2000/12/TIME.daml</a:status>
            <a:date>12-14-2000 17:44</a:date>
            <a:by>becker@kestrel.edu</a:by>
        </a:Action>
    </rdf:Description>
    <!--      Does importing an ontology adds a new recognizable namespace or we still need to declare the namespace
and use the corresponding  tag to refer to resource-->
    <Ontology about="">
        <versionInfo>$Id$</versionInfo>
        <comment>Kestrel Scheduling Ontology

    This ontology is based on the time ontology available from Stanford's ontoserver.

    Other groups also defined time ontologies in DAML:
    http://www.ai.sri.com/daml/ontologies/sri-basic/1-0/Time.daml
    http://www.cs.cmu.edu/~softagents/atlas/ontologies/atlas-date.daml

    But they do not provide all the entities needed in a scheduling domain.

    </comment>
        <imports resource="http://www.daml.org/2000/10/daml-ont"/>
    </Ontology>
    <Class ID="Temporal-Entity"/>
    <Class ID="Time-Point">
        <comment>

        From KSL - Simple-Time ontology "A time-point is a point in real, historical time (on earth).  It is independent of
observer and context.  A time-point is not a measurement of
        time, nor is it a specification of time.  It is the point  in time. The time-points at which events occur can be known
with various degrees of precision and approximation, but
        conceptually time-points are point-like and not interval-like.  That is, it doesn't make sense to talk about what
happens during a time-point, or how long the
        time-point lasts."  Although one can think of TimePoint as an integer representing an offset from a certain
TimePointconsidered to be the start of the time scale, we will
        represent a time point as a subclass of daml:#Thing andwould use a property to express the value of the time point
with respect to a certain reference point using a certain
        unit of measure or time glanularity.
        </comment>
        <label> Time-Point </label>
        <subClassOf resource="#Temporal-Entity"/>
    </Class>
    <Property ID="time">
        <domain resource="#Time-Point"/>
        <range resource="num:#Integer"/>
        <comment>
         For now, assume the value of a time point to be expressed by an integer representing a number of certain time
units since the origin of the time scale.
         </comment>
    </Property>
    <!-- How do I enumertate the elements of a class ? -->
    <Class ID="Time-Unit">
        <subClassOf resource="num:#Literal"/>
```

```
        <label> Time-Unit</label>
        <comment>

      The Time-Unit represents the granularity  of the time representation. The property time of a TimePoint represents
the number of time units since a certain time point TimeZero.

      </comment>
    </Class>
    <Property ID="time-unit">
        <domain resource="#Time-Point"/>
        <range resource="#Time-Unit"/>
        <comment>

      Using small letters to represent a property that has as domain the class with the same name as the property but
starting with capital letter.

      </comment>
    </Property>
    <!--      How do I say that the value of a Second should be an    integer between 0 and 59 without defining a property
value andrestricting the value of the property using the onProperty and
    toValue construct.  Should the restriction be on the propertydeclaration or in the domain class declaration.  -->
    <Class ID="Month">
        <label>Month</label>
        <subClassOf resource="daml:#Literal"/>
        <comment>
      This has been based on
      http://www.cs.cmu.edu/~softagents/atlas/ontologies/atlas-date.daml.
      But I do not add the month names in the definition of the class
      since the names are just literals.
      Is it correct to define all the member of the class using
      the oneOf construct?
      </comment>
    </Class>
    <Class ID="Between0and59">
        <comment>
     Trick to get the correct restiction on seconds and
     minutes without having to define a class Seconds and Minutes
     Again, how do I express the value of an Integer??
     </comment>
        <label>Between0and56</label>
        <subClassOf resource="num:#Integer"/>
        <restrictedBy>
            <restriction>
                <onProperty resource="num:min"/>
                <hasValue resource="0"/>
            </restriction>
            <restriction>
                <onProperty resource="num:max"/>
                <hasValue resource="59"/>
            </restriction>
        </restrictedBy>
    </Class>
    <Class ID="Between0and23">
        <comment>
     Trick to get the correct restiction on hours without having
     to define a class Hours
     Again, how do I express the value of an Integer??
     </comment>
        <label>Between0and56</label>
        <subClassOf resource="num:#Integer"/>
        <restrictedBy>
            <restriction>
                <onProperty resource="num:min"/>
                <hasValue resource="0"/>
```

```
            </restriction>
            <restriction>
                <onProperty resource="num:max"/>
                <hasValue resource="23"/>
            </restriction>
        </restrictedBy>
    </Class>
    <Class ID="Between1and31">
        <comment>
     Trick to get the correct restiction on days and
     minutes without having to define a class Day.
     Again, how do I express the value of an Integer??
     </comment>
        <label>Between0and31</label>
        <subClassOf resource="num:#Integer"/>
        <restrictedBy>
            <restriction>
                <onProperty resource="num:min"/>
                <hasValue resource="1"/>
            </restriction>
            <restriction>
                <onProperty resource="num:max"/>
                <hasValue resource="31"/>
            </restriction>
        </restrictedBy>
    </Class>
    <Class ID="Day-Of-The-Week">
        <label>dayOfTheWeek</label>
        <subClassOf resource="daml:#Literal"/>
        <comment>The name of the week day.
      </comment>
    </Class>
    <Property ID="second">
        <domain resource="#Date"/>
        <range resource="num:#Integer"/>
        <comment/>
    </Property>
    <Property ID="minute">
        <domain resource="#Date"/>
        <range resource="num:#Integer"/>
        <comment/>
    </Property>
    <Property ID="hour">
        <domain resource="#Date"/>
        <range resource="num:#Integer"/>
        <comment/>
    </Property>
    <Property ID="day">
        <domain resource="#Date"/>
        <range resource="num:#Integer"/>
        <comment/>
    </Property>
    <Property ID="day-Of-The-Week">
        <subPropertyOf resource="#day"/>
        <range resource="#Day-Of-The-Week"/>
    </Property>
    <Property ID="month">
        <domain resource="#Date"/>
        <range resource="#Month"/>
        <comment/>
    </Property>
    <Property ID="monthNumber">
        <subPropertyOf resource="month"/>
        <range resource="num:#Integer"/>
```

```
    </Property>
    <Property ID="year">
        <domain resource="#Date"/>
        <range resource="num:#Integer"/>
        <comment/>
    </Property>
    <Class ID="Calendar">
        <label>Calendar</label>
        <comment>A calendar class provides rules on how to translate the
  time property of a TimePoint object into a Date object expressed in
  terms of month, day, etc..</comment>
    </Class>
    <Property ID="calendar">
        <domain resource="#Date"/>
        <range resource="#Calendar"/>
        <comment>Maybe Date should be a sub-class of Calendar.
   Here I based the ontology on the Java API that defines an
   abstract class Calendar,  and a separate class for representing
   the date. In the Java implementation, however, the class Date is
   equivalent to the class TimePoint.
   </comment>
    </Property>
    <Class ID="Date">
        <comment>
 A date is the entity that provides a representation for a time point in a given calendar. It has properties day, second,minute, hour, year. I will restrict the values of the properties in the class definition. I am here using a construct presented in DAML-OIL and not standard DAML. How do I say that the value of a certain property should be in a certain range??
The default DAML syntax for restriction uses the onProperty/hasValue or onProperty/allValuesFrom construct. The problem with these types of construct is that I can only constrain the range of a property value to belong to a certain class. For example, if I want to say that the value of the property second is an integer between 0 and 59, I need to declare a class, forexample Seconds, and constrain the its values to be between 0 and 59. Then I can use this class in the allValuesFrom construct. The problem is that if I wand to use seconds that go only from 0 to 30, I need to declare a new sub-class of Seconds. How can I restrict only the value without having to create a new class  every time I need to impose restrictions. How can I refer to the value of a certain property?

   </comment>
        <intersectionOf parseType="daml:collection">
            <Restriction>
                <onProperty resource="#second"/>
                <hasValue resource="Between0and59"/>
            </Restriction>
            <Restriction>
                <onProperty resource="#minute"/>
                <hasValue resource="Between0and59"/>
            </Restriction>
            <Restriction>
                <onProperty resource="#hour"/>
                <hasValue resource="Between0and23"/>
            </Restriction>
            <Restriction>
                <onProperty resource="#day"/>
                <hasValue resource="Between1and31"/>
            </Restriction>
        </intersectionOf>
    </Class>
    <!-- Here are some instances of TimePoints and Calendars -->
    <!--
  Definitions for Time Intervals
  How do I express the fact that the start time of an interval
  should be less than the end time?

  -->
    <Class ID="Time-Interval">
```

```
    <subClassOf resource="#Temporal-Entity"/>
    <label>Time-Interval</label>
    <comment>
  This class is characterized
  by a pair of TimePoints and a given duration representing
  the difference between the two time points
  </comment>
    <restrictedBy>
        <restriction>
            <onProperty resource="#start-time-point"/>
            <onProperty resource="#end-time-point"/>
        </restriction>
    </restrictedBy>
</Class>
<Property ID="start-time-point">
    <domain resource="Time-Interval"/>
    <range resource="Time-Point"/>
</Property>
<Property ID="end-time-point">
    <domain resource="Time-Interval"/>
    <range resource="Time-Point"/>
</Property>
<Property ID="duration">
    <domain resource="Time-Interval"/>
    <range resource="num:#Integer"/>
    <comment>
  The duration is just the difference between the endTimePoint and
  the startTimePoint of the interval
  </comment>
</Property>
<!-- CLASS INTERVAL-SEQUENCE -->
<Class ID="Interval-Sequence">
    <label>Interval-Sequence</label>
    <comment> Sequence of time interval defining a temporal profile </comment>
</Class>
<Class ID="Terporal-Relation">
    <label>Temporal-Relation</label>
    <comment>
 The temporal relation is the super class to represent
 relations between Time-Points and Time-Intervals.
  </comment>
</Class>
<Property ID="from">
    <domain resource="#Temporal-Relation"/>
    <range resource="#Termporal-Entity"/>
</Property>
<Property ID="to">
    <domain resource="#Temporal-Relation"/>
    <range resource="#Termporal-Entity"/>
</Property>
<Property ID="lower-bound">
    <domain resource="#Temporal-Relation"/>
    <range resource="daml:#Integer"/>
</Property>
<Property ID="upper-bound">
    <domain resource="#Temporal-Relation"/>
    <range resource="daml:#Integer"/>
</Property>
<!--
In the definition of the Before class I want to define a
property - the temporal relation between two temporal
entities, as a first class object.
I also want to say that:
```

```
1) If the value of the property from and to are Time-Points, then
   Before (t1, t2 ) implies t1.time < t2.time

2) If the value of the property from and to are Time-Intervals,
then Before (I1, I2) implies I1.end-time-point.time <=I2.start-time-point.time

3) If from is a Time-Point and to a Time-Interval, then Before(t1,I1) implies t1.time < I1.start-time-point.time

4) If from is a Time-Interval and to a Time-Point, then

Before(I1, t1) implies I1.end-time-point.time <= t1.time.
-->
<Class ID="Before">
      <label>Before</label>
      <comment/>
      <subClassOf resource="#Temporal-Relation"/>
</Class>
<Class ID="After">
      <label>Before</label>
      <comment/>
      <subClassOf resource="#Temporal-Relation"/>
</Class>
<Property ID="before">
      <domain resource="#Temporal-Entity"/>
      <range resource="#Temporal-Entity"/>
</Property>
<Property ID="after">
      <domain resource="#Temporal-Entity"/>
      <range resource="#Temporal-Entity"/>
      <inverseOf resource="#before"/>
</Property>
<Property ID="same-start">
      <domain resource="#Temporal-Interval"/>
      <range resource="#Temporal-Interval"/>
</Property>
<Property ID="same-end">
      <domain resource="#Temporal-Interval"/>
      <range resource="#Temporal-Interval"/>
</Property>
<Property ID="meets">
      <domain resource="#Temporal-Interval"/>
      <range resource="#Temporal-Interval"/>
      <subPropertyOf resource="#same-start"/>
      <subPropertyOf resource="#same-end"/>
</Property>
<Property ID="overlaps">
      <domain resource="#Temporal-Interval"/>
      <range resource="#Temporal-Interval"/>
</Property>
<Property ID="disjoint">
      <domain resource="#Temporal-Interval"/>
      <range resource="#Temporal-Interval"/>
</Property>
</rdf:RDF>
```

5.6 Location Ontology

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE rdf:RDF [
    <!ENTITY daml 'http://www.w3.org/2002/07/owl#'>
    <!ENTITY xsd 'http://www.w3.org/2001/XMLSchema#'>
    <!ENTITY rdfs 'http://www.w3.org/2000/01/rdf-schema#'>
    <!ENTITY countries-ont 'http://www.daml.org/2001/09/countries/fips-10-4-ont#'>
]>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#" xmlns:rdfs="http://www.w3.org/2000/01/rdf-
schema#" xmlns:owl="&daml;" xml:base="http://www.daml.org/2002/04/geonames/geonames-ont">
    <owl:Ontology rdf:about="">
        <owl:versionInfo>$Id: geonames-ont.daml,v 1.5 2002/09/18 22:34:48 mdean Exp $</owl:versionInfo>
        <rdfs:comment>Geographic Feature Names Ontology, based on NIMA Geonet Names Server</rdfs:comment>
        <rdfs:comment>see http://164.214.2.59/gns/html/index.html</rdfs:comment>
    </owl:Ontology>
    <rdf:Description rdf:about="&countries-ont;Country">
        <rdfs:subClassOf>
            <owl:Restriction>
                <owl:onProperty rdf:resource="#feature"/>
                <owl:allValuesFrom rdf:resource="#Feature"/>
            </owl:Restriction>
        </rdfs:subClassOf>
    </rdf:Description>
    <rdfs:Class rdf:ID="Feature">
        <rdfs:subClassOf>
            <owl:Restriction>
                <owl:onProperty rdf:resource="#uniqueIdentifier"/>
                <owl:allValuesFrom rdf:resource="&xsd;long"/>
            </owl:Restriction>
        </rdfs:subClassOf>
        <rdfs:subClassOf>
            <owl:Restriction>
                <owl:onProperty rdf:resource="#latitude"/>
                <owl:allValuesFrom rdf:resource="&xsd;double"/>
            </owl:Restriction>
        </rdfs:subClassOf>
        <rdfs:subClassOf>
            <owl:Restriction>
                <owl:onProperty rdf:resource="#longitude"/>
                <owl:allValuesFrom rdf:resource="&xsd;double"/>
            </owl:Restriction>
        </rdfs:subClassOf>
        <rdfs:subClassOf>
            <owl:Restriction>
                <owl:onProperty rdf:resource="#primaryCountry"/>
                <owl:allValuesFrom rdf:resource="&countries-ont;Country"/>
            </owl:Restriction>
        </rdfs:subClassOf>
        <rdfs:subClassOf>
            <owl:Restriction>
                <owl:onProperty rdf:resource="#dimension"/>
                <owl:allValuesFrom rdf:resource="&xsd;double"/>
            </owl:Restriction>
        </rdfs:subClassOf>
        <rdfs:subClassOf>
            <owl:Restriction>
                <owl:onProperty rdf:resource="#secondaryCountry"/>
                <owl:allValuesFrom rdf:resource="&countries-ont;Country"/>
            </owl:Restriction>
        </rdfs:subClassOf>
        <rdfs:subClassOf>
```

```
                <owl:Restriction>
                    <owl:onProperty rdf:resource="#name"/>
                    <owl:allValuesFrom rdf:resource="#Name"/>
                </owl:Restriction>
            </rdfs:subClassOf>
            <rdfs:subClassOf>
                <owl:Restriction>
                    <owl:onProperty rdf:resource="#modifyDate"/>
                    <owl:allValuesFrom rdf:resource="&xsd;double"/>
                </owl:Restriction>
            </rdfs:subClassOf>
        </rdfs:Class>
        <rdfs:Class rdf:ID="Name">
            <rdfs:subClassOf>
                <owl:Restriction>
                    <owl:onProperty rdf:resource="#uniqueIdentifier"/>
                    <owl:allValuesFrom rdf:resource="&xsd;long"/>
                </owl:Restriction>
            </rdfs:subClassOf>
            <rdfs:subClassOf>
                <owl:Restriction>
                    <owl:onProperty rdf:resource="#shortName"/>
                    <owl:allValuesFrom rdf:resource="&xsd;string"/>
                </owl:Restriction>
            </rdfs:subClassOf>
            <rdfs:subClassOf>
                <owl:Restriction>
                    <owl:onProperty rdf:resource="#fullName"/>
                    <owl:allValuesFrom rdf:resource="&xsd;string"/>
                </owl:Restriction>
            </rdfs:subClassOf>
        </rdfs:Class>
        <owl:DatatypeProperty rdf:ID="dimension"/>
        <owl:ObjectProperty rdf:ID="feature"/>
        <owl:DatatypeProperty rdf:ID="fullName"/>
        <owl:DatatypeProperty rdf:ID="latitude"/>
        <owl:DatatypeProperty rdf:ID="locatedIn"/>
        <owl:DatatypeProperty rdf:ID="longitude"/>
        <owl:DatatypeProperty rdf:ID="modifyDate"/>
        <owl:ObjectProperty rdf:ID="name"/>
        <owl:ObjectProperty rdf:ID="primaryCountry"/>
        <owl:ObjectProperty rdf:ID="secondaryCountry"/>
        <owl:DatatypeProperty rdf:ID="shortName"/>
        <owl:DatatypeProperty rdf:ID="sortName"/>
        <owl:DatatypeProperty rdf:ID="uniqueIdentifier"/>
    </rdf:RDF>
```

5.7 Payment Ontology


```xml
<rdf:RDF xmlns="http://owl.protege.stanford.edu#" xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#" xmlns:owl="http://www.w3.org/2002/07/owl#">
    <owl:Ontology rdf:about=""/>
    <owl:Class rdf:ID="Direct">
        <rdfs:comment>If the payment is invoiced before the shipment is done</rdfs:comment>
        <rdfs:subClassOf>
            <owl:Class rdf:about="#Means"/>
        </rdfs:subClassOf>
    </owl:Class>
    <owl:Class rdf:ID="BeforeCollection">
        <rdfs:comment>If the payment is made before the shipment is collected</rdfs:comment>
        <rdfs:subClassOf>
            <owl:Class rdf:about="#Terms"/>
        </rdfs:subClassOf>
    </owl:Class>
    <owl:Class rdf:ID="AfterCollection">
        <rdfs:comment>If the payment is made after the shipment is collected</rdfs:comment>
        <rdfs:subClassOf>
            <owl:Class rdf:about="#Terms"/>
        </rdfs:subClassOf>
    </owl:Class>
    <owl:Class rdf:ID="Terms">
        <rdfs:comment>Diferent terms of a payment</rdfs:comment>
    </owl:Class>
    <owl:Class rdf:ID="Credited">
        <rdfs:subClassOf>
            <owl:Class rdf:about="#Means"/>
        </rdfs:subClassOf>
        <rdfs:comment>If the payment is invoiced after the shipment is done</rdfs:comment>
    </owl:Class>
    <owl:Class rdf:ID="Means">
        <rdfs:comment>The different means that could be used to express a payment</rdfs:comment>
    </owl:Class>
    <owl:FunctionalProperty rdf:ID="isDeferred">
        <rdfs:domain rdf:resource="#Means"/>
        <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#boolean"/>
        <rdfs:comment>Boolean that would express if the payment is done after the shipment</rdfs:comment>
        <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#DatatypeProperty"/>
    </owl:FunctionalProperty>
    <owl:FunctionalProperty rdf:ID="name">
        <rdfs:domain>
            <owl:Class>
                <owl:unionOf rdf:parseType="Collection">
                    <owl:Class rdf:about="#Terms"/>
                    <owl:Class rdf:about="#Means"/>
                </owl:unionOf>
            </owl:Class>
        </rdfs:domain>
        <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#DatatypeProperty"/>
        <rdfs:comment>String that includes the identification</rdfs:comment>
        <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
    </owl:FunctionalProperty>
</rdf:RDF>
```

6 References

---

[1] Javier Esplugas Cuadrado, *WP8 Case Study Requirements*,

http://swws.semanticweb.org/private_doc/D8.1.pdf

[2] The OWL Services Coalition, *OWL-S: Semantic Markup for Web Services,*

http://www.daml.org/services/owl-s/1.0/owl-s.pdf

[3] W3C, *Web Services Description Language (WSDL) 1.1*, http://www.w3.org/TR/wsdl

[4] T. R. Gruber. *A translation approach to portable ontologies*. Knowledge Acquisition, 5(2):199-220, 1993. http://ksl-web.stanford.edu/KSL_Abstracts/KSL-92-71.html

[5] Krestel Institute, *Time Ontology for scheduling*, http://www.kestrel.edu/DAML/2000/12/TIME.daml

[6] *Geographic Feature Names Ontology, based on NIMA Geonet Names Server* http://www.daml.org/2002/04/geonames/geonames-ont