Abstract

This document specifies a concrete use case for modeling Semantic Web Services with the Web Service Modeling Ontology WSMO in the domain of e-tourism. A Virtual Travel Agency sells tickets for international train tickets, and a customer defines a Goal for purchasing such a ticket. This use case has been the initial WSMO Use Case defined in previous versions of the WSMO D3.2 Deliverable - WSMO Use Case and Testing. The main focus of this document is the concrete modeling of the top WSMO components, resulting in the specification of WSML, and to test and elaborate the approach and technologies for Web Service Discovery in WSMO.

For use case modeling, we stick to the final working draft of Web Service Modeling Ontology WSMO, Version 1.0, 20 September 2004 [Roman et al., 2004].

Related Documents
1. Introduction

This use case is defined in the domain of e-tourism: a Virtual Travel Agency sells tickets for international train tickets, and a customer defines a Goal for purchasing such a ticket. Using the conceptual framework of WSMO we specify the top-level notions of Ontologies, Goals, Web Services and Mediators for this use case.

A Web Service of a Virtual Travel Agency, short: VTA, offers end-user services for searching and buying train tickets for itineraries in Austria and in Germany. This Web Service is composed out of other Web Services, namely one for searching existing train connections, and one for purchasing train tickets online. As a user request we assume that the user wants to purchase an international train ticket. The course of the use case shall be the following:
the customer creates specifies his high level goal of purchasing international train tickets.
the VTA is one of the Service Providers that is identified as suitable fulfilling his request
the customer refines his request to an international train ticket valid from Innsbruck to Frankfurt on 17th July 2004, between 6 and 7 p.m. local time
the VTA returns a set of possible connections
the user selects one of these connections and poses a request for booking the ticket online
the VTA combines the online train ticket booking services from ÖBB and DB, executes the booking and payment process, and sends an online ticket per email to the Customer.

This document is structured as follows: Section 2 gives an overview of the use case, describing the overall setting for of the use case and identifying the technical challenges arising for Semantic Web Service technologies; Section 3 defines the needed WSMO components and provides the WSML models for the distinct WSMO components of the use case along with explanations of the design and modeling decisions; Section 4 explains the WSMO Discovery as realized within this use case; finally, Section 5 concludes the use case. Appendix A provides a change tracking to previous version of the document; Appendix B provides related resources for this use case.

2. Use Case Overview

According to the general framework for defining use cases for WSMO defined in WSMO Use Case Overview document, section 2, this section provides a description of the setting of this use case. In [He et al., 2004], the travel agency use case is separated into two use cases - one with static discovery and one with automated discovery. With Semantic Web Services we clearly want to support automated discovery.

2.1 Description

Imagine a “Virtual Traveling Agency”, called VTA for short, which is an end user service providing eTourism services to customers. These services can cover all kinds of information services concerned with tourism information - from information about events and sights in an area to services that support booking of flights, hotels, rental cars, etc. online. Such VTAs are already existent, currently those portals are accessible via html sites. The partners of the VTA are integrated via conventional B2B integration. By applying Semantic Web Services, a VTA will be more easily be able to configure an invoke Web Services provided by several eTourism suppliers and aggregate them into new customer services. Such VTAs providing automated eTourism services to end users via different interfaces and can be more easily reconfigured according to the actual needs..

Our VTA use case that aggregates Web Services of different tourism service providers. In a nutshell shall it provides the following functionality: A customer uses the VTA service as the entry point for his requests. These end-user services are aggregated by the VTA by invoking and combining Web Services offered by several tourism service providers. To facilitate this, there can be a so called "umbrella" contract between the service providers and the VTA for regulating usage and allowance of the Web Services. Figure 2 gives an overview (modified and extended from W3C Travel Agent Use Case overview, as defined in [He et al., 2004]).
2.2 Scope

The scenario outlines a general structure for VTAs that can be extended to more complex scenarios wherein the customer can be a Web Service itself, thus creating a network of composed services that offer complex tourism services. For example, one VTA can provide flight booking services for an airline union, another VTA aggregates booking service for a worldwide hotel chain, and a third VTA provides booking services for rental cars by combining the services of several worldwide operating car rental agencies. Then, another VTA uses these services for providing an end-user service for booking complete holiday trips worldwide.

2.3 Actors, Roles and Goals

In the general use case there are 3 actors. The following defines why they participate in this use case (goal) and the particular interactions they are involved in (roles).

1. **Customer**: the end-user that requests a service provided by the VTA
   - **Goal**: automated resolution of the request by a user-friendly tourism service
   - **Role**: end-user, interacts with VTA for service usage, payment, and non-computational assets (e.g. receiving the actual ticket when booking a trip)

2. **Tourism Service Providers**: commercial companies that provides specific tourism services
   - **Goal**: sell service to end customers, maximize profit as a commercial company
   - **Role**: provides tourism service as a Web Service (also provides the necessary semantic descriptions of the Web Services), may have a usage and allowance contract with the VTA

3. **VTA**: the intermediate between the Customer and the Tourism Service Providers. It provides tourism services to customers by aggregating the separate services provided by the single Service Providers.
   - **Goal**: provide high-quality end-user tourism services, uses existing tourism services and aggregates these into new services, maximize profit as a commercial company / represent union of service providers (depending on the owners of the VTA).
   - **Role**: interacting with customer via user interface (can be web-based for direct human customers interaction or via Web Services for machine-users), usage and allowance contract for Web Services offered by Service Providers, centrally holding all functionalities for handling Semantic Web Services (mechanisms for discovery, composition, execution, etc.)

2.4 Usage Scenarios
We identify the following usage scenarios

1. **VTA interacts with Service Providers on contract and Web Service usage and allowance**
   - **Participating Actors:** VTA and Service Providers
   - **Activities:** business contract negotiation
   - **Technological Requirements:** contract information requirements are modeled in the system, i.e. Web Service usage is implemented via Policies
   - **Possible Extensions:** contract negotiation can be supported by automated mechanisms

2. **Customer requests VTA for searching tourism service offers, VTA detects and queries suitable Web Services and forwards results to Customer**
   - **Participating Actors:** Customer, VTA, Tourism Service Providers
   - **Activities:**
     1. Customer selects "Search" services as provided by the VTA
     2. VTA discovers, invokes and executes corresponding Web Services
   - **Technological Requirements:**
     1. VTA has to pre-define the "Search" functionality that can be requested by a Customer
     2. the Tourism Service Providers' Web Services must be semantically described in order to support dynamic discovery (assuming that single Web Services can perform the search functionality)
     3. VTA has to provide mechanisms for automated Service Discovery
   - **Possible Extensions:**
     - the Customer specifies its request in natural language and the request is translated into machine readable form and processed by the VTA service automatically

3. **Customer selects a concrete offer and requests booking for this offer (interacting with the VTA), VTA detects and aggregates Web Services for booking (incl. booking, payment, etc.), displays result to Customer and handles complete execution of customer-interaction (computational part)**
   - **Participating Actors:** Customer, VTA, Tourism Service Providers
   - **Activities:**
     1. Customer selects one concrete offer out of the Search results of usage scenario 2
     2. VTA discovers and composes available Web Services from Service Providers and composes them into the functionality to satisfy the user request
     3. VTA executes the Web Services in the sequence determined, controls the execution (handles errors and detects alternative paths if a Web Service fails)
     4. VTA interacts with Customer during execution when further information is needed (e.g. a credit card number for payment)
   - **Technological Requirements:**
     1. Web Services must be semantically described in order to support dynamic discovery, composition, and execution
     2. VTA has to hold mechanisms for automated Service Discovery, Composition, and Execution
     3. VTA has to provide and interaction interface for contingent Customer-interaction during Service execution
   - **Possible Extensions:** advanced mechanisms for automated execution of aggregated Web Services

4. **VTA interacts with Customer and Service Provider for non-computational parts (e.g. delivery of actual tickets)**
   - **Participating Actors:** Customer, VTA, Tourism Service Providers
   - **Activities:** customer notification, accounting, good delivery (out of computational system), etc.
   - **Technological Requirements:** mechanisms for notification and accounting
   - **Possible Extensions:** Web Services can be used for:
     - customer notification
     - VTA-Service Provider interaction on accounting and good delivery mandate
2.5 System Architecture

In this use case, the VTA is the central point of interaction between the Customer and other Web Services. Regarding the technological requirements, it is obvious from the usage scenario descriptions that (1) the Web Services offered by the Service Providers have to carry sufficient descriptive information to support automated Web Service usage, and (2) that the VTA has to provide all mechanisms to handle Semantic Web Services. The basic architecture of such a VTA as a central entity for Semantic Web Services handling is shown in Figure 3. The essential functionalities of Semantic Web Service enabled VTAs – with special regard to the requirements for Semantic Web Service technologies – are:

- It has to provide a user interface for customer interaction (for both human and machine users)
- It has to discover suitable Web Services for an user request
- It has to invoke and combine external Semantic Web Services
- It has to provide a Web Service Execution Environment with control functions, error handling, and support for optional user interaction
- It has to deal properly with heterogeneous resources, thus allowing for appropriate mediation facilities.
- It has to provide interfaces for cooperation with Service Providers.

Figure 3. General Architecture of a SWS-enabled VTA

Summarizing, the VTA is a SWS-enabled B2C application that provides an end-user service following a client/server model. In order to support coherent functionality of the VTA and to ensure that the descriptions of Web Services are compatible to this, an overall framework for SWS technologies is needed. This is provided by WSMO.

Semantic Web Service technology can be applied to ease the described use case (a) for the
customer (to find the VTA and resolve his goal) and (b) for the VTA that benefits from semantic description of its business partners. These ease the combination and rearrangement of existing services for the VTA and allows him to flexibly offer new services. In case composition and discovery have reached a higher level of maturity it can even be thought that the technologies (in this case discovery and composition) can be used directly by the end user. So that he himself combines the travel services offered, thus not using the extra intermediate party (the VTA).

3. WSMO Use Case Modeling

This section exemplifies the specification of this use case within the Web Service Modeling Ontology WSMO. The provided listings use the conceptual model and syntax presented in WSMO, final version 1.0 [Roman et al., 2004]. The listings provide have been validated with the WSML Online Validation Service.

The subsequent sections provide the modeling of the resources along with detailed explanations on modeling decisions and related issues. The following tables provide an overview of the resources specified in this use case.

<table>
<thead>
<tr>
<th>Table 1. &quot;International Train Ticket Ontology&quot;</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>WSMO component type</strong></td>
</tr>
<tr>
<td><strong>name</strong></td>
</tr>
<tr>
<td><strong>description</strong></td>
</tr>
</tbody>
</table>
| **imported ontologies / used mediators** | - Date and Time Ontology  
- Location Ontology  
- OWL Person Mediator  
- OWL Fact Book Mediator |
| **main constructs** | **main concepts:**  
station, itinerary, ticket, trip, traintrip  
**axioms:**  
stationCountry, departureBeforeArrival, startNotEqualEnd |
| **WSML model** | Listing 1 |
### Table 2. "Date and Time Ontology"

<table>
<thead>
<tr>
<th>WSMO component type</th>
<th>ontology</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>Date and Time Ontology</td>
</tr>
<tr>
<td>description</td>
<td>defines notions of date and time, and general interdependencies between them</td>
</tr>
<tr>
<td>imported ontologies / used mediators</td>
<td>none</td>
</tr>
<tr>
<td>main constructs</td>
<td>main concepts: instant, interval, date, time, dateandtime</td>
</tr>
<tr>
<td></td>
<td>functions: julianDayNumber, daysBetween, secondsBetween, secondsFromMidnight</td>
</tr>
<tr>
<td></td>
<td>relations: contains (for intervals)</td>
</tr>
<tr>
<td></td>
<td>axioms: equality / before / after / between for date and time notions, integrity contraints</td>
</tr>
<tr>
<td>WSML model</td>
<td>Listing 2</td>
</tr>
</tbody>
</table>

### Table 3. "Purchase Ontology"

<table>
<thead>
<tr>
<th>WSMO component type</th>
<th>ontology</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>Purchase Ontology</td>
</tr>
<tr>
<td>description</td>
<td>defines notions of purchasing, incl. buyer, seller, purchase order, purchase partners, payment notions, etc.; based on RosettaNet, but adopted for B2C setting</td>
</tr>
</tbody>
</table>
| imported ontologies / used mediators | - Date an Time Ontology  
- OWL Currency Mediator |
| main constructs | main concepts: purchase, buyer, seller, contactInformation, purchaseOrder, product, paymentMethod, paymentTerms, delivery |
| WSML model          | Listing 3 |

### Table 4. "Location Ontology"

<table>
<thead>
<tr>
<th>WSMO component type</th>
<th>ontology</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>Location Ontology</td>
</tr>
<tr>
<td>description</td>
<td>defines geographical notions and postal address; extends / combines existing domain ontologies</td>
</tr>
</tbody>
</table>
| imported ontologies / used mediators | - OWL Fact Book Mediator  
- OWL Address Mediator  
- OWL Geo Mediator |
main constructs

<table>
<thead>
<tr>
<th>main concepts:</th>
</tr>
</thead>
<tbody>
<tr>
<td>country, address, city, state, border, distance; incl. concepts / attributes imported via mediators</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>functions:</th>
</tr>
</thead>
<tbody>
<tr>
<td>distanceInKilometers, distanceInMiles,</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>relations:</th>
</tr>
</thead>
<tbody>
<tr>
<td>equalDistance, lessThanDistance, moreThanDistance</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>axioms:</th>
</tr>
</thead>
<tbody>
<tr>
<td>integrity contraints</td>
</tr>
</tbody>
</table>

WSML model | Listing 4

Table 5. "VTA Use Case Knowledge Base"

<table>
<thead>
<tr>
<th>WSMO component type</th>
<th>ontology</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>VTA Use Case Knowledge Base</td>
</tr>
<tr>
<td>description</td>
<td>holds all pre-defined instance data needed within the use case</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>imported ontologies / used mediators</th>
</tr>
</thead>
<tbody>
<tr>
<td>- Train Connection Ontology</td>
</tr>
<tr>
<td>- Purchase Ontology</td>
</tr>
<tr>
<td>- Date and Time Ontology</td>
</tr>
<tr>
<td>- Location Ontology</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>main constructs</th>
<th>instances for:</th>
</tr>
</thead>
<tbody>
<tr>
<td>stations, currentDate, credit card types, locations (continents, countries, states, cities), drop ship carriers, transportations means</td>
<td></td>
</tr>
</tbody>
</table>

| WSML model | Listing 45 |

Table 6. "General Goal for buying a ticket online"

<table>
<thead>
<tr>
<th>WSMO component type</th>
<th>goal</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>Buying a train ticket online</td>
</tr>
<tr>
<td>description</td>
<td>defines the general ontological structure of goal notions for buying tickets online according to the ontologies; this goal serves as a template for concrete goals (see table 6).</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>imported ontologies / used mediators</th>
</tr>
</thead>
<tbody>
<tr>
<td>- Train Connection Ontology</td>
</tr>
<tr>
<td>- Purchase Ontology</td>
</tr>
<tr>
<td>- Location Ontology</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>main constructs</th>
<th>postcondition:</th>
</tr>
</thead>
<tbody>
<tr>
<td>purchase a ticket for an itinerary (a trip and a passenger)</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>effect:</th>
</tr>
</thead>
<tbody>
<tr>
<td>get the purchased ticket delivered to the buyer</td>
</tr>
</tbody>
</table>

| WSML model | Listing 6 |
Table 7. "Concrete Goal for buying a ticket online"

<table>
<thead>
<tr>
<th>WSMO component type</th>
<th>goal</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>Buying a train ticket from Innsbruck to Frankfurt</td>
</tr>
<tr>
<td>description</td>
<td>Buying a train ticket online for an itinerary from Innsbruck to Frankfurt for a specific passenger with payment by credit card. This goal is derived by specializing of the general goal (table 5) via an GG Mediator (table 12)</td>
</tr>
</tbody>
</table>
| imported ontologies / used mediators | - Date and Time  
- GG Mediator 1 |
| main constructs     | postcondition: purchase a ticket for an itinerary for a trip from Innsbruck to Frankfurt on 17th July 2004 for Tim Berners Lee, departure between 6 and 7 p.m.  
effect: get the purchased ticket delivered to the buyer |
| WSML model          | Listing 7 |

Table 8. "ÖBB Web Service for selling international train tickets online"

<table>
<thead>
<tr>
<th>WSMO component type</th>
<th>web service</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>ÖBBB Online Ticket Booking Web Service</td>
</tr>
<tr>
<td>description</td>
<td>web service for booking online train tickets for Austria and Germany, offered by the ÖBB (Austrian national train operator)</td>
</tr>
</tbody>
</table>
| imported ontologies / used mediators | - Train Connection Ontology  
- Purchase Ontology  
- Date and Time Ontology  
- Location Ontology |
| main constructs     | precondition: a buyer, and information on the trip that a ticket is searched and sold for and the passenger for whom the ticket shall be valid  
assumption: if the payment method is credit card, then the credit card has to be valid (i.e. not expired)  
postcondition: a purchase for a ticket by ÖBB as provider, for itineraries valid for a train trip with start- and end location in Austria or Germany and for a passenger, incl. price, and payment only via credit card  
effect: delivery of sold ticket by drop ship carrier or by online delivery |
| WSML model          | Listing 8 |
### Table 9. "OWL Address Mediator"

<table>
<thead>
<tr>
<th>WSMO component type</th>
<th>oo mediator</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>importing the OWL Factbook ontology to WSML</td>
</tr>
<tr>
<td>description</td>
<td>Mediator to import an OWL address ontology into a WSML locations ontology</td>
</tr>
<tr>
<td>imported ontologies / used mediators</td>
<td></td>
</tr>
<tr>
<td>main constructs</td>
<td>source component: an Address Ontology in OWL target component: Location Ontology (Table 4) mediation service: not specified</td>
</tr>
<tr>
<td>WSML model</td>
<td>Listing 9</td>
</tr>
</tbody>
</table>

### Table 10. "OWL Currency Mediator"

<table>
<thead>
<tr>
<th>WSMO component type</th>
<th>oo mediator</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>importing the OWL Currency ontology to WSML</td>
</tr>
<tr>
<td>description</td>
<td>Mediator to import an OWL currency ontology into a WSML purchase order ontology</td>
</tr>
<tr>
<td>imported ontologies / used mediators</td>
<td></td>
</tr>
<tr>
<td>main constructs</td>
<td>source component: an Currency Ontology in OWL target component: Purchase Ontology (Table 3) mediation service: not specified</td>
</tr>
<tr>
<td>WSML model</td>
<td>Listing 10</td>
</tr>
</tbody>
</table>

### Table 11. "OWL Factbook Mediator"

<table>
<thead>
<tr>
<th>WSMO component type</th>
<th>oo mediator</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>importing the OWL Factbook ontology to WSML</td>
</tr>
<tr>
<td>description</td>
<td>Mediator to import an OWL factbook ontology into a WSML ontology</td>
</tr>
<tr>
<td>imported ontologies / used mediators</td>
<td></td>
</tr>
<tr>
<td>main constructs</td>
<td>source component: the OWL Factbook (in OWL) target component: Location Ontology (Table 4) mediation service: not specified</td>
</tr>
<tr>
<td>WSML model</td>
<td>Listing 11</td>
</tr>
</tbody>
</table>

### Table 12. "OWL Person Mediator"

<table>
<thead>
<tr>
<th>WSMO component type</th>
<th>oo mediator</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>importing the OWL Person ontology to WSML</td>
</tr>
</tbody>
</table>
3.1 Ontologies

With regard to modularized ontologies as a basic design principle of WSMO, we define four separate domain ontologies as the terminology definitions for the use case:

1. "International Train Ticket" describes the domain of train tickets
2. "Date and Time" defines a general model for specifying time and dates and relationships of them
3. "Purchase" describes generic elements of purchasing a product between a buyer and a seller.
4. "Locations" describes locations (such as continents, countries and cities and their interrelation).

The ontologies specified in the following are intended to be "real ontologies" in the sense that they describe the specific domain as a shared conceptualization in a sufficient manner. This allows to reuse this ontologies in different settings and use cases - for example, notions or date and time or a general purchase ontology are needed in a lot of other possible scenarios. However, we do not claim the defined below to be such generic ontologies, but they will be enhanced and completed within cooperations with other use cases, projects, and initiatives. Appendix B provides the original ontologies that are used within the domain ontologies specified in this section.

At this point in time, WSMO does not provide a technique to link to large set of instances. Therefore, in this version of the ontology we only include some example instances, which holds for the other ontologies defined in this use case as well. For reader's convenience, the relevant
instances for this use case are gathered in the VTA Use Case Knowledge Base, a separate ontology. The inclusion of links to large set of instances will be considered in future versions of WSMO.

3.1.1 International Train Ticket Ontology

The "International Train Ticket" Ontology defines a train trip and the surrounding concepts as defined the WSML definition of the ontology shown in Listing 1.

The definition of the ontology is based on the travel itinerary ontology from the DAML ontology library, which is also available in Appendix B6 in OWL abstract syntax. The ontology defines travel itineraries for trips by plane. Our ontology reuses the itinerary and flight concepts and adapt them to define train trips, also introducing new concepts such as train station. The international train ticket ontology also makes use of the person ontology defined at http://daml.umbc.edu/ontologies/ittalks/person (Appendix B1), which defines a subset of vCard. The person concept is used to define the passenger information for an itinerary. We did not find any other available ontologies that model the domain of train tickets or itineraries. The first version of the harmonize ontology for the tourism domain focuses on the events and accommodations subdomains. We will take into account future versions of the harmonise ontology, as they are likely to include the travelling subdomain.
Listing 1. Domain Ontology “International Train Ticket”

```xml
namespace <<http://www.wsmo.org/ontologies/trainConnection>>
dc:<http://purl.org/dc/elements/1.1#>
dt:<http://www.wsmo.org/ontologies/dateTime#>
prs:<http://www.wsmo.org/2004/d3/d3.3/v0.1/20041008/resources/owlPersonMediator.wsml>
loc:<http://www.wsmo.org/ontologies/location#>
geo:<http://www.wsmo.org/2004/d3/d3.3/v0.1/20041008/resources/owlGeoMediator.wsml#>
xsd:<http://www.w3.org/2001/XMLSchema#>
ontology <<http://www.wsmo.org/ontologies/trainConnection>>

nonFunctionalProperties
  dc:title hasValue "International Train Connections Ontology"
  dc:creator hasValue "DERI International"
  dc:subject hasValues {"Train", "Itinerary", "Train Connection", "Ticket"}
  dc:description hasValue "International Train Connections"
  dc:publisher hasValue "DERI International"
  dc:date hasValue "2004-10-08"
  dc:type hasValue <<http://www.wsmo.org/2004/d2#ontologies>>
  dc:format hasValue "text/html"
  dc:identifier hasValue <<http://www.wsmo.org/ontologies/trainConnection>>
  dc:language hasValue "en-US"
    <<http://www.wsmo.org/ontologies/dateTime>>, <<http://www.wsmo.org/ontologies/location>>,
    <<http://www.daml.org/2001/02/geofile/geofile-ont>>,
    <<http://www.daml.org/2001/02/geofile/geofile-ont>>}
  dc:coverage hasValue "ID:7029392 Name:World"
  dc:rights hasValue "<http://www.deri.org/privacy.html>"
  version hasValue "$Revision: 1.6 $"
endNonFunctionalProperties

importedOntologies {<<http://www.wsmo.org/ontologies/dateTime>>,
  <<http://www.wsmo.org/ontologies/location>>}

usedMediators {<<http://www.wsmo.org/2004/d3/d3.3/v0.1/20041008/resources/owlPersonMediator.wsml>,
  <<http://www.wsmo.org/2004/d3/d3.3/v0.1/20041008/resources/owlFactBookMediator.wsml>>}

concept station subConceptOf geo:geographicLocation
  nonFunctionalProperties
  dc:description hasValue "Train station"
endNonFunctionalProperties

concept ticket
  nonFunctionalProperties
  dc:description hasValue "a ticket for an itinerary"
endNonFunctionalProperties

concept itinerary
  nonFunctionalProperties
  dc:description hasValue "An itinerary between two locations"
endNonFunctionalProperties

concept trip
  nonFunctionalProperties
  dc:description hasValue "An itinerary between two locations"
endNonFunctionalProperties

concept passenger
  nonFunctionalProperties
  dc:description hasValue "prs:person is a subset of vCard (http://www.ietf.org/rfc/rfc2425.txt)"
endNonFunctionalProperties

concept recordLocatorNumber
  nonFunctionalProperties
  dc:description hasValue "For stations located at the border"
endNonFunctionalProperties

concept start
  nonFunctionalProperties
  dc:description hasValue "The start of the trip"
endNonFunctionalProperties

concept end
  nonFunctionalProperties
  dc:description hasValue "The end of the trip"
endNonFunctionalProperties
```

D3.3 WSMO Use Case "Virtual Travel Agency" v0.1

http://www.wsmo.org/2004/d3/d3.3/v0.1/

24.11.2004 10:59
via ofType set loc:location
departure ofType dt:dateTime
duration ofType dt:interval
distance ofType loc:distance

concept trainTrip subConceptOf trip
nonFunctionalProperties
dc:description hasValue "A train trip"
endNonFunctionalProperties
start ofType station
end ofType station
via ofType set station
seat ofType xsd:string
train ofType xsd:string
class ofType xsd:string

axiom stationCountry
nonFunctionalProperties
dc:description hasValue "Integrity constraint: if a station is located in a place
which is located in a given country, the country of the station is the
same"
endNonFunctionalProperties
definedBy
constraint
?S [locatedIn hasValue ?L,
country hasValue ?C]
]memberOf station
and not ?L[

country hasValue ?C
]memberOf loc:location .

axiom departureBeforeArrival
nonFunctionalProperties
dc:description hasValue "Integrity Constraint: departure has to be before arrival"
endNonFunctionalProperties
definedBy
constraint
?T[
departure hasValue ?D,
arrival hasValue ?A
]memberOf trip
and ?A <= ?D.

axiom startNotEqualEnd
nonFunctionalProperties
dc:description hasValue "Integrity Constraint: the start and end of a trip have to be different"
endNonFunctionalProperties
definedBy
constraint
?T[
start hasValue ?Start,
end hasValue ?End
]memberOf trip

3.1.2 Date and Time Ontology

The "Date and Time Ontology" in Listing 2 defines models for dates (i.e. certain days) and time
(i.e. definition of certain points in time). Further, it defines axioms that represent conventional
aspects of date and time, like ‘before’ and ‘after’, etc. In the use case, this is needed to
determine validity of train connections, e.g for ensuring that a ticket is not for an itinerary that is
in the past. It also can be used generally for expressing dates and time and relationships
between them.

The main ontology taken into consideration for developing this conceptual model of Date and
Time is an entry sub-ontology of time, available at
http://www.isi.edu/~pan/damltime/time-entry.owl. This ontology uses abstract temporal concepts
like instant, interval and event and uses the Gregorian calendar as representation (partly using
own encoding and partly using XSD encoding). Axioms are defined in first order logic in the
accompanying paper [Pan and Hobbs]; there also is a LISP version of these axioms available at http://www.cs.rochester.edu/~ferguson/daml/daml-time-20030728.lisp. Other ontologies like COBRA calenderclock ontology (http://daml.umbc.edu/ontologies/cobra/0.4/calendarclock) are only a straight forward representation of the Gregorian calendar, without any abstraction of concepts and description of axioms. Widely used concrete representations for date and time are defined in ISO 8601 (Numeric representation of Dates and Time) and in the XML Schema Definition (http://www.w3.org/TR/xmlschema-2/), which is based on ISO 8601. In a later stage when it is clear which build in predicates can be used we will add a syntactical mapping to xsd:dateTime.
Listing 2. Domain Ontology "Date and Time"

namespace
dc: <http://purl.org/dc/elements/1.1#>
targetNameSpace: <http://www.wsmo.org/ontologies/dateTime#>
ontology <http://wsmo.org/ontologies/dateTime/>

nonFunctionalProperties
dc:title hasValue "Date and Time Ontology"
dc:creator hasValue "DERI International"
dc:subject hasValues ["Date", "Time", "Date and Time Algebra"]
dc:publisher hasValue "DERI International"
dc:date hasValue "2004-10-04"
dc:type hasValue <http://www.wsmo.org/2004/d2/v1.0/#ontologies>.
dc:format hasValue "text/html"
dc:language hasValue "en-US"
dc:relation hasValues [<<http://www.isi.edu/~pan/damltime/time-entry.owl>>,
<<http://www.w3.org/TR/xmlschema-2/>>]
dc:coverage hasValue "World"
dc:rights hasValue <http://www.deri.org/privacy.html>.
version hasValue "$Revision: 1.2 $"
endNonFunctionalProperties

comment: conceptDefinitions

concept instant
  nonFunctionalProperties
dc:description hasValue "An instant represents a particular point in time and is the super concept of all concrete representations such as the Gregorian calendar"
endNonFunctionalProperties

concept interval
  nonFunctionalProperties
dc:description hasValue "An interval represents a duration between 2 points in time"
endNonFunctionalProperties

concept date subConceptOf instant
  nonFunctionalProperties
dc:description hasValue "concept date and its representation according to the Gregorian Calendar"
endNonFunctionalProperties
dayOfMonth ofType dayOfMonth
monthOfYear ofType monthOfYear
year ofType year

calendar year subConceptOf xsd:integer
  nonFunctionalProperties
dc:description hasValue "year is represented by an integer"
endNonFunctionalProperties

calendar monthOfYear subConceptOf xsd:integer
  nonFunctionalProperties
dc:description hasValue "monthOfYear is represented by an integer"
endNonFunctionalProperties

calendar dayOfMonth subConceptOf xsd:integer
  nonFunctionalProperties
dc:description hasValue "dayOfMonth is represented by an integer"
endNonFunctionalProperties

calendar time
  hourOfDay ofType hourOfDay
  minuteOfHour ofType minuteOfHour
  secondOfMinute ofType secondOfMinute

concept secondOfMinute subConceptOf xsd:integer
  nonFunctionalProperties
dc:description hasValue "a secondOfMinute is represented by an integer"
endNonFunctionalProperties

concept minuteOfHour subConceptOf xsd:integer
  nonFunctionalProperties
dc:description hasValue "a minuteOfHour is represented by an integer"
endNonFunctionalProperties

concept hourOfDay subConceptOf xsd:integer
  nonFunctionalProperties
dc:description hasValue "a hourOfDay is represented by an integer"
endNonFunctionalProperties
concept dateAndTime subConceptOf instant
    nonFunctionalProperties
    dc:description hasValue "concept date and time and representing together a specific point of time (instant)"
endNonFunctionalProperties
date ofType date
time ofType time

comment: functionDefinitions
function julianDayNumber
    nonFunctionalProperties
    dc:description hasValue "The Julian Day Count is a uniform count of days from a remote epoch in the past (about 4712 BC). At this instant, the Julian Day Number is 0. Once you have the Julian Day Number of a particular date in history, it is easy to calculate time elapsed between it and any other Julian Day Number"
    dc:source hasValue "<http://quasar.as.utexas.edu/BillInfo/JulianDatesG.html>"
    dc:description hasValue "For each instant there should exist a corresponding Julian Day Number, however it may not be always defined only by this binary predicate, e.g. if the instant is represented as Gregorian Date and it is a date between 1582 and 1924 a country must be given as third parameter (since e.g. Greece changed no earlier than 9th of March 1924 from the Julian to the Gregorian Calendar)"
    dc:source hasValue "<http://members.brabant.chello.nl/~h.reints/cal/whenjul2greg.htm>"
endNonFunctionalProperties
instant ofType instant
range ofType xsd:integer

function daysBetween
    nonFunctionalProperties
    dc:description hasValue "(Instant1, Instant2, Difference) is a triple of the ternary relation corresponding to this function iff Instant1 and Instant2 are members of the concept instant (particular point in time) and Instant2 is Difference days after Instant1."
endNonFunctionalProperties
instant1 ofType instant
instant2 ofType instant
range ofType xsd:integer

function secondsBetween
    nonFunctionalProperties
    dc:description hasValue "(Instant1, Instant2, Difference) is a triple of the ternary relation corresponding to this function iff Instant1 and Instant2 are members of the concept instant (particular point in time) and Instant2 is Difference seconds after Instant1."
endNonFunctionalProperties
instant1 ofType instant
instant2 ofType instant
range ofType xsd:integer

function secondsFromMidnight
    nonFunctionalProperties
    dc:description hasValue "(Time, SecondsFromMidnight) is a tuple of the binary relation corresponding to this function iff SecondsFromMidnight are the seconds elapsed from 00:00:00 of the same day. This simplifies the axiomatization of the difference between two given times"
endNonFunctionalProperties
time ofType time
range ofType xsd:integer

comment: relationDefinitions
relation contains
    nonFunctionalProperties
    dc:description hasValue "(Interval, X) is a tuple of the binary relation corresponding to this function iff Interval contains X or an interval"
endNonFunctionalProperties
interval ofType interval
intervalOrInstant ofType instantOrInterval

concept instantOrInterval definedBy
    forAll ?x (?xmemberOf instantOrInterval <-> ?x memberOf instant or ?x memberOf interval).

comment: axiomDefinitions
axiom invalidMonthOfYear
    nonFunctionalProperties
    dc:description hasValue "integrity constraint for valid monthOfYear"
endNonFunctionalProperties
definedBy
    constraint
    ?X memberOf monthOfYear and
(\(X < 1\) or \(X > 12\)).

**axiom invalidDayOfMonth**

**nonFunctionalProperties**

\(-\text{dc:hasValue}\text{ Integrity constraint for valid dayOfMonths}\)

**endNonFunctionalProperties**

**definedBy**

**constraint**

\(\langle X \text{ memberOf dayOfMonth and (X < 1 or X > 31).} \rangle\)

**axiom validDate**

**nonFunctionalProperties**

\(-\text{dc:hasValue}\text{ Integrity Constraints for date.}\)

\(-\text{The dayOfMonth is valid in dependency of the actual monthOfYear, in a leap year the month 2 of the Year has 29 days otherwise 28. For leap years holds the following: Every year divisible by 4 is a leap year. However, every year divisible by 100 is not a leap year. However, every year divisible by 400 is a leap year after all.}\)

\(-\text{Note: This axiomatization is still imprecise, since the country plays a role when defining a valid day of the month: E.g. 1712 was a double leap year in Sweden, i.e. February 1712 had 30 days in Sweden.}\)

\(-\text{The mathematical function symbol modulo is assumed to be defined elsewhere as that it returns the remainder after an integer division of its first argument by its second}\)

\(-\text{dc:source}\text{ <-http://www.tondering.dk/claus/cal/node3.html}>\)

**endNonFunctionalProperties**

**definedBy**

**constraint**

\(\langle X \text{ memberOf date and (}\langle X.\text{dayOfMonth} > 28 \text{ and } X.\text{monthOfYear} = 2 \text{ and not (}(\text{modulo}(X.\text{year}, 4) = 0 \text{ and not modulo}(X.\text{year}, 100) = 0) \text{ or modulo}(X.\text{year}, 400) = 0)\rangle \text{ or (}\langle X.\text{dayOfMonth} > 29 \text{ and } X.\text{monthOfYear} = 2\rangle \text{ or (}\langle X.\text{dayOfMonth} > 30 \text{ and } X.\text{monthOfYear} = 4\rangle \text{ or (}\langle X.\text{dayOfMonth} > 30 \text{ and } X.\text{monthOfYear} = 6\rangle \text{ or (}\langle X.\text{dayOfMonth} > 30 \text{ and } X.\text{monthOfYear} = 9\rangle \text{ or (}\langle X.\text{dayOfMonth} > 30 \text{ and } X.\text{monthOfYear} = 11\rangle))\rangle\rangle\).

**axiom invalidHourOfDay**

**nonFunctionalProperties**

\(-\text{dc:hasValue}\text{ integrity constraint for valid hourOfDay:}\)

**endNonFunctionalProperties**

**definedBy**

**constraint**

\(\langle X \text{ memberOf hourOfDay and (X < 0 or X >= 24).} \rangle\)

**axiom invalidMinuteOfHour**

**nonFunctionalProperties**

\(-\text{dc:hasValue}\text{ integrity constraint for valid minuteOfHour:}\)

**endNonFunctionalProperties**

**definedBy**

**constraint**

\(\langle X \text{ memberOf minuteOfHour and (X < 0 or X >= 60).} \rangle\)

**axiom invalidSecondOfMinute**

**nonFunctionalProperties**

\(-\text{dc:hasValue}\text{ integrity constraint for valid secondOfMinute:}\)

**endNonFunctionalProperties**

**definedBy**

**constraint**

\(\langle X \text{ memberOf secondOfMinute and (X < 0 or X >= 60).} \rangle\)

**axiom invalidInterval**

**definedBy**

**constraint**

\(\langle X \text{ memberOf interval and X.start >= X.end.} \rangle\)

**axiom equalityDate**

**nonFunctionalProperties**

\(-\text{dc:hasValue}\text{ computes equality of a date}\)

**endNonFunctionalProperties**

**definedBy**

\(X \iff Y \iff\)
?Y memberOf date and ?X memberOf date and
?X.dayOfMonth = ?Y.dayOfMonth and
?X.monthOfYear = ?Y.monthOfYear and

axiom beforeDate
nonFunctionalProperties
dc:description hasValue "computes if a given date X is before another date ?Y"
endNonFunctionalProperties
definedBy
?X < ?Y <-
?Y memberOf date and ?X memberOf date and
((?X.dayOfMonth = ?Y.dayOfMonth and ?X.monthOfYear = ?Y.monthOfYear and ?X.year = ?Y.year) or
(?X.monthOfYear < ?Y.monthOfYear and ?X.year = ?Y.year) or
(?X.year < ?Y.year)).

axiom afterDate
nonFunctionalProperties
dc:description hasValue "defined as inverse of beforeDate"
endNonFunctionalProperties
definedBy

axiom julianDayNumber
nonFunctionalProperties
dc:description hasValue "This Axiom describes how the correct Julian Day Number can be computed for a given Gregorian Calendar Date. Note that the Gregorian Calendar was introduced in 15.October 1582. however until 1919 this axiomatization is not unambiguous since the country should be taken into to account as 3rd parameter (e.g. Greece changed at the 9 Mar 1924 from the Julian to the Gregorian calendar). Details to the axiomatization If the month is January or February we subtract 1 from the year to get a new Year and add 12 to the month to get a new Month. (Thus, we are thinking of January and February as being the 13th and 14th month of the previous year and March is the start of the year, this simplifies the calculation considering the leap year) Within the calculation the fractional part of all results has to be dropped, here we use the function symbol floor() [It can be rewritten as predicate, however it gets less readable] A more lengthy description of this axiomatization can be found at http://quasar.as.utexas.edu/BillInfo/JulianDatesG.html"
dc:source hasValues
{<<http://quasar.as.utexas.edu/BillInfo/JulianDatesG.html>>,
<<http://members.brabant.chello.nl/~h.reints/cal/whenjul2greg.htm>>>}
endNonFunctionalProperties
definedBy
julianDayNumber[instant hasValue ?X, result hasValue ?JDN] <-
?X memberOf date and
{(?X.monthOfYear < 3 and
?Y = ?X.year -1 and
?M = ?X.monthOfYear + 12
) or
(?X.monthOfYear > 2 and
?Y = ?X.year and
M = ?X.monthOfYear
))
and
?D = ?X.dayOfMonth and
?A = floor((?Y / 100) and
?B = floor(?A / 4) and
?C = 2 - ?A + ?B and
?E = floor(365.25 * (?Y + 4716)) and
?F = floor(30.6001 * (?M + 1)) and

axiom daysBetweenDates
nonFunctionalProperties
dc:description hasValue "the difference in days between 2 dates"
endNonFunctionalProperties
definedBy
daysBetween[instant1 hasValue ?D1, instant2 hasValue ?D2, result hasValue ?D] <-
?D1 memberOf date and ?D2 memberOf date and
\[ ?X = \text{julianDayNumber}(?D1) - \text{julianDayNumber}(?D2) \].

axiom equalityTime
nonFunctionalProperties
\( \text{dc:description} \) "computes if two given times are the same"
endNonFunctionalProperties
definedBy
\[ ?X = ?Y \leftarrow
?X \text{memberOf} \text{time and } ?Y \text{memberOf} \text{time}
?X.\text{secondOfMinute} = ?Y.\text{secondOfMinute}
?X.\text{minuteOfHour} = ?Y.\text{minuteOfHour}
?X.\text{hourOfDay} = ?Y.\text{hourOfDay}. \]

axiom beforeTime
nonFunctionalProperties
dc:description "computes if a given time ?X is before another time ?Y"
endNonFunctionalProperties
definedBy
\[ ?X < ?Y \leftarrow
?X \text{memberOf} \text{time and } ?Y \text{memberOf} \text{time}
(\( ?X.\text{secondOfMinute} < ?Y.\text{secondOfMinute} \) and
\( ?X.\text{minuteOfHour} = ?Y.\text{minuteOfHour} \) and
\( ?X.\text{hourOfDay} = ?Y.\text{hourOfDay} \) or
\( ?X.\text{hourOfDay} < ?Y.\text{hourOfDay} \)). \]

axiom afterTime
nonFunctionalProperties
dc:description "defined as inverse of beforeTime"
endNonFunctionalProperties
definedBy
\[ ?X > ?Y \leftarrow ?Y < ?X. \]

axiom secondsFromMidnight
nonFunctionalProperties
dc:description "computes the amount of seconds from midnight"
endNonFunctionalProperties
definedBy
\[ \text{secondsFromMidnight}\[ \text{time}\] \text{hasValue} ?T, \text{result}\[ \text{hasValue} ?X \rightarrow
?T \text{memberOf} \text{time}
?X = ?T.\text{secondOfMinute} + (?T.\text{minuteOfHour}*60) + (?T.\text{hourOfDay}*60*60). \]

axiom secondsBetweenTimes
nonFunctionalProperties
dc:description "the difference in seconds between 2 times"
endNonFunctionalProperties
definedBy
\[ \text{secondsBetween}\[ \text{instant}1\] \text{hasValue} ?T1, \text{instant}2 \text{hasValue} ?T2, \text{result}\[ \text{hasValue} ?X \rightarrow
?T1 \text{memberOf} \text{time and } ?T2 \text{memberOf} \text{time}
?X = \text{secondsFromMidnight}(?T1) - \text{secondsFromMidnight}(?T2). \]

axiom equalityDateAndTime
nonFunctionalProperties
dc:description "computes if Date and Time are equal"
endNonFunctionalProperties
definedBy
\[ ?X = ?Y \leftarrow
?X \text{memberOf} \text{dateAndTime and } ?Y \text{memberOf} \text{dateAndTime and}
?X.\text{date} = ?Y.\text{date}
?X.\text{time} = ?Y.\text{time}. \]

axiom beforeDateAndTime
nonFunctionalProperties
dc:description "computes if a given date and time ?X is before another date and time ?Y"
endNonFunctionalProperties
definedBy
\[ ?X < ?Y \leftarrow
?X \text{memberOf} \text{dateAndTime and } ?Y \text{memberOf} \text{dateAndTime and}
(\( ?X.\text{date} = ?Y.\text{date} \) and
\( ?X.\text{time} < ?Y.\text{time} \) or
\( ?X.\text{time} < ?Y.\text{date} \)). \]

axiom afterDateAndTime
nonFunctionalProperties
dc:description "defined as inverse of beforeDateAndTime"
endNonFunctionalProperties
definedBy
\[ ?X > ?Y \leftarrow ?X \text{memberOf} \text{dateAndTime and } ?Y \text{memberOf} \text{dateAndTime and}
?Y < ?X. \]

axiom secondsBetweenDateAndTime
nonFunctionalProperties
dc:description hasValue "computes the difference in seconds between two different DateAndTime"
endNonFunctionalProperties
definedBy
  secondsBetween[instant1 hasValue ?D1, instant2 hasValue ?D2, result hasValue ?X] <-
    ?D1 memberOf dateAndTime and ?D2 memberOf dateAndTime and
    ?X = secondsFromMidnight(?D1.time) + julianDayNumber(?D1.date) * 24 * 60 * 60 -
    (secondsFromMidnight(?D2.time) + julianDayNumber(?D2.date) * 24 * 60 * 60).

axiom daysBetweenDateAndTime
nonFunctionalProperties
dc:description hasValue "the difference in days between two different DateAndTime"
endNonFunctionalProperties
definedBy
daysBetween[instant1 hasValue ?T1, instant2 hasValue ?T2, result hasValue ?X] <-
  D1 memberOf dateAndTime and D2 memberOf dateAndTime and
  ?X = daysBetween(D1.date, D2.date).

axiom intervalContainment
nonFunctionalProperties
dc:description hasValue "computes if a interval ?X contains a second interval ?Y"
endNonFunctionalProperties
definedBy
  contains(?X, ?Y) <-
    ?X memberOf interval and ?Y memberOf interval and
    (?X.start < ?Y.start or ?X.start = ?Y.start) and
    (?X.end > ?Y.end or ?X.end = ?Y.end).

axiom instantContainment
nonFunctionalProperties
dc:description hasValue "computes if a interval ?X contains a instant ?Y"
endNonFunctionalProperties
definedBy
  contains(?X, ?Y) <-
    ?X memberOf interval and ?Y memberOf instant and
    (?X.start < ?Y or ?X.start = ?Y) and
    (?X.end > ?Y or ?X.end = ?Y).

### 3.1.3 Purchase Ontology

The Purchase Ontology describes the domain of purchasing within a B2C scenario. In order to base this ontology on existing, commonly accepted conceptual models for purchasing, RosettaNet’s PIP3A4 “PurchaseOrderRequest” [RosettaNet] has been transformed into a WSMO ontology. However, RosettaNet as well as the other existing conceptual models for purchasing like ebXML [ebXML] and EDIFACT [EDIFACT] are designed for B2B purchase scenarios and thus not applicable for B2C settings as the one of this use case. Because of this, the Purchase Ontology defined in the following listing describes ontological notions relevant for purchasing within a B2C setting. We refer to the WSML representation of RosettaNet’s PIP3A4 “PurchaseOrderRequest” which is provided in listing B5, Appendix B, denoting the realted concepts within the relation attribute of the non functional properties.

The main constructs of the Purchase Ontology are:

- **Purchase**: the overall construct that holds all aspects for a purchase where a buyer buys a product from a seller
- **Purchase Order**: specifies the product, payment, and delivery terms of a purchase
- **Purchase Partners**: the parties involved in a purchase, i.e. buyer and seller
- **Product**: information about a product provided by a seller
- **Payment**: specifies notions of payment within B2C purchasing
- **Delivery**: specifies delivery methods for delivering a purchased product from the seller to the buyer
Listing 3. Domain Ontology “Purchase”

```xml
namespace <http://www.wsmo.org/ontologies/purchase#>
dc:<http://purl.org/dc/elements/1.1#>
cu:<http://www.wsmo.org/2004/d3/d3.2/v0.1/20040628/resources/owlCurrencyMediator.wsml#>
dt:<http://www.wsmo.org/ontologies/dateTime#>
loc:<http://www.wsmo.org/ontologies/location#>
prs:<http://www.wsmo.org/2004/d3/d3.3/v0.1/20041008/resources/owlPersonMediator.wsml#>
rn:<http://www.wsmo.org/ontologies/rosettanet#>
xsd:<http://www.w3.org/2001/XMLSchema#>
targetnamespace <<http://www.wsmo.org/ontologies/purchase#>>
ontology <<http://wsmo.org/ontologies/purchase/>>

nonFunctionalProperties
dc:title hasValue "B2C Purchase Ontology"
dc:creator hasValue "DERI International"
dc:description hasValue "general ontology for purchasing in B2C settings, related to 3A4 PIP of RosettaNet"
dc:publisher hasValue "DERI International"
dc:contributor hasValues {<<http://sw.deri.ie/~haller/foaf.rdf>>, "Michael Stollberg"}
dc:date hasValue "2004-10-27"
dc:type hasValue <http://www.wsmo.org/2004/d2/#ontologies> dc:format hasValue "text/html"
dc:language hasValue "en-US"
dc:relation hasValues {<<http://wsmo.org/ontologies/rosettanet#>>,
  <http://www.daml.ecs.soton.ac.uk/ont/currency.daml>>,
  <<http://www.wsmo.org/ontologies/dateTime#>>,
  <<http://www.wsmo.org/ontologies/location#>>}
dc:rights hasValue <<http://www.deri.org/privacy.html>>
version hasValue "$Revision: 1.7 $"
endNonFunctionalProperties

importedOntologies {<<http://www.wsmo.org/ontologies/dateTime.wsml>>,
  <<http://www.wsmo.org/ontologies/location.wsml>>}

usedMediators
  ooMediator {<http://www.wsmo.org/2004/d3/d3.3/v0.1/20041008/resources/owlCurrencyMediator.wsml#>,
    <http://www.wsmo.org/2004/d3/d3.3/v0.1/20041008/resources/owlPersonMediator.wsml#>}

/*
 * Purchase, Purchase Order
 */

concept purchase
  nonFunctionalProperties
  dc:relation hasValues {<http://wsmo.org/ontologies/rosettanet#pip3A4Purchase#>
endNonFunctionalProperties
  purchaseIdentifier ofType identifier
  purchaseorder ofType purchaseOrder
  buyer ofType buyer
  seller ofType seller

concept identifier subConceptOf xsd:string
  nonFunctionalProperties
  dc:relation hasValues {<http://wsmo.org/ontologies/rosettanet#proprietaryDocumentIdentifier#>
endNonFunctionalProperties

concept purchaseOrder
  nonFunctionalProperties
  dc:relation hasValues {<http://wsmo.org/ontologies/rosettanet#purchaseOrder#>
endNonFunctionalProperties
  product ofType set product
  payment ofType paymentMethod
  delivery ofType delivery
  contractDate ofType dt:date
  paymentTerms ofType paymentTerms
  totalPrice ofType financialAmount

/*
 * Purchase Partners
 */

concept buyer
  nonFunctionalProperties
  dc:relation hasValues "the partner that buys a product in a purchase"
```
concept seller
  nonFunctionalProperties
    dc:description hasValue "the partner that sells a product in a purchase"
  endNonFunctionalProperties
concept contactInformation
  nonFunctionalProperties
    dc:description hasValue "The collection of business properties that provide communication and address information for contacting a person, organization or business."
  endNonFunctionalProperties
concept product
  nonFunctionalProperties
    dc:description hasValue "The collection of business properties that describes a product."
  endNonFunctionalProperties
concept productItem
  nonFunctionalProperties
    dc:description hasValue "a placeholder for an item that is provided for sale by a seller. Any concept can be declared as a subconcept of productItem"
  endNonFunctionalProperties
concept financialAmount
  nonFunctionalProperties
    dc:description hasValue "The collection of business properties that describe the monetary amount defined by a specified currency."
  endNonFunctionalProperties
concept paymentMethod
  nonFunctionalProperties
    dc:description hasValue "superconcept of payment methods"
  endNonFunctionalProperties
concept creditCard
  subConceptOf paymentMethod
  nonFunctionalProperties
    dc:description hasValue "payment method credit card"
  endNonFunctionalProperties
  type ofType creditCardType
  creditCardNumber ofType xsd:integer
  holder ofType xsd:string
  expMonth ofType dt:monthOfYear
expYear oType dt:year

concept creditCardType
  nonFunctionalProperties
    dc:description hasValue "specifies type of credit card. This is a unique code defined by the first 4 digits of the credit card number; thus, we we would have the knowledge for decoding these digits, the credit card type could be inferred from the credit card number."
  endNonFunctionalProperties

concept invoice subConceptOf paymentMethod
  nonFunctionalProperties
    dc:description hasValue "payment method invoice"
  endNonFunctionalProperties
  invoiceNumber oType xsd:string
  payer oType buyer
  receiver oType seller
  receiverAccount oType account

concept check subConceptOf paymentMethod
  nonFunctionalProperties
    dc:description hasValue "payment method check"
  endNonFunctionalProperties
  checkNumber oType xsd:integer
  drawer oType buyer
  receiver oType seller
  drawerAccount oType account

concept cash subConceptOf paymentMethod
  nonFunctionalProperties
    dc:description hasValue "payment method cash"
  endNonFunctionalProperties
  payer oType buyer
  receiver oType seller

concept account
  nonFunctionalProperties
    dc:description hasValue "The collection of business properties that describe a bank account."
    dc:relation hasValues [{<<http://wsmo.org/ontologies/rosettanet#accountDescription>>>}
  endNonFunctionalProperties
  accountnumber oType xsd:string
  holder oType prs:person
  provider oType financialInstitute
  globalaccountclassificationcode oType xsd:string
    nonFunctionalProperties
      dc:description hasValue "Account classification indicating its functionality, e.g., credit card or a giro account."
    endNonFunctionalProperties

concept financialInstitute
  nonFunctionalProperties
    dc:description hasValue "an organization that provides financial services."
  endNonFunctionalProperties
  name oType xsd:string
  contactInformation oType contactInformation

concept bank subConceptOf financialInstitute
  nonFunctionalProperties
    dc:description hasValue "specifies a bank (financial institute)"
  endNonFunctionalProperties
  bankIdentifierCode oType xsd:string
    nonFunctionalProperties
      dc:description hasValue "the (inter)national code of the bank"
    endNonFunctionalProperties

/*
 * Payment Terms
 */

concept paymentTerms
  nonFunctionalProperties
    dc:description hasValue "The collection of business properties that describe payment terms."
    dc:relation hasValues [{<<http://wsmo.org/ontologies/rosettanet#paymentTerms>>>}
  endNonFunctionalProperties
  discounts oType discounts
  globalpaymentconditioncode oType xsd:string
nonFunctionalProperties
  dc:description hasValue "A code identifying the conditions under which payment will be made."
endNonFunctionalProperties

nettermsday ofType dt:dayOfMonth
nonFunctionalProperties
  dc:description hasValue "The specific day of the month a payment is due without incurring late charges."
endNonFunctionalProperties

nettermsdays ofType dt:interval
nonFunctionalProperties
  dc:description hasValue "The number of days within which a payment is due without incurring late charges."
endNonFunctionalProperties

percentdue ofType xsd:float
nonFunctionalProperties
  dc:description hasValue "The amount owed expressed as a percentage."
endNonFunctionalProperties

concept discounts
  nonFunctionalProperties
    dc:description hasValue "The collection of business properties that describe payment discounts."
    dc:relation hasValues {<<http://wsmo.org/ontologies/rosettanet#discounts>>}
  endNonFunctionalProperties

discountday ofType dt:dayOfMonth
  nonFunctionalProperties
    dc:description hasValue "The specific day of the month a payment is required in order to receive a discount."
  endNonFunctionalProperties

discountdays ofType dt:interval
  nonFunctionalProperties
    dc:description hasValue "The number of days within which a payment is required in order to receive a discount."
  endNonFunctionalProperties

discountpercent ofType xsd:float
  nonFunctionalProperties
    dc:description hasValue "The financial percent representing a reduction to the total amount due."
  endNonFunctionalProperties

concept delivery
  nonFunctionalProperties
    dc:description hasValue "superconcept of delivery methods. one instance of a delivery is a main function of the marketplace in addition to purchase contracting."
  endNonFunctionalProperties

deliveryItem ofType set product
sender ofType seller
receiver ofType buyer

collection dropShip subConceptOf delivery
  nonFunctionalProperties
    dc:description hasValue "delivery directly to the buyer shipping address by a delivery service"
    dc:relation hasValues {<<http://wsmo.org/ontologies/rosettanet#dropShip>>}
  endNonFunctionalProperties

carrier ofType dropShipCarrier
  nonFunctionalProperties
    dc:description hasValue "a company that provides a drop ship delivery service"
    dc:relation hasValues {<<http://wsmo.org/ontologies/rosettanet#carrierInformation>>}
  endNonFunctionalProperties

contactInformation ofType contactInformation
businessIdentifier ofType rn:globalBusinessIdentifier
transportBy ofType transportationMean
deliveryCoverage ofType loc:location

concept transportationMean
  nonFunctionalProperties
    dc:description hasValue "mean of transportation used by a drop ship delivery service"
  endNonFunctionalProperties

concept selfCollection subConceptOf delivery
  nonFunctionalProperties
    dc:description hasValue "buyer collects purchased"
  endNonFunctionalProperties
3.1.4 Location Ontology

The "Location Ontology" defines concepts for locations, including cities and states, as well as postal addresses. This ontology is based on the DAML ontology for geographical locations (Appendix B5), an ontology describing a wide variety of locations and geographical areas. The concept country is extended using the OWL-Factbook ontology (Appendix B2). The concept address reuses the DAML address ontology (Appendix B3).
Listing 4. Domain Ontology “Locations”

```xml
namespace <<http://www.wsmo.org/ontologies/location#>>
dc:<<http://purl.org/dc/elements/1.1#>>
wsml:<<http://www.wsmo.org/d2#>>
cnt:<<http://www.wsmo.org/2004/d3/d3.3/v0.1/20041008/resources/owlFactbookMediator.wsml#>>
geo:<<http://www.wsmo.org/2004/d3/d3.3/v0.1/20041008/resources/owlGeoMediator.wsml#>>
ad:<<http://www.wsmo.org/2004/d3/d3.3/v0.1/20041008/resources/owlAddressMediator.wsml#>>
xsd:<<http://www.w3.org/2001/XMLSchema#>>
ontology <<http://www.wsmo.org/ontologies/location#>>

nonFunctionalProperties
  dc:title hasValue "Locations Ontology"
  dc:creator hasValue "DERI International"
  dc:description hasValue "Ontology for representing locations in the current political/social system"
  dc:contributor hasValues {<<http://www.deri.org/foaf#lara>>, <<http://homepage.uibk.ac.at/~c703262/foaf.rdf>>}
  dc:date hasValue "2004-10-04"
  dc:type hasValue http://www.wsmo.org/2004/d2#ontologies
  dc:format hasValue "text/html"
  dc:language hasValue "en-US"
  dc:source hasValue <<http://www.daml.org/2001/02/geofile/geofile-ont#>>
  dc:coverage hasValue "ID:7029392 Name:World"
  dc:rights hasValue <<http://www.deri.org/privacy.html#>>
version hasValue "$Revision: 1.3 $"
endNonFunctionalProperties


concept country subConceptOf {cnt:country, geo:country}
  nonFunctionalProperties
    dc:description hasValue "Add the codes to the CIA country properties"
endNonFunctionalProperties
  fipsCode ofType xsd:string
  nonFunctionalProperties
    dc:description hasValue "FIPS 10-4 Country Code"
endNonFunctionalProperties
  isoCode ofType xsd:string
  nonFunctionalProperties
    dc:description hasValue "ISO 3166 Country Code"
endNonFunctionalProperties

concept address subConceptOf ad:address
  nonFunctionalProperties
    dc:description hasValue "Extended address, adding more details to city, state and country"
endNonFunctionalProperties
  city ofType city
  state ofType state
  country ofType country

concept city subConceptOf geo:city
  nonFunctionalProperties
    dc:description hasValue "City"
endNonFunctionalProperties
  population ofType xsd:integer
  extension ofType xsd:integer
  nonFunctionalProperties
    dc:description hasValue "Extension of the city in square kilometers"
endNonFunctionalProperties
  zipcodes ofType set xsd:string

concept state subConceptOf geo:state
  nonFunctionalProperties
    dc:description hasValue "State"
endNonFunctionalProperties
  cities ofType set city
  population ofType xsd:integer
  extension ofType xsd:integer
```

concept border subConceptOf geo:geographicLocation
  nonFunctionalProperties
      dc:description hasValue "Border between two countries. Notice that it
would be more natural to model this as a location with a cardinality
constraint = 2 for the country property. However, it is not clear
how to do this in F-Logic"
endNonFunctionalProperties
  countryA ofType country
  countryB ofType country

concept distance
  nonFunctionalProperties
      dc:description hasValue "Distance between two points"
endNonFunctionalProperties
  amount ofType xsd:float
  units ofType xsd:string

axiom validDistance
  nonFunctionalProperties
      dc:description hasValue "The amount in a distance cannot be less than 0.
We only accept kilometers and miles."
endNonFunctionalProperties
  definedBy
      constraint
          ?D[
              amount hasValue ?A,
              units hasValue ?U
          ]memberOf distance
          and ?A < 0
          and not (U="Kilometers" or U="Miles")

function kilometers
  nonFunctionalProperties
      dc:description hasValue "Calculates a distance in kilometers"
endNonFunctionalProperties
  d ofType distance
  range ofType xsd:float
  definedBy
      forAll ?x, ?y ( kilometers[d hasValue ?d, result hasValue ?y] equivalent
          ?d[
              amount hasValue ?a,
              units hasValue ?u
          ])
          and ((?u="Kilometers" and ?y=?a) or
          (?a="Miles" and ?y=?a*1.609344)).

function miles
  nonFunctionalProperties
      dc:description hasValue "Calculates a distance in miles"
endNonFunctionalProperties
  d ofType distance
  range ofType xsd:float
  definedBy
      forAll ?x, ?y ( miles[d hasValue ?d, result hasValue ?y] equivalent
          ?d[
              amount hasValue ?a,
              units hasValue ?u
          ])
          and ((?u="Miles" and ?y=?a) or
          (?a="Kilometers" and ?y=?a/1.609344)).

relation equalDistance
  nonFunctionalProperties
      dc:description hasValue "Computes equality of a distance"
endNonFunctionalProperties
  d1 ofType distance
  d2 ofType distance
  definedBy
      forAll ?x, ?y ( equalDistance[d1 hasValue ?x, d2 hasValue ?y] equivalent
          kilometers(?x, ?k1) and kilometers(?y, ?k2) and
          ?k1=?k2).

relation lessThanDistance
  nonFunctionalProperties
      dc:description hasValue "Computes -less- for a distance"
endNonFunctionalProperties
  d1 ofType distance
  d2 ofType distance
  definedBy...
3.1.5 VTA Use Case Knowledge Base

The VTA Use Case Knowledge Base holds all instance data that are needed within the use case descriptions. The knowledge base is defined as an WSMO ontology that holds instances of all four domain ontologies defined above. Within this knowledge base, only a selection of instances is defined that is used within the subsequent WSMO component models.
Listing 5. VTA Use Case Knowledge Base

/*
 * VTA Use Case Knowledge Base
 * this is a collection of pre-defined data for the WSMO D3.3 VTA Use Case
 */

namespace <<http://www.wsmo.org/ontologies/kb.wsml#>>
dc:<<http://purl.org/dc/elements/1.1#>>
wsml:<<http://www.wsmo.org/2004/d16/d16.1/v0.2/20040418#>>
tc:<<http://www.wsmo.org/ontologies/trainConnection#>>
dt:<<http://www.wsmo.org/ontologies/dateTime#>>
po:<<http://www.wsmo.org/ontologies/purchase#>>
cu:<<http://www.wsmo.org/2004/d3/d3.2/v0.1/20040628/resources/owlCurrencyMediator.wsml#>>
loc:<<http://www.wsmo.org/ontologies/location#>>

ontology <<http://www.wsmo.org/ontologies/kb.wsml#>>

nonFunctionalProperties
dc:title hasValue "VTA Use Case Knowledge Base"
dc:creator hasValue "SWF Project"
dc:description hasValue "describes pre-defined instance data for the VTA Use Case"
dc:publisher hasValue "SWF Project"
dc:contributor hasValue "Michael Stollberg, Holger Lausen, Ruben Lara, Axel Polleres, Armin Haller"
dc:date hasValue "2004-10-27"
dc:type hasValue <<http://www.wsmo.org/2004/d2/#ontologies>>
dc:format hasValue "text/html"
dc:language hasValue "en-US"
dc:relation hasValues {
    <<http://www.wsmo.org/ontologies/dateTime>>,
    <<http://www.wsmo.org/ontologies/trainConnection>>,
    <<http://www.wsmo.org/ontologies/purchase>>,
    <<http://www.wsmo.org/ontologies/location>>}
dc:coverage hasValue "VTA virtual marketplace"
dc:rights hasValue <<http://www.deri.org/privacy.html>>
version hasValue "$Revision: 1.2 $"
endNonFunctionalProperties

importedOntologies {
    <<http://www.wsmo.org/ontologies/dateTime>>,
    <<http://www.wsmo.org/ontologies/trainConnection>>,
    <<http://www.wsmo.org/ontologies/purchase>>,
    <<http://www.wsmo.org/ontologies/location>>}

usedMediators ooMediator <<http://www.wsmo.org/2004/d3/d3.3/v0.1/20041008/resources/owlCurrencyMediator.wsml#>>

// Station instances

instance innsbruckHbf memberOf station
nonFunctionalProperties
dc:description hasValue "Innsbruck central station"
endNonFunctionalProperties
name hasValue "Innsbruck Hbf"^^xsd:string
code hasValue "INN"^^xsd:string
locatedIn hasValues {loc:innsbruck}

instance frankfurtHbf memberOf station
name hasValue "Frankfurt Hbf"^^xsd:string
code hasValue "FKF"^^xsd:string
locatedIn hasValues {loc:frankfurt}

// Date and Time Instance

comment: the current date is defined as an instance here, as there is no build in function available for this at the moment.

instance currentDate memberOf dt:dateandtime
date hasValue currentDateDate
time hasValue currentDateTime

instance currentDateDate memberOf dt:date
dayOfMonth hasValue 22
monthOfYear hasValue 5
year hasValue 2004

instance currentDateTime memberOf dt:time
hourOfDay hasValue 15
minuteOfHour hasValue 27
secondOfMinute hasValue 38
// Locations
    instance europe memberOf loc:continent
        name hasValue "Europe"

    instance austria memberOf country
        fipsCode hasValue "AU"^^xsd:string
        isoCode hasValue "AT"^^xsd:string

    instance germany memberOf country
        fipsCode hasValue "GM"^^xsd:string
        isoCode hasValue "DE"^^xsd:string

    instance usa memberOf country
        fipsCode hasValue "US"^^xsd:string
        isoCode hasValue "US"^^xsd:string

    instance massachusetts memberOf state
        name hasValue "Massachusetts"^^xsd:string
        locatedIn hasValue usa

    instance innsbruck memberOf city
        name hasValue "Innsbruck"^^xsd:string
        locatedIn hasValue austria

    instance frankfurt memberOf city
        name hasValue "Frankfurt"^^xsd:string
        locatedIn hasValue germany

    instance boston memberOf city
        name hasValue "Boston"^^xsd:string
        locatedIn hasValue usa

    /*
     * Drop Ship Deliverers
     */
    instance PostAt http://www.post.at/ memberOf po:dropShipCarrier
        name hasValue "Oesterreichische Post AG"
        companyNumber hasValue "post"
        contactaddress hasValue PostATAddress
        transportBy hasValue truck
        deliveryCoverage hasValue europe

    instance PostATAddress memberOf loc:address
        street hasValue "Südtiroler Platz"
        number hasValue "5-7"
        city hasValue innsbruck
        zip hasValue 6020

    // transportation means
    instance truck memberOf po:transportationMean
    instance train memberOf po:transportationMean
    instance plane memberOf po:transportationMean

    /*
     * currencies - only Euro, as the marketplace is limited to Austria
     */
    instance euro memberOf cu:curreny
        name hasValue "Euro"
        code hasValue "EUR"

    /*
     * payment methods
     */
    instance masterCard memberOf po:creditCardType
    instance visa memberOf po:creditCardType
    instance americalExpress memberOf po:creditCardType
3.2. Goals

Goals denote what a user wants as the result of the Web Service. For modeling the goal, WSMO describes the information elements that the user wants to get from the service (the postcondition) together with the state of the world desired after the service execution (the effect).

In WSMO, Goals can be defined at different levels of granularity. By so-called GG Mediators, new, more specific Goals can be created out of generic existing Goals. You can also think of generic Goals as being pre-defined in a specific application context, wherefrom concrete Goals can be generated from. In order to showcase this, we define a generic Goal for buying a ticket for any kind of trip (Listing 6), a concrete Goal wherein a user wants to buy a train itinerary from Innsbruck to Frankfurt on a certain date (Listing 7), and a GG Mediator that connects the generic Goal to tickets for train trips within Austria and Germany (see Section 3.4.3 for the GG Mediator that connects this two Goals).

Listing 6 shows the generic Goal with the following description elements:

- **postcondition**: A purchase for a buyer with a ticket as the product
- **effect**: the purchased ticket is delivered to the buyer
Listing 6: Goal - buying a ticket online

```xml
namespace <<http://www.wsmo.org/2004/d3/d3.3/v0.1/20041119/resources/GeneralTrainTrip#>>
dc:<http://purl.org/dc/elements/1.1#>>
tc:<http://www.wsmo.org/ontologies/trainConnection#>>
po:<http://www.wsmo.org/ontologies/purchase#>>
loc:<http://www.wsmo.org/ontologies/location#>>
goal <<http://www.wsmo.org/2004/d3/d3.3/v0.1/20041119/resources/goal1.wsml>>

nonFunctionalProperties
dc:title hasValue "Buying a ticket online"
dc:creator hasValue "DERI International"
dc:subject hasValues {"Tickets", "Online Ticket Booking", "trip"}
dc:description hasValue "Express the goal of buying a ticket for a trip"
dc:publisher hasValue "DERI International"
dc:date hasValue "2004-10-04"
dc:type hasValue <<http://www.wsmo.org/2004/d2#goals>>
dc:format hasValue "text/html"
dc:identifier hasValue <<http://www.wsmo.org/2004/d3/d3.3/v0.1/20041119/resources/goal1.wsml>>
dc:language hasValue "en-US"
dc:coverage hasValue "ID:7029392 Name:World"
dc:rights hasValue "<http://deri.at/privacy.html>>
version hasValue "$Revision: 1.4 $"
endNonFunctionalProperties

postcondition

axiom purchasingTicketForTrip
  nonFunctionalProperties
dc:description hasValue "This goal expresses the general desire of buying a ticket for any kind of itinerary."
endNonFunctionalProperties
definedBy
  ?Purchase memberOf po:purchase
    po:purchaseorder hasValue ?Purchaseorder, po:buyer hasValue ?Buyer
  ] and
  ?Buyer memberOf po:buyer and
  ?Purchaseorder memberOf po:purchaseOrder[
    po:product hasValues {?Product},
    po:payment hasValue ?PaymentMethod
  ] and
  ?PaymentMethod memberOf po:paymentMethod and
  ?Product memberOf po:product[
    po:item hasValues {?Ticket}
  ] and
  ?Ticket memberOf tc:ticket[
    po:itinerary hasValue ?Itinerary
  ] and
  ?Itinerary memberOf tc:itinerary[
    po:passenger hasValue ?Passenger, po:trip hasValue ?Trip
  ] and
  ?Passenger memberOf tc:person and
  ?Trip memberOf tc:Trip).

effect
axiom havingTradeForTrip
  nonFunctionalProperties
dc:description hasValue "The goal effect is to get the purchased ticket delivered to the buyer."
endNonFunctionalProperties
definedBy
  ?Delivery memberOf po:delivery[
    po:deliveryItem hasValues {?Product},
    po:receiver hasValue ?Buyer
```
The concrete Goal expresses the desire to buy a train ticket by refining the notions of the general Goal. Listing 7 shows this Goal with the following elements:

- **postcondition**: A purchase for a train ticket for a train trip from Innsbruck to Frankfurt on July, 17th 2004, departure between 6 and 7 p.m.; Tim Berners-Lee is the buyer as well as the passenger for whom the ticket is valid.

- **effect**: the ticket shall be delivered to Tim Berners-Lee's shipping address by online delivery via email.
Listing 7: Goal - buying a train ticket from Innsbruck to Frankfurt online

```xml
namespace <<http://www.wsmo.org/2004/d3/d3.3/v0.1/20041119/resources/SpecificTrainTripInnsbruckFrankfurt#>>
dc:<<http://purl.org/dc/elements/1.1#>>
dt:<<http://www.wsmo.org/ontologies/dateTime#>>
tc:<<http://www.wsmo.org/ontologies/trainConnection#>>
po:<<http://www.wsmo.org/ontologies/purchase#>>
loc:<<http://www.wsmo.org/ontologies/location#>>
kb:<<http://www.wsmo.org/ontologies/kb#>>
wsml:<<http://www.wsmo.org/2004/d2/#>>
xsd:<<http://www.w3.org/2001/XMLSchema#>>
goal <<http://www.wsmo.org/2004/d3/d3.3/v0.1/20041119/resources/goal.wsml>>

nonFunctionalProperties
dc:title hasValue "Buying a train ticket from Innsbruck to Frankfurt on..."
dc:creator hasValue "DERI International"
dc:description hasValue "Express the goal of buying a concrete ticket for a train trip"
dc:publisher hasValue "DERI International"
dc:date hasValue "2004-10-04"
dc:type hasValue <<http://www.wsmo.org/2004/d2#goals>>
dc:format hasValue "text/html"
dc:identifier hasValue <<http://www.wsmo.org/2004/d3/d3.3/v0.1/20041008/resources/goal.wsml>>
dc:language hasValue "en-US"
dc:coverage hasValue "ID:7029392 Name:World"
dc:rights hasValue <http://deri.at/privacy.html>
version hasValue "$Revision: 1.4 $"
endNonFunctionalProperties

postcondition
axiom purchasingTicketForSpecificTraintrip

nonFunctionalProperties
dc:description hasValue "The goal postcondition specifies that Tim Berners-Lee wants to go buy a train ticket from Innsbruck to Frankfurt, departing from innsbruckHbf on 17th July 2004 at 6 p.m."
endNonFunctionalProperties

definedBy
()
?Purchase memberOf po:purchase[ po:purchaseorder hasValue ?Purchaseorder, po:buyer hasValue ?Buyer ] and
?Buyer memberOf po:buyer[ po:contactInformation hasValue ?TBLContactInformation, po:billToAddress hasValue ?TBLAddress, po:hasPayment hasValues {?PaymentMethod} ] and
?TBLContactInformation memberOf po:contactInformation[ po:name hasValue "Tim Berners-Lee", po:emailAddress hasValue "tbl@w3c.org" ] and
?TBLAddress memberOf po:address[ po:streetAddress hasValue "32 Vassar Street", po:city hasValue boston, po:state hasValue massachusetts, po:country hasValue usa ] and
?Purchaseorder memberOf po:purchaseOrder[
```
3.3 Web Services

We define one (imaginary) Web Service in this use case: an end-user service (means that the user interacts with this service) for purchasing international train tickets offered by the Austrian national train operator ÖBB; this Web Service can be composed of other Web Services, each for the search and buying facility of international train tickets. This setting allows modeling all notions of a WSMO Web Service description: a Capability of the end-user service and its Choreography for user-service interaction, as well as the orchestration which incorporates the aggregated Web Services. The current version of WSMO Standard does only provide a stable
specification for describing Capabilities, the model below is restricted to the overall Web Service description and the Capability definition. The modeling for the WSMO Web Service Interface will be added in a later version.

A Web Service Capability in WSMO is described by pre- and postconditions, assumptions and effects, as defined in [Roman et al., 2004]. Listing 8 shows the ÖBB Web Service description, currently the Capability only. More detailed discussion of the Discovery mechanism of WSMO Goals and Capabilities is provided in section 3.1.3. The Capability description elements are defined as follows:

- **precondition:** the input has to be the information about the buyer, a train trip that a ticket is to be purchased for, and information on the passenger for whom the ticket shall be valid. To be valid input, the following restrictions are defined for the trip: the start and end locations are restricted to stations in Austria or Germany; the departure date for the trip has to be after the current date; and the payment method of the buyer has to be a credit card.

- **assumption:** the credit card submitted as input has to be valid (not expired).

- **postcondition:** the service returns a purchase for train tickets valid in Austria and Germany by the ÖBB as the provider, with payment by credit card only.

- **effect:** the sold ticket is delivered to the buyer shipping address, either by a drop ship carrier or via online delivery.
Listing 8: ÖBB Web Service for Booking Online Train Tickets for Austria and Germany

```xml
namespace <<http://www.wsmo.org/2004/d3/d3.3/v0.1/20041119/resources/VTAService#>>
dc: <<http://purl.org/dc/elements/1.1#>>
dt: <<http://www.wsmo.org/ontologies/dateTime#>>
tc: <<http://www.wsmo.org/ontologies/trainConnection#>>
po: <<http://www.wsmo.org/ontologies/purchase#>>
loc: <<http://www.wsmo.org/ontologies/location#>>
ucase: <<http://www.wsmo.org/2004/d3/d3.3/v0.1/20041105/resources/>>
targetnamespace: <<http://www.wsmo.org/2004/d3/d3.3/v0.1/20041119/resources/ws#>>

webservice <<http://www.wsmo.org/2004/d3/d3.3/v0.1/20041119/resources/ws.wsml>>

nonFunctionalProperties
dc:title hasValue "ÖBB Online Ticket Booking Web Service"
dc:creator hasValue "DERI International"
dc:description hasValue "web service for booking online train tickets for Austria and Germany"
dc:publisher hasValue "DERI International"
dc:contributor hasValues {"Michael Stollberg", "Ruben Lara", "Holger Lausen"}
dc:date hasValue "2004-10-04"
dc:type hasValue <<http://www.wsmo.org/2004/d2/webservice>>
dc:format hasValue "text/html"
dc:language hasValue "en-us"
dc:coverage hasValues {tc:austria, tc:germany}
dc:rights hasValue <<http://deri.at/privacy.html>>
version hasValue "$Revision: 1.5 "$
endNonFunctionalProperties


capability oebbWSCapability
  precondition
    axiom oebbWSprecondition
      nonFunctionalProperties
dc:description hasValue "The oebbWSprecondition puts the following conditions on the input: it has to include a buyer with a billTo and a shipTo address, and credit card as a paymentMethod, and trip with the start- and endlocation have to be in Austria or in Germany, and the departure date has to be later than the current date."
endNonFunctionalProperties
definedBy
    ?Buyer memberOf po:buyer po:billToAddress hasValue ?BuyerBilltoAddress,
    po:shipToAddress hasValue ?BuyerShiptoAddress,
    po:hasPaymentMethod hasValues {?BuyerPaymentMethod}
    and
    ?BuyerBilltoAddress memberOf loc:address
    ?BuyerShiptoAddress memberOf loc:address
    ?BuyerPaymentMethod memberOf po:creditCard
    ?Trip memberOf tc:trainTrip
      tc:start hasValue ?Start,
      tc:end hasValue ?End,
      tc:departure hasValue ?Departure
      and
    (?Start.locatedIn = austria or ?Start.locatedIn = germany) and
    (?End.locatedIn = austria or ?End.locatedIn = germany) and
    dt:after(?Departure,?currentDate)
  ).

postcondition
  axiom oebbWSpostcondition
    nonFunctionalProperties
dc:description hasValue "the output of the service is a purchase for a ticket for train trips wherefore the start- and endlocation have to be in Austria or in Germany and the departure date has to be later than the current Date."
endNonFunctionalProperties
definedBy
    ?Purchase memberOf po:purchase
```
purchaseorder hasValue Purchaseorder, seller hasValue Seller ] and
 ?Seller memberOf po:seller [ contactInformation hasValue OEBBContactInformation, acceptsPaymentMethod hasValues [ ?PaymentMethod ] ] and
 ?OEBBContactInformation memberOf po:contactInformation [ name hasValue "Oesterreichische Bundesbahn", emailaddress hasValue "office@oebb.at", physicalAddress hasValue OEBBAddress ] and
 ?OEBBAddress memberOf loc:address [ streetAddress hasValue "Hauptfrachtenbahnhof 4", city hasValue innsbruck, country hasValue austria ] and
 ?Purchaseorder memberOf po:purchaseOrder [ product hasValues [ ?Product ], payment hasValue ?PaymentMethod ] and
 ?PaymentMethod memberOf po:creditCard and
 ?Product memberOf po:product [ item hasValues [ ?Ticket ] ] and
 ?Ticket memberOf tc:ticket [ itinerary hasValue ?Itinerary ] and
 ?Itinerary memberOf tc:itinerary [ trip hasValue ?Trip ] and
 ?Trip memberOf tc:trainTrip [ start hasValue ?Start, end hasValue ?End, 
   departure hasValue ?Departure ] and
 (?Start.locatedIn = austria or ?Start.locatedIn = germany) and
 (?End.locatedIn = austria or ?End.locatedIn = germany) and
 dt:after(?Departure, currentDate) ).

effect
 axiom oebbWSEffect

 nonFunctionalProperties
  dc:description hasValue "the sold ticket is delivered to the buyer via a drop ship carrier or via email."
 endNonFunctionalProperties
 definedBy
  ( (?Delivery memberOf po:dropShipDelivery [ deliveryItem hasValues [ ?Product ], receiver hasValue ?Buyer, sender hasValue ?Seller, 
  carrier hasValue PostAt ] ) or
  (?Delivery memberOf po:onlineDelivery [ deliveryItem hasValues [ ?Product ], receiver hasValue ?Buyer, 
  onlineDeliveryMethod hasValue "email" ] ) ) ) and
 ?Product memberOf po:product [ item hasValues [ ?Ticket ] ] and
 ?Buyer memberOf po:buyer [ shipToAddress hasValue ?BuyerShipToAddress ] and
 ?BuyerShipToAddress memberOf loc:address and
 ?Seller memberOf po:seller [ contactInformation hasValue OEBBContactInformation ] and
 ?OEBBContactInformation memberOf po:contactInformation [ name hasValue "Oesterreichische Bundesbahn", 
 emailaddress hasValue "office@oebb.at", physicalAddress hasValue OEBBAddress ] and
 ?OEBBAddress memberOf loc:address [ streetAddress hasValue "Hauptfrachtenbahnhof 4", city hasValue innsbruck, country hasValue austria ]
3.4 Mediators

3.4.1 OO Mediators

OO Mediators import other ontologies or OO Mediators into WSMO entities and resolve possible terminology mismatches. If no mismatch has to be resolved the syntactical simplification "importOntologies" can be used. In the following we specify the mediators used:

1. owlAddressMediator.wsml
2. owlCurrencyMediator.wsml
3. owlFactbookMediator.wsml
4. owlPersonMediator.wsml

Listing 9: OO-Mediator "importing the OWL Address Ontology to the Location Ontology"
Listing 10: OO-Mediator "importing the OWL Currency Ontology into the Purchase Ontology"

```
namespace
dc:<http://purl.org/dc/elements/1.1#>;
wsml:<http://www.wsmo.org/2004/d2/##>;

ooMediator <<http://www.wsmo.org/2004/d3/d3.3/v0.1/20041008/resources/owlCurrencyMediator.wsml>>
nonFunctionalProperties
  dc:title hasValue "OO Mediator importing the OWL Currency ontology to WSML"
  dc:creator hasValue "DERI International"
  dc:subject hasValues ("ooMediator", "Currency", "Purchase Order")
  dc:description hasValue "Mediator to import an OWL currency ontology into a WSML purchase order ontology"
  dc:publisher hasValue "DERI International"
  dc:creator hasValue <<http://homepage.uibk.ac.at/~c703240/foaf.rdf>>
  dc:date hasValue "2004-08-30"
  dc:type hasValue <<http://www.wsmo.org/2004/d2/#ooMediator>>
  dc:format hasValue "text/html"
  dc:identifier hasValue <<http://www.wsmo.org/2004/d3/d3.3/v0.1/20041008/resources/owlCurrencyMediator.wsml>>
  dc:language hasValue "en-us"
  dc:rights hasValue <<http://www.deri.org/privacy.html>>
  version hasValue "$Revision: 1.3 $"
endNonFunctionalProperties
source <<http://www.daml.ecs.soton.ac.uk/ont/currency.daml>>
target <<http://www.wsmo.org/ontologies/purchaseOrder>>
```

Listing 11: OO-Mediator "importing the OWL Factbook into the Location Ontology"

```
namespace
dc:<http://purl.org/dc/elements/1.1#>;
wsml:<http://www.wsmo.org/2004/d2/##>;

ooMediator <<http://www.wsmo.org/2004/d3/d3.3/v0.1/20041008/resources/owlFactbookMediator.wsml>>
nonFunctionalProperties
  dc:title hasValue "OO Mediator importing the OWL Factbook ontology to WSML"
  dc:creator hasValue "DERI International"
  dc:subject hasValues ("ooMediator", "Factbook", "Locations")
  dc:description hasValue "Mediator to import an OWL factbook ontology into a WSML locations ontology"
  dc:publisher hasValue "DERI International"
  dc:creator hasValue <<http://homepage.uibk.ac.at/~c703262/foaf.rdf>>
  dc:date hasValue "2004-08-30"
  dc:type hasValue <<http://www.wsmo.org/2004/d2/#ooMediator>>
  dc:format hasValue "text/html"
  dc:identifier hasValue <<http://www.wsmo.org/2004/d3/d3.3/v0.1/20041008/resources/owlFactbookMediator.wsml>>
  dc:language hasValue "en-us"
  dc:rights hasValue <<http://www.deri.org/privacy.html>>
  version hasValue "$Revision: 1.3 $"
endNonFunctionalProperties
source <<http://www.daml.ecs.soton.ac.uk/ont/factbook.daml>>
target <<http://www.wsmo.org/ontologies/locations>>
```
Listing 12: OO-Mediator "importing the OWL Person Ontology into the Train Connection Ontology"

namespace
dc:<<http://purl.org/dc/elements/1.1#>>
wsml:<<http://www.wsmo.org/2004/d2/#>>

oooMediator <<http://www.wsmo.org/2004/d3/d3.3/v0.1/20041008/resources/owlPersonMediator.wsml>>

nonFunctionalProperties
dc:title hasValue "OO Mediator importing the OWL Person ontology to WSML"
dc:creator hasValue "DERI International"
dc:subject hasValues ['oooMediator', 'Person', 'Train Connections']
dc:description hasValue "Mediator to import an OWL person ontology into a WSML train connections ontology"
dc:publisher hasValue "DERI International"
dc:date hasValue "2004-08-30"
dc:type hasValue <<http://www.wsmo.org/2004/d2/##ooMediator>>
dc:format hasValue "text/html"
dc:identifier hasValue <<http://www.wsmo.org/2004/d3/d3.3/v0.1/20041008/resources/owlPersonMediator.wsml>>
dc:language hasValue "en-us"
dc:relation hasValues [{'<http://daml.umbc.edu/ontologies/ittalks/person/>',
  '<http://www.wsmo.org/ontologies/trainConnection>>'}]
dc:coverage hasValue "ID:7029392 Name:World"
dc:rights hasValue <<http://www.deri.org/privacy.html>>
version hasValue "$Revision: 1.3 $"

endNonFunctionalProperties

source <<http://daml.umbc.edu/ontologies/ittalks/person/>>
target <<http://www.wsmo.org/ontologies/trainConnection/>>


Notice that the mediation service is not implemented yet. Furthermore we do not specify the capability of the mediator, since it is outside of the scope of this deliverable to define the required terminology such as an ontology about ontology languages and mediation patterns.

3.4.2 WG Mediators

A WG Mediator links a Web Service to a Goal, resolves terminological mismatches, and may state the functional difference (if any) between both. WG Mediators can be used to pre-link Services to existing Goals. For resolving terminological mismatches, OO Mediators are applied, similar to the ones specified above.

In our use case, we do not need an WG Mediator, because the Goal and the Web Service Description use the same domain ontologies (i.e. there are no terminology mismatches).

3.4.3 GG Mediators

A GG Mediator connects Goals and possible specifying the functional reduction. For example, a GG Mediator can connect a Goal "buy a ticket" with another Goal "buy a train ticket". Assuming 'train ticket' is a subclass of 'ticket', the GG Mediator would specify that the set described by the goal "buy a train ticket" is more specific then "buy a ticket".

In our use case, we have defined a generic Goal for buying a ticket for any kind of trip (Listing 6), a concrete Goal wherein a user wants to buy a train ticket from Innsbruck to Frankfurt on a certain date (Listing 7). The GG Mediator is used within the concrete Goal, so defining a connection between the general Goal as a 'template' which is 'instantiated' in the concrete Goal.
3.4.4 WW Mediators

A WW Mediator allows establishing interoperability of Web Services that are not interoperable a priori by using Web Services and resolving mismatches. The mediation facility needed for WW Mediators is to mediate between the Choreographies of Web Services that are to interact. This requires mediation on the data, protocol, and process level, i.e. all levels of mediation relevant for Semantic Web Services [Fensel & Bussler, 2002], in order to establish valid global interaction models of Web Services.

There is no WW Mediator in this use case as the corresponding notions of Choreography and Orchestration are still under development at this point in time. Future versions might include the definition of the composition of the search and buying service, wherein a WW Mediator might be applied.

4. Web Service Discovery within WSMO

Web Service Discovery is a core technology for Semantic Web Services: the aim is to detect web services that can resolve a given Goal, working on the descriptions of the goal and of web services. WSMO supports Web Service discovery by defining Goals and Web Services as top level components. Web Service discovery is a complex topic of its own, and is addressed within related documents in WSMO. Thus, we do not specify and explain realization of Web Service discovery here, but link to the related documents.

This use case has been serving as a test bed for developing the WSMO Discovery Framework that is elaborated in WSMO Deliverable 5.1 (see: http://www.wsmo.org/2004/d5/d5.1/); and it serves as the testing environment for the implementation of a WSMO discovery engine that is presented in WSMO Deliverable 5.2 (see: http://www.wsmo.org/2004/d5/d5.2/). Please refer to these documents for a complete overview of discovery within WSMO.
5. Conclusions

This document provides a real-world setting of using Semantic Web Services for a Virtual Travel Agency (VTA) that provides an end-user service for booking international train tickets, thereby aggregating Web Services of different e-Tourism Service Providers. We have modeled ontologies, goals, a Web Service, and OO Mediators according to their current specification status in WSMO [Roman et al., 2004].

This use case is the first, initial use case defined for testing and recursively developing WSMO. The major outcomes of this use case are:

- insights and test case for final specification of the WSMO top level components as defined in WSMO, D2, final version 1.0 [Roman et al., 2004]
- specification and elaboration of WSML (syntax, general structure, and logical expressions)
- test case for Web Service Discovery within WSMO.

Other use cases address different aspects of Semantic Web Services around the Web Service Modeling Ontology WSMO; WSMO Use Case are gathered in the WSMO Use Case Overview document.

References


Acknowledgements

The work is funded by the European Commission under the projects DIP, Knowledge Web, InfraWebs, SEKT, SWWS, ASG and Esperonto; by Science Foundation Ireland under the DERI-Lion project; by the Vienna city government under the CoOperate programme and by the FIT-IT (Forschung, Innovation, Technologie - Informationstechnologie) under the projects RW² and TSC.
The editors would like to thank all the members of the WSMO working group for their advice and input into this document.

Appendix A: Change Tracking

To facilitate retracing of changes between different version of this deliverable, the following lists the essential changes done in comparison to the preceding version.

The change tracking starts with the version of 28 June 2004.

- changed owl listings to abstract syntax
- corrected references and relations to other WSMO deliverables

- incorporated requested updates on use case description
- corrected listings to actual WSML syntax
- updated section 4, on discovery

- re-added the general use case overview in section 2
- reworked the use case overview: tabular overview of the resources modeled in this use case
- corrected the models of all resources:
  - International Train Connection Ontology: added concept ticket
  - Purchase Ontology: complete revision, made it 'B2C'-adequate
  - added related / used ontologies in Appendix B
  - updated models of Goals and Web Services to corrected ontologies

- separated VTA Use Case document into separated documents

- separated documents; this document only includes the concrete B2C - Virtual Travel Agency Use Case
- models / listings updated to valid WSML in accordance to WSMO D2, final version 1-0, 20 September 2004
- updated Web Service discovery part to new WSMO Web Service Discovery as defined in D5.1, 13 September 2004

- ontologies: rationales and updates, PO Ontology currently under development
- added general Goal and GG Mediator; the concrete Goal is derived from these
- updated WS Capability (assumption is now that the credit card is valid)

- complete read-thru with corrections of deliverable text (regarding comments from Jos de Bruijn)
- corrections of domain ontologies
  * changed section 3.1.1 to "Use Case Overview", describes the properties of the WSMO components modeled below
  * the web service described now is understood as an aggregated / composed web service that offers the overall functionality for purchasing train tickets online. In later versions, the Choreography description as well as the
Orchestration with specific Web Services for searching and buying train tickets can be adopted.

* corrected / clarified descriptions for modeling descriptions.
- correction of WSML-models for Goals, Web Services, Mediators
- revised the Web Service Discovery description (section 3.1.3)
- updated the FLORA2 resources to the WSML models (as in Listings)
- namespace handling refined

Appendix B: Related Resources

To ease understanding of the resources modeled within this use case, the section gathers all ontologies (re)used within this use case as well as other related resources.

B.1 Used Ontologies in OWL

The ontologies specified in Section 3.1 use existing ontologies. Most of these ontologies are modeled in OWL. To use them within WSML, the syntax has to be converted to WSML (using an OO Mediator). A mediation Service has not yet been implemented, however a partial mapping is defined in [de Bruijn, 2004].

The following listings provide the ontologies in OWL abstract syntax, since this syntax is more human readable than the RDF/XML syntax. The conversion was done using the OWL API developed by the University of Manchester. Specific Notes to the conversion can be found below each ontology (if applicable). The Listings are given to illustrate the expressive power used in this OWL ontologies and to allow the reader the comparison of between the wsml and owl ontologies.
### Listing B1: Person Ontology

```xml
Namespace rdf = <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
Namespace xsd = <http://www.w3.org/2001/XMLSchema#>
Namespace owl = <http://www.w3.org/2002/07/owl#>
Namespace a = <http://daml.umbc.edu/ontologies/ittalks/person#>
Namespace b = <http://daml.umbc.edu/ontologies/ittalks/address#>

Ontology ( <http://daml.umbc.edu/ontologies/ittalks/person>

  Annotation (owl:imports http://daml.umbc.edu/ontologies/ittalks/address)
  Annotation (owl:imports http://daml.umbc.edu/ontologies/ittalks/event)
  Annotation (owl:versionInfo "$Revision: 1.1 $")
  Annotation (rdfs:comment "A ontology that describe a person. The properties listed in this ontology is a subset of vCard. see also: http://www.ietf.org/rfc/rfc2426.txt. This ontology is created by Li Ding -- http://www.csee.umbc.edu/~dingli1/, Harry Chen -- http://www.csee.umbc.edu/~hchen4/, Lalana Kagal -- http://www.cs.umbc.edu/~lkagal1/, Tim Finin -- http://www.csee.umbc.edu/~finin/.")

ObjectProperty (a:homeAddress domain (a:Person) range (b:Address))
ObjectProperty (a:officeAddress domain (a:Person) range (b:Address))

DatatypeProperty (a:birthday domain (a:Person) range (<http://daml.umbc.edu/ontologies/ittalks/event#Instant>))
DatatypeProperty (a:cellPhone domain (a:Person) range (xsd:string))
DatatypeProperty (a:fax domain (a:Person) range (xsd:string))
DatatypeProperty (a:firstName domain (a:Person) range (xsd:string))
DatatypeProperty (a:gender domain (a:Person) range (xsd:string))
DatatypeProperty (a:homePhone domain (a:Person) range (xsd:string))
DatatypeProperty (a:email domain (a:Person) range (xsd:string))
DatatypeProperty (a:pageTitle domain (a:Person) range (xsd:string))
DatatypeProperty (a:lastname domain (a:Person) range (xsd:string))
DatatypeProperty (a:officePhone domain (a:Person) range (xsd:string))
DatatypeProperty (a:homepage domain (a:Person) range (xsd:anyURI))
DatatypeProperty (a:pageTitle domain (a:Person) range (xsd:string))
DatatypeProperty (a:officePhone domain (a:Person) range (xsd:string))
DatatypeProperty (a:homepage domain (a:Person) range (xsd:string))

Class (b:Address partial)
Class (a:Person partial
  restriction (a:firstName cardinality(1))
  restriction (a:gender maxCardinality(1))
  restriction (a:birthday maxCardinality(1))
  restriction (a:lastname maxCardinality(1)))

Class (a:Person partial
  annotation (rdfs:comment "OWL description of a person")
  annotation (rdfs:label "Person")
)

AnnotationProperty (rdfs:comment)
AnnotationProperty (rdfs:label)
AnnotationProperty (owl:versionInfo)
```

The original ontology in OWL/RDF format is available at:

[http://daml.umbc.edu/ontologies/ittalks/person](http://daml.umbc.edu/ontologies/ittalks/person)
Listing B2: OWL Factbook Ontology

Namespace(rdfs = http://www.w3.org/2001/XMLSchema#)
Namespace(rdf = http://www.w3.org/2000/01/RDF-schema#)
Namespace(xsd = http://www.w3.org/2001/XMLSchema#)
Namespace(owl = http://www.w3.org/2002/07/owl#)
Namespace(b = http://www.daml.org/2003/09/factbook/factbook-ont#)

Ontology(

Annotation(owl:imports http://www.daml.org/2001/09/countries/fips-10-4-ont)
Annotation(rdfs:comment "CIA World Fact Book (2003) Ontology")
Annotation(rdfs:comment "see http://www.cia.gov/cia/publications/factbook/")

Class(b:Country partial

restriction(a:AdministrativeDivision allValuesFrom (b:AdministrativeDivision))
restriction(a:agricultureProduct allValuesFrom (a:AgricultureProduct))
restriction(a:airportBreakdown allValuesFrom (a:AirportBreakdown))
restriction(a:airports allValuesFrom (xsd:integer))
restriction(a:background allValuesFrom (xsd:string))
restriction(a:birthRate allValuesFrom (xsd:float))
restriction(a:border allValuesFrom (a:Border))
restriction(a:BudgetCapitalExpenditures allValuesFrom (xsd:decimal))
restriction(a:BudgetExpenditures allValuesFrom (xsd:decimal))
restriction(a:BudgetRevenues allValuesFrom (xsd:decimal))
restriction(a:capital allValuesFrom (a:CapitalCity))
restriction(a:climate allValuesFrom (xsd:string))
restriction(a:coastline allValuesFrom (xsd:decimal))
restriction(a:comparativeArea allValuesFrom (xsd:string))
restriction(a:constitution allValuesFrom (xsd:string))
restriction(a:conventionalLongCountryName allValuesFrom (xsd:string))
restriction(a:countryAbbreviation allValuesFrom (xsd:string))
restriction(a:currencyCode allValuesFrom (xsd:string))
restriction(a:deathRate allValuesFrom (xsd:integer))
restriction(a:dependencyStatus allValuesFrom (xsd:string))
restriction(a:dependentArea allValuesFrom (b:Country))
restriction(a:DistributionOfFamilyIncomeGiniIndex allValuesFrom (xsd:float))
restriction(a:economicAidDonor allValuesFrom (xsd:decimal))
restriction(a:economicAidRecipient allValuesFrom (xsd:decimal))
restriction(a:economyOverview allValuesFrom (xsd:string))
restriction(a:electricityConsumption allValuesFrom (xsd:string))
restriction(a:electricityExports allValuesFrom (xsd:decimal))
restriction(a:electricityImports allValuesFrom (xsd:decimal))
restriction(a:electricityProduction allValuesFrom (xsd:decimal))
restriction(a:electricityProductionPercent allValuesFrom (xsd:float))
restriction(a:environmentalIssue allValuesFrom (xsd:string))
restriction(a:ethnicGroup allValuesFrom (a:EthnicGroupPercent))
restriction(a:exportPartner allValuesFrom (a:CountryPercent))
restriction(a:exports allValuesFrom (xsd:decimal))
restriction(a:exportsCommodity allValuesFrom (a:CommodityPercent))
restriction(a:externalDebt allValuesFrom (xsd:decimal))
restriction(a:externalMilitaryService allValuesFrom (xsd:integer))
restriction(a:femaleMilitaryAge allValuesFrom (xsd:integer))
restriction(a:femalesReachingMilitaryAgeAnnually allValuesFrom (xsd:integer))
restriction(a:fiscalYear allValuesFrom (xsd:string))
restriction(a:flagDescription allValuesFrom (xsd:string))
restriction(a:formerCountryName allValuesFrom (xsd:string))
restriction(a:geographicCoordinates allValuesFrom (a:LatLon))
restriction(a:governmentType allValuesFrom (a:GovernmentType))
restriction(a:GrossDomesticProduct allValuesFrom (xsd:decimal))
restriction(a:GrossDomesticProductComposition allValuesFrom (xsd:float))
restriction(a:GrossDomesticProductPerCapita allValuesFrom (xsd:decimal))
restriction(a:GrossDomesticProductRealGrowth allValuesFrom (xsd:float))
restriction(a:helicopters allValuesFrom (xsd:integer))
restriction(a:highestPoint allValuesFrom (a:ElevationExtreme))
restriction(a:highwaysPaved allValuesFrom (xsd:integer))
restriction(a:highwaysUnpaved allValuesFrom (xsd:integer))
restriction(a:HouseholdIncomeByPercentageShare allValuesFrom (xsd:float))
restriction(a:illicitDrugs allValuesFrom (xsd:string))
restriction(a:importPartner allValuesFrom (a:CountryPercent))
restriction(a:imports allValuesFrom (xsd:decimal))
restriction(a:importsCommodity allValuesFrom (a:CommodityPercent))
restriction(a:independence allValuesFrom (xsd:string))
restriction(a:industrialProductionGrowthRate allValuesFrom (xsd:float))
restriction(a:industry allValuesFrom (a:Industry))
restriction(a:infantMortalityRateFemale allValuesFrom (xsd:float))
restriction(a:infantMortalityRateMale allValuesFrom (xsd:float))
restriction(a:infantMortalityRateTotal allValuesFrom (xsd:float))
restriction(a:inflationRate allValuesFrom (xsd:float))
restriction(a:internationalDispute allValuesFrom (xsd:string))
restriction(a:internetCountryCode allValuesFrom (xsd:string))
restriction(a:internetServiceProviders allValuesFrom (xsd:integer))
restriction(a:internetUsers allValuesFrom (xsd:integer))
restriction(a:irrigatedLand allValuesFrom (xsd:float))
restriction(a:laborForce allValuesFrom (xsd:integer))
restriction(a:landArea allValuesFrom (xsd:decimal))
restriction(a:landBoundaries allValuesFrom (xsd:float))
restriction(a:language allValuesFrom (xsd:integer))
restriction(a:largeFlag allValuesFrom (xsd:string))
restriction(a:legalSystem allValuesFrom (xsd:string))
restriction(a:literacyDefinition allValuesFrom (xsd:string))
restriction(a:literacyFemale allValuesFrom (xsd:float))
restriction(a:literacyMale allValuesFrom (xsd:float))
restriction(a:literacyTotal allValuesFrom (xsd:float))
restriction(a:localLongCountryName allValuesFrom (xsd:string))
restriction(a:localShortCountryName allValuesFrom (xsd:string))
restriction(a:location allValuesFrom (xsd:string))
restriction(a:lowestPoint allValuesFrom (a:ElevationExtreme))
restriction(a:mainTelephoneLines allValuesFrom (xsd:integer))
restriction(a:malesFitForMilitaryService allValuesFrom (xsd:integer))
restriction(a:malesOfMilitaryAge allValuesFrom (xsd:integer))
restriction(a:malesReachingMilitaryAgeAnnually allValuesFrom (xsd:integer))
restriction(a:map allValuesFrom (xsd:anyURI))
restriction(a:mapReferences allValuesFrom (xsd:string))
restriction(a:maritimeClaim allValuesFrom (xsd:integer))
restriction(a:medianAgeFemale allValuesFrom (xsd:float))
restriction(a:medianAgeMale allValuesFrom (xsd:float))
restriction(a:medianAgeTotal allValuesFrom (xsd:float))
restriction(a:militaryAge allValuesFrom (xsd:integer))
restriction(a:militaryExpenditures allValuesFrom (xsd:decimal))
restriction(a:militaryExpendituresPercentGDP allValuesFrom (xsd:float))
restriction(a:mobileTelephoneLines allValuesFrom (xsd:integer))
restriction(a:nationalityAdjective allValuesFrom (xsd:string))
restriction(a:nationalityNoun allValuesFrom (xsd:string))
restriction(a:naturalGasProvedReserves allValuesFrom (xsd:decimal))
restriction(a:naturalHazard allValuesFrom (xsd:string))
restriction(a:naturalResource allValuesFrom (a:NaturalResource))
restriction(a:netMigrationRate allValuesFrom (xsd:float))
restriction(a:occupation allValuesFrom (a:OccupationPercent))
restriction(a:oilConsumption allValuesFrom (xsd:decimal))
restriction(a:oilExports allValuesFrom (xsd:decimal))
restriction(a:oilImports allValuesFrom (xsd:decimal))
restriction(a:oilProduction allValuesFrom (xsd:decimal))
restriction(a:oilProvedReserves allValuesFrom (xsd:decimal))
restriction(a:participatesIn allValuesFrom (a:InternationalOrganization))
restriction(a:pipelines allValuesFrom (a:PipelineDistance))
restriction(a:population allValuesFrom (xsd:integer))
restriction(a:populationBelowPovertyLine allValuesFrom (xsd:float))
restriction(a:populationGrowthRate allValuesFrom (xsd:float))
restriction(a:port allValuesFrom (a:Port))
restriction(a:radioStations allValuesFrom (a:RadioBandBreakdown))
restriction(a:radars allValuesFrom (xsd:integer))
restriction(a:railwaysBroadGauge allValuesFrom (xsd:integer))
restriction(a:railwaysDualGauge allValuesFrom (xsd:integer))
restriction(a:railwaysNarrowGauge allValuesFrom (xsd:integer))
restriction(a:railwaysStandardGauge allValuesFrom (xsd:integer))
restriction(a:railwaysTotal allValuesFrom (xsd:integer))
restriction(a:religion allValuesFrom (a:ReligionPercent))
restriction(a:sexRatio allValuesFrom (a:SexRatioBreakdown))
restriction(a:smallFlag allValuesFrom (xsd:anyURI))
restriction(a:suffrage allValuesFrom (xsd:string))
restriction(a:telephoneSystemDomestic allValuesFrom (xsd:string))
restriction(a:telephoneSystemGeneralAssessment allValuesFrom (xsd:string))
restriction(a:telephoneSystemInternational allValuesFrom (xsd:string))
restriction(a:televisionBroadcastStations allValuesFrom (xsd:integer))
restriction(a:televisions allValuesFrom (xsd:integer))
restriction(a:terrain allValuesFrom (xsd:string))
restriction(a:totalArea allValuesFrom (xsd:decimal))
restriction(a:totalFertilityRate allValuesFrom (xsd:float))
restriction(a:unemploymentRate allValuesFrom (xsd:float))
restriction(a:water allValuesFrom (xsd:decimal))
restriction(a:waterways allValuesFrom (xsd:integer))

Class(a:AdministrativeDivision partial)
Class(a:AdministrativeDivision
    annotation(rdfs:comment "intended to be subclassed for each country"))
Class(a:AgeBreakdown partial
    restriction(a:maxAge allValuesFrom (xsd:integer))
    a:Breakdown
    restriction(a:minAge allValuesFrom (xsd:integer)))
Class(a:AgricultureProduct partial
    a:SharedObject)
Class(a:AirportBreakdown partial
    restriction(a:maxRunway allValuesFrom (xsd:integer))
    a:CountBreakdown
    restriction(a:minRunway allValuesFrom (xsd:integer))
    restriction(a:paved allValuesFrom (xsd:boolean)))
Class(a:Border partial
    restriction(a:country allValuesFrom (b:Country))
    restriction(a:distance allValuesFrom (xsd:decimal)))
Class(a:Breakdown
    annotation(rdfs:comment "abstract base for classes associating categorizing a value by some other index (percentage, count, -
    Class(a:CapitalCity partial
    restriction(a:name allValuesFrom (xsd:string)))
Class(a:Commodity partial
    a:SharedObject)
Class(a:CommodityPercent partial
    a:PercentBreakdown
    restriction(a:commodity allValuesFrom (a:Commodity)))
Class(a:CountBreakdown partial
    restriction(a:count allValuesFrom (xsd:integer))
    a:Breakdown)
Class(a:CountryPercent partial
    a:PercentBreakdown
    restriction(a:country allValuesFrom (b:Country)))
Class(a:ElevationExtreme partial
    restriction(a:elevation allValuesFrom (xsd:integer))
    restriction(a:name allValuesFrom (xsd:string)))
Class(a:EnvironmentalAgreement partial
    a:SharedObject)
Class(a:Estimate partial
    rdf:Statement)
Class(a:Estimate
    annotation(rdfs:comment "rdf:type added to some (reified) Statements")
Class(a:EthnicGroup partial
    a:SharedObject)
Class(a:EthnicGroupPercent partial
    a:PercentBreakdown
    restriction(a:ethnicGroup allValuesFrom (a:EthnicGroup)))
Class(a:GovernmentType partial
    a:SharedObject)
Class(a:Industry partial
    a:SharedObject)
Class(a:InternationalOrganization partial
    a:SharedObject)
Class(a:Language partial
    a:SharedObject)
Class(a:LanguagePercent partial
    a:PercentBreakdown
    restriction(a:language allValuesFrom (a:Language)))
Class(a:LatLon partial
    restriction(a:longitude allValuesFrom (xsd:double))
    restriction(a:latitude allValuesFrom (xsd:double)))
Class(a:NaturalResource partial
    a:SharedObject)
Class(a:Occupation partial
    a:SharedObject)
Class(a:OccupationPercent partial
    a:PercentBreakdown
    restriction(a:occupation allValuesFrom (a:Occupation)))
Class(a:PercentBreakdown partial
    restriction(a:percent allValuesFrom (xsd:float))
    a:Breakdown)
Class(a:PipelineDistance partial
    restriction(a:type allValuesFrom (a:PipelineType))
    restriction(a:distance allValuesFrom (xsd:integer)))
Class(a:PipelineType partial
    a:SharedObject)
Class(a:Port partial
    restriction(a:name allValuesFrom (xsd:string)))
Class(a:RadioBand partial
    a:SharedObject)
Class(a:RadioBandBreakdown partial
    restriction(a:band allValuesFrom (a:RadioBand)))
a:CountBreakdown)
Class(a:Religion partial
a:SharedObject)
Class(a:ReligionPercent partial
a:PercentBreakdown
restriction(a:religion allValuesFrom (a:Religion)))
Class(a:SexRatioBreakdown partial
a:AgeBreakdown
restriction(a:ratio allValuesFrom (xsd:float)))
Class(a:SharedObject partial
restriction(a:name allValuesFrom (xsd:string)))
Class(a:SharedObject annotation(rdfs:comment "abstract base for common instances that are referred to by multiple Countries"))
Class(rdf:Statement partial
restriction(a:note allValuesFrom (xsd:string))
restriction(a:year allValuesFrom (xsd:gYear))
restriction(a:fiscalYear allValuesFrom (xsd:gYear)))
Class(rdf:Statement annotation(rdfs:comment "associate a parenthesized comment with a (reified) Statement")
annotation(rdfs:comment "associate a timestamp with a (reified) Statement"))
Class(xsd:string partial)
AnnotationProperty(rdfs:comment)
AnnotationProperty(owl:versionInfo)
ObjectProperty(a:administrativeDivision)
ObjectProperty(a:agricultureProduct)
ObjectProperty(a:border)
ObjectProperty(a:capital)
ObjectProperty(a:commodity)
ObjectProperty(a:country)
ObjectProperty(a:dependentArea)
ObjectProperty(a:environmentalAgreementPartyTo)
ObjectProperty(a:environmentalAgreementSigned)
ObjectProperty(a:ethnicGroup)
ObjectProperty(a:exportPartner)
ObjectProperty(a:exportsCommodity)
ObjectProperty(a:geographicCoordinates)
ObjectProperty(a:governmentType)
ObjectProperty(a:importPartner)
ObjectProperty(a:importsCommodity)
ObjectProperty(a:industry)
ObjectProperty(a:language)
ObjectProperty(a:naturalHazard)
ObjectProperty(a:naturalResource)
ObjectProperty(a:occupation)
ObjectProperty(a:participatesIn)
ObjectProperty(a:pipelines)
ObjectProperty(a:port)
ObjectProperty(a:radioStations)
ObjectProperty(a:sexRatio)
DatatypeProperty(a:airportBreakdown)
DatatypeProperty(a:airports)
DatatypeProperty(a:arableLand)
DatatypeProperty(a:background)
DatatypeProperty(a:band)
DatatypeProperty(a:birthRate)
DatatypeProperty(a:budgetCapitalExpenditures)
DatatypeProperty(a:budgetExpenditures)
DatatypeProperty(a:budgetRevenues)
DatatypeProperty(a:climate)
DatatypeProperty(a:coastline)
DatatypeProperty(a:comparativeArea)
DatatypeProperty(a:constitution)
DatatypeProperty(a:contiguousZone)
DatatypeProperty(a:continentalShelf)
DatatypeProperty(a:conventionalLongCountryName)
DatatypeProperty(a:conventionalShortCountryName)
DatatypeProperty(a:count)
DatatypeProperty(a:countryAbbreviation)
DatatypeProperty(a:currency)
DatatypeProperty(a:currencyCode)
DatatypeProperty(a:deathRate)
DatatypeProperty(a:dependencyStatus)
DatatypeProperty(a:distance)
DatatypeProperty(a:distributionOfFamilyIncomeGiniIndex)
DatatypeProperty(a:economicAidDonor)
DatatypeProperty(a:economicAidRecipient)
DatatypeProperty(a:economyOverview)
DatatypeProperty(a:militaryAge)
DatatypeProperty(a:militaryExpenditures)
DatatypeProperty(a:militaryExpendituresPercentGDP)
DatatypeProperty(a:minAge)
DatatypeProperty(a:minRunway)
DatatypeProperty(a:mobileTelephoneLines)
DatatypeProperty(a:name)
DatatypeProperty(a:nationalityAdjective)
DatatypeProperty(a:nationalityNoun)
DatatypeProperty(a:naturalGasProvedReserves)
DatatypeProperty(a:netMigrationRate)
DatatypeProperty(a:oilConsumption)
DatatypeProperty(a:oilExports)
DatatypeProperty(a:oilImports)
DatatypeProperty(a:oilProduction)
DatatypeProperty(a:oilProvedReserves)
DatatypeProperty(a:otherLandUse)
DatatypeProperty(a:paved)
DatatypeProperty(a:percent)
DatatypeProperty(a:permanentCrops)
DatatypeProperty(a:permanentPastures)
DatatypeProperty(a:population)
DatatypeProperty(a:populationBelowPovertyLine)
DatatypeProperty(a:populationGrowthRate)
DatatypeProperty(a:permanents)
DatatypeProperty(a:radio)
DatatypeProperty(a:railwaysBroadGauge)
DatatypeProperty(a:railwaysDualGauge)
DatatypeProperty(a:railwaysNarrowGauge)
DatatypeProperty(a:railwaysStandardGauge)
DatatypeProperty(a:railwaysTotal)
DatatypeProperty(a:ratio)
DatatypeProperty(a:religion)
DatatypeProperty(a:seafarer)
DatatypeProperty(a:seafarers)
DatatypeProperty(a:smallFlag)
DatatypeProperty(a:smallFlag)
DatatypeProperty(a:suffrage)
DatatypeProperty(a:telephoneSystemDomestic)
DatatypeProperty(a:telephoneSystemGeneralAssessment)
DatatypeProperty(a:telephoneSystemInternational)
DatatypeProperty(a:televisionBroadcastStations)
DatatypeProperty(a:televisionBroadcastStations)
DatatypeProperty(a:television)
DatatypeProperty(a:terrain)
DatatypeProperty(a:territorialSea)
DatatypeProperty(a:totalArea)
DatatypeProperty(a:totalFertilityRate)
DatatypeProperty(a:terrain)
DatatypeProperty(a:terrain)
DatatypeProperty(a:type)
DatatypeProperty(a:unemploymentRate)
DatatypeProperty(a:units)
DatatypeProperty(a:waterArea)
DatatypeProperty(a:waterways)
DatatypeProperty(a:year)

Individual(_ value(a:units "US dollars"))
Individual(_ value(a:units "bbl"))
Individual(_ value(a:units "bbl/day"))
Individual(_ value(a:units "births/1000 population"))
Individual(_ value(a:units "children born/woman"))
Individual(_ value(a:units "cu m"))
Individual(_ value(a:units "deaths/1000 population"))
Individual(_ value(a:units "degrees"))
Individual(_ value(a:units "index"))
Individual(_ value(a:units "kWh"))
Individual(_ value(a:units "km"))
Individual(_ value(a:units "male(s)/female"))
Individual(_ value(a:units "meters"))
Individual(_ value(a:units "migrants/1000 population"))
Individual(_ value(a:units "nautical miles"))
Individual(_ value(a:units "people"))
Individual(_ value(a:units "percent"))
Individual(_ value(a:units "percent/year"))
Individual(_ value(a:units "sq km"))
Individual(_ value(a:units "years"))

SubPropertyOf(a:arableLand a:landUse)
SubPropertyOf(a:continentalZone a:maritimeClaim)
SubPropertyOf(a:continentalShelf a:maritimeClaim)
SubPropertyOf(a:electricityProductionFossilFuel a:electricityProductionPercent)
SubPropertyOf(a:electricityProductionHydro a:electricityProductionPercent)
SubPropertyOf(a:electricityProductionNuclear a:electricityProductionPercent)
SubPropertyOf(a:electricityProductionOther a:electricityProductionPercent)
The original ontology in OWL/RDF format is available at:
http://www.w3.org/2001/XMLSchema#decimal

However those individuals have semantically no connection to the properties they occur next to within the XML representation. As shown in the abstract syntax they are just individual assertions.

The original ontology in OWL/RDF format is available at:
http://www.daml.umbc.edu/ontologies/ittalks/address
<table>
<thead>
<tr>
<th>Class</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Adminstration</td>
<td>complete restriction (b:installationTypeCode value (&quot;ADM&quot;))</td>
</tr>
<tr>
<td>AirForceStation</td>
<td>complete restriction (b:installationTypeCode value (&quot;AFS&quot;))</td>
</tr>
<tr>
<td>AirLandingArea</td>
<td>partial b:AirLandingArea</td>
</tr>
<tr>
<td>AirNationalGuardStation</td>
<td>complete restriction (b:installationTypeCode value (&quot;AGS&quot;))</td>
</tr>
<tr>
<td>AirStation</td>
<td>complete restriction (b:installationTypeCode value (&quot;ASN&quot;))</td>
</tr>
<tr>
<td>AirTerminal</td>
<td>complete restriction (b:installationTypeCode value (&quot;ATM&quot;))</td>
</tr>
<tr>
<td>Airfield</td>
<td>complete restriction (b:installationTypeCode value (&quot;AFD&quot;))</td>
</tr>
<tr>
<td>Airport</td>
<td>partial b:AirLandingArea</td>
</tr>
<tr>
<td>AmmunitionStorage</td>
<td>complete restriction (b:installationTypeCode value (&quot;AMO&quot;))</td>
</tr>
<tr>
<td>AmmunitionLandingArea</td>
<td>partial b:SupplyArea</td>
</tr>
<tr>
<td>Annex</td>
<td>complete restriction (b:installationTypeCode value (&quot;ANX&quot;))</td>
</tr>
<tr>
<td>Annex</td>
<td>partial b:Infrastructure</td>
</tr>
<tr>
<td>ArmyInstallation</td>
<td>complete restriction (b:installationTypeCode value (&quot;AIN&quot;))</td>
</tr>
<tr>
<td>ArmyInstallation</td>
<td>partial b:MilitaryInstallation</td>
</tr>
<tr>
<td>Bay</td>
<td>complete restriction (b:installationTypeCode value (&quot;BAY&quot;))</td>
</tr>
<tr>
<td>Bay</td>
<td>partial b:SeaArea</td>
</tr>
<tr>
<td>Canal</td>
<td>complete restriction (b:installationTypeCode value (&quot;CLN&quot;))</td>
</tr>
<tr>
<td>Canal</td>
<td>partial b:SeaArea</td>
</tr>
<tr>
<td>Cape</td>
<td>complete restriction (b:installationTypeCode value (&quot;CPE&quot;))</td>
</tr>
<tr>
<td>Cape</td>
<td>partial b:SeaArea</td>
</tr>
<tr>
<td>Channel</td>
<td>complete restriction (b:installationTypeCode value (&quot;CHL&quot;))</td>
</tr>
<tr>
<td>Channel</td>
<td>partial b:SeaArea</td>
</tr>
<tr>
<td>City</td>
<td>complete restriction (b:installationTypeCode value (&quot;CTY&quot;))</td>
</tr>
<tr>
<td>City</td>
<td>partial b:GeographicalArea</td>
</tr>
<tr>
<td>CivilAirPatrol</td>
<td>complete restriction (b:installationTypeCode value (&quot;CAP&quot;))</td>
</tr>
<tr>
<td>CivilAirPatrol</td>
<td>partial b:MilitaryInstallation</td>
</tr>
<tr>
<td>CoastGuardInstallation</td>
<td>complete restriction (b:installationTypeCode value (&quot;CGI&quot;))</td>
</tr>
<tr>
<td>CoastGuardInstallation</td>
<td>partial b:MilitaryInstallation</td>
</tr>
<tr>
<td>CommandOperations</td>
<td>complete restriction (b:installationTypeCode value (&quot;COC&quot;))</td>
</tr>
<tr>
<td>CommandOperations</td>
<td>partial b:MilitaryInstallation</td>
</tr>
<tr>
<td>Communication</td>
<td>complete restriction (b:installationTypeCode value (&quot;COM&quot;))</td>
</tr>
<tr>
<td>Communication</td>
<td>partial b:Infrastructure</td>
</tr>
<tr>
<td>Country</td>
<td>partial b:GeographicalArea</td>
</tr>
<tr>
<td>Clinic</td>
<td>partial b:MedicalFacility</td>
</tr>
<tr>
<td>Clinic</td>
<td>partial b:MilitaryInstallation</td>
</tr>
<tr>
<td>CoastGuardInstallation</td>
<td>complete restriction (b:installationTypeCode value (&quot;CGI&quot;))</td>
</tr>
<tr>
<td>CoastGuardInstallation</td>
<td>partial b:MilitaryInstallation</td>
</tr>
<tr>
<td>CommandOperations</td>
<td>complete restriction (b:installationTypeCode value (&quot;COC&quot;))</td>
</tr>
<tr>
<td>CommandOperations</td>
<td>partial b:MilitaryInstallation</td>
</tr>
<tr>
<td>Communication</td>
<td>complete restriction (b:installationTypeCode value (&quot;COM&quot;))</td>
</tr>
<tr>
<td>Communication</td>
<td>partial b:Infrastructure</td>
</tr>
<tr>
<td>Country</td>
<td>partial b:GeographicalArea</td>
</tr>
</tbody>
</table>

Listing B5: Geo Ontology

```
Ontology: <http://www.daml.org/2001/02/geofile/geofile-ont#>

Annotation: rdfs:comment "Ontology for the GEOFILE"

ObjectProperty: b:classification domain (b:GeographicLocation) range (b:Classification)

ObjectProperty: b:containsLocation inverseOf (b:locatedIn) domain (b:GeographicArea) range (b:Location)

ObjectProperty: b:locatedIn domain (b:Location) range (b:GeographicArea)

ObjectProperty: b:recordOwner domain (b:GeographicLocation)

ObjectProperty: b:status domain (b:GeographicLocation) range (b:Status)

DatatypeProperty: b:cancelledDate domain (b:GeographicLocation) range (xsd:date)

DatatypeProperty: b:countryStateCode domain (b:GeographicArea) range (a:cc)

DatatypeProperty: b:createdDate domain (b:GeographicLocation) range (xsd:date)

DatatypeProperty: b:geolocationCode domain (b:GeographicLocation) range (a:geocode)

DatatypeProperty: b:icao domain (b:GeographicLocation) range (a:icaoCode)

DatatypeProperty: b:installationTypeCode domain (b:GeographicLocation) range (a:installationTypeCode)

DatatypeProperty: b:lastChangedDate domain (b:GeographicLocation) range (xsd:date)

DatatypeProperty: b:locationTypeCode domain (b:Location) range (a:installationTypeCode)

DatatypeProperty: b:name domain (b:GeographicLocation) range (xsd:string)

DatatypeProperty: b:name domain (b:GeographicLocation) range (a:installationTypeCode)

DatatypeProperty: b:shortName domain (b:GeographicArea) range (xsd:string)

Class: b:Airfield complete restriction (b:installationTypeCode value ("AFD"))

Class: b:Airport complete restriction (b:installationTypeCode value ("APT"))

Class: b:AmmunitionStorage complete restriction (b:installationTypeCode value ("AMO"))

Class: b:AmmunitionLandingArea partial b:SupplyArea

Class: b:Annex complete restriction (b:installationTypeCode value ("ANX"))

Class: b:Annex partial b:Infrastructure

Class: b:ArmyInstallation complete restriction (b:installationTypeCode value ("AIN"))

Class: b:ArmyInstallation partial b:MilitaryInstallation

Class: b:Bay complete restriction (b:installationTypeCode value ("BAY"))

Class: b:Bay partial b:SeaArea

Class: b:Canal complete restriction (b:installationTypeCode value ("CNL"))

Class: b:Canal partial b:SeaArea

Class: b:Cape complete restriction (b:installationTypeCode value ("CPE"))

Class: b:Cape partial b:SeaArea

Class: b:Channel complete restriction (b:installationTypeCode value ("CHL"))

Class: b:Channel partial b:SeaArea

Class: b:City complete restriction (b:installationTypeCode value ("CTY"))

Class: b:City partial b:GeographicalArea

Class: b:CivilAirPatrol complete restriction (b:installationTypeCode value ("CAP"))

Class: b:CivilAirPatrol partial b:MilitaryInstallation

EnumeratedClass: b:Classification b:classified b:unclassified

Class: b:Clinic complete restriction (b:installationTypeCode value ("CLN"))

Class: b:Clinic partial b:MedicalFacility

Class: b:CoastGuardInstallation complete restriction (b:installationTypeCode value ("CGI"))

Class: b:CoastGuardInstallation partial b:MilitaryInstallation

Class: b:CommandOperations complete restriction (b:installationTypeCode value ("COC"))

Class: b:CommandOperations partial b:MilitaryInstallation

Class: b:Communication complete restriction (b:installationTypeCode value ("COM"))

Class: b:Communication partial b:Infrastructure

Class: b:Country partial b:GeographicalArea

Class: b:Country partial

annotation (rdfs:comment "")
```
The original ontology in OWL/RDF format is available at:
http://www.daml.org/2001/02/geofile/geofile-ont
Note that this ontology is in OWL Full, in order to show it in the more concise abstract syntax and to be able to reason with it using Description Logic Reasoners we converted it to OWL DL by applying the following fixes:

- rdfs:class statements have been transformed to owl:class statements
- rdf:property has been transformed to owl:DatatypeProperty respectively owl:ObjectProperty
- unknown data types have been explicitly declared

The rdf/xml version of the ontology with those fixes is available at:
http://www.wsmo.org/2004/d3/d3.3/v0.1/20041119/resources/ontology/geofile.owl
Listing B6: OWL - Travel Itinerary Ontology

```xml
Namespace rdf = http://www.w3.org/1999/02/22-rdf-syntax-ns#
Namespace xsd = http://www.w3.org/2001/XMLSchema#
Namespace rdfs = http://www.w3.org/2000/01/rdf-schema#
Namespace owl = http://www.w3.org/2002/07/owl#
Namespace a = http://www.daml.org/2001/06/itinerary/itinerary-ont#
Namespace b = http://www.daml.org/2001/06/itinerary/icao#
Namespace c = http://www.daml.ri.cmu.edu/ont/AirportCodes.daml#

Ontology

Annotation(rdfs:comment "Travel Itinerary")

Class(b:Airline partial)
EnumeratedClass(a: Class a:Business a:Coach a:First)
Class(a:Flight partial)
  restriction(a:aircraft allValuesFrom (a:Aircraft))
  restriction(a:airline allValuesFrom (b:Airline))
  restriction(a:airline cardinality(1))
  restriction(a:arrive allValuesFrom (xsd:dateTime))
  restriction(a:arrive cardinality(1))
  restriction(a:class allValuesFrom (a:Class))
  restriction(a:class cardinality(1))
  restriction(a:depart allValuesFrom (xsd:dateTime))
  restriction(a:depart cardinality(1))
  restriction(a:destination allValuesFrom (c:AirportCode))
  restriction(a:destination cardinality(1))
  restriction(a:duration allValuesFrom (xsd:timeDuration))
  restriction(a:duration cardinality(1))
  restriction(a:flight allValuesFrom (xsd:nonNegativeInteger))
  restriction(a:flight cardinality(1))
  restriction(a:meal allValuesFrom (a:Meal))
  restriction(a:miles allValuesFrom (xsd:integer))
  restriction(a:miles cardinality(1))
  restriction(a:origin allValuesFrom (c:AirportCode))
  restriction(a:origin cardinality(1))
  restriction(a:seat allValuesFrom (xsd:string))
Class(a:HotelReservation partial)
  restriction(a:address allValuesFrom (xsd:string)))
  restriction(a:address cardinality(1))
  restriction(a:checkin allValuesFrom (xsd:date))
  restriction(a:checkin cardinality(1))
  restriction(a:checkout allValuesFrom (xsd:date))
  restriction(a:checkout cardinality(1))
  restriction(a:confirmation allValuesFrom (xsd:string))
  restriction(a:confirmation cardinality(1))
  restriction(a:hotelName allValuesFrom (xsd:string))
  restriction(a:hotelName cardinality(1))
  restriction(a:rate allValuesFrom (xsd:decimal))
  restriction(a:rate cardinality(1))
  restriction(a:smoking allValuesFrom (xsd:boolean))
  restriction(a:smoking maxCardinality(1))
Class(a:Itinerary partial)
  restriction(a:flight allValuesFrom (a:Flight))
  restriction(a:hotel allValuesFrom (a:HotelReservation)))
  restriction(a:passenger allValuesFrom (xsd:string))
  restriction(a:passenger cardinality(1))
  restriction(a:rentalCar allValuesFrom (a:RentalCar))
  restriction(a:rln allValuesFrom (a:RecordLocatorNumber))
EnumeratedClass(a:Meal a:Lunch a:Breakfast a:Snack a:Dinner)
Class(a:RecordLocatorNumber partial)
  restriction(a:airline allValuesFrom (b:Airline))
  restriction(a:airline cardinality(1))
  restriction(a:rloc allValuesFrom (xsd:string))
  restriction(a:rloc cardinality(1))
Class(a:RentalCar partial)
Class(c:AirportCode partial)
Class(xsd:nonNegativeInteger partial)

AnnotationProperty(rdfs:comment)
AnnotationProperty(owl:versionInfo)
ObjectProperty(a:aircraft)
ObjectProperty(a:airline)
ObjectProperty(a:class)
ObjectProperty(a:destination)
```
B.2 Related Ontologies

Within the elaboration of the use case, the RosettaNet's PIP3A4 "PurchaseOrderRequest" has been transformed into a WSMO ontology. As the RosettaNet model is designed for B2B purchase orders, it is not applicable for the B2C setting of the use case demonstrated in this document. However, the Purchase Ontology of this use case as defined in Listing 3, Section 3.1, is based and heavily related to the RosettaNet Purchase Model. The listing below therefore provides the RosettaNet Purchase Model modeled in WSML.

The ontology is an WSML representation of the RosettaNet's PIP3A4 "PurchaseOrderRequest" [RosettaNet]. RosettaNet is a consortium of major Information Technology, Electronic Components, Semiconductor Manufacturing, Telecommunications and Logistics companies working to create and implement industry-wide, open e-business process standards. These standards form a common e-business language, aligning processes between supply chain partners on a global basis.

Every standard business transaction within the RosettaNet trading network is defined in a so called PIP (Partner Interface Process) which defines the XML code, activities, decisions and Partner Role interactions between two partners in the supply chain. Each partner participating in the "Partner Interface Process" must fulfill the obligations specified in a PIP. These PIPs are organized into seven clusters, or groups of core business processes, that represent the backbone of the trading network. Each cluster is broken down into segments which are cross-enterprise processes involving more than one type of trading partner. Within each segment are individual PIPs, whereas the above mentioned PIP3A4 is part of Segment 3A.
"Quote and Order Entry". This segment allows partners to exchange price and availability information, quotes, purchase orders and order status, and enables partners to send requested orders, or shopping carts, to other partners.

At the current state this domain ontology is preliminary and will be further enhanced in future versions. As far as RosettaNet's PIPs are only intended for the use in the above mentioned industry sectors we also consider and partly work on the ontologizing of other conceptualizations, inter alia ebXML [ebXML] and EDIFACT [EDIFACT].
namespace <<http://www.wsmo.org/ontologies/rosettanet#>>
dc:: <<http://purl.org/dc/elements/1.1#>>
cu: <<http://www.wsmo.org/2004/d3/d3.2/v0.1/20040628/resources/owlCurrencyMediator.wsml#>>
dt: <<http://www.wsmo.org/ontologies/dateTime#>>
targetnamespace: <<http://www.wsmo.org/ontologies/rosettanet#>>

ontology <<http://wsmo.org/ontologies/rosettanet/>>

nonFunctionalProperties
dc:title hasValue "Purchase Ontology"
dc:creator hasValue "DERI International"
dc:description hasValue "general purchase order request ontology based on the 3A4 PIP of RosettaNet"
dc:publisher hasValue "DERI International"
dc:contributor hasValues [<<http://sw.deri.ie/~haller/foaf.rdf>>]
dc:date hasValue "2004-10-04"
dc:type hasValue <<http://www.wsmo.org/2004/d2/#ontologies>>
dc:format hasValue "text/html"
dc:language hasValue "en-US"
dc:rights hasValue <<http://www.deri.org/privacy.html>>
version hasValue "$Revision: 1.1 $"
endNonFunctionalProperties

importedOntologies
<<http://www.wsmo.org/ontologies/datetime>>

usedMediators
ooMediator <<http://www.wsmo.org/2004/d3/d3.3/v0.1/20041008/resources/owlCurrencyMediator.wsml>>

comment: conceptDefinitions
concept pip3A4Purchase
  nonFunctionalProperties
  dc:description hasValue "Rosettanets Purchase Order Request Document"
endNonFunctionalProperties
  buyer {1} ofType buyer
globaldocumentfunctioncode ofType globalDocumentFunctionCode
purchaseorder {1} ofType purchaseOrder
thisdocumentgenerationdatetime {1} ofType thisDocumentGenerationDateTime
thisdocumentidentifier {1} ofType thisDocumentIdentifier
  seller {1} ofType seller
concept buyer
  nonFunctionalProperties
  dc:description hasValue "The role initiating a business document exchange."
endNonFunctionalProperties
  partnerroledescription {1} ofType partnerRoleDescription
concept seller
  nonFunctionalProperties
  dc:description hasValue "The role receiving the document in a business document exchange."
endNonFunctionalProperties
  partnerroledescription {1} ofType partnerRoleDescription
concept partnerRoleDescription
  nonFunctionalProperties
  dc:description hasValue "The collection of business properties that describe a business partners role in the purchase order request."
endNonFunctionalProperties
  contactinformation ofType contactInformation
globalpartnerroleclassificationcode {1} ofType globalPartnerRoleClassificationCode
  partnerdescription ofType partnerDescription
concept contactInformation
  nonFunctionalProperties
  dc:description hasValue "The collection of business properties that provide communication and address information for contacting a person, organization or business."
endNonFunctionalProperties
  contactname ofType contactName
emailaddress ofType emailAddress
  facsimilenumber ofType facsimileNumber
  telephonenumber ofType telephoneNumber
  physicallocation ofType physicalLocation
concept contactName
  nonFunctionalProperties
<table>
<thead>
<tr>
<th>Concept</th>
<th>SubConceptOf</th>
<th>ValueType</th>
<th>NonFunctionalProperties</th>
</tr>
</thead>
<tbody>
<tr>
<td>dc:domain</td>
<td>freeFormText</td>
<td>freeFormText</td>
<td>hasValue &quot;Name of the contact person(s) within the organization.&quot;</td>
</tr>
<tr>
<td>nonFunctionalProperties</td>
<td>freeFormText</td>
<td>freeFormText</td>
<td></td>
</tr>
<tr>
<td>concept</td>
<td>emailAddress</td>
<td>xsd:string</td>
<td>nonFunctionalProperties</td>
</tr>
<tr>
<td>nonFunctionalProperties</td>
<td>dc:description</td>
<td>hasValue &quot;Electronic mail address.&quot;</td>
<td></td>
</tr>
<tr>
<td>concept</td>
<td>facsimileNumber</td>
<td>xsd:string</td>
<td>nonFunctionalProperties</td>
</tr>
<tr>
<td>nonFunctionalProperties</td>
<td>dc:description</td>
<td>hasValue &quot;The numerical schema designed to achieve contact via facsimile.&quot;</td>
<td></td>
</tr>
<tr>
<td>concept</td>
<td>communicationsNumber</td>
<td>xsd:string</td>
<td>nonFunctionalProperties</td>
</tr>
<tr>
<td>nonFunctionalProperties</td>
<td>dc:description</td>
<td>hasValue &quot;The electro-technical communication number, e.g., telephone number, facsimile number, pager number.&quot;</td>
<td></td>
</tr>
<tr>
<td>concept</td>
<td>telephoneNumber</td>
<td>xsd:string</td>
<td>nonFunctionalProperties</td>
</tr>
<tr>
<td>nonFunctionalProperties</td>
<td>dc:description</td>
<td>hasValue &quot;The numerical schema designed to achieve contact via telephone.&quot;</td>
<td></td>
</tr>
<tr>
<td>concept</td>
<td>globalPartnerRoleClassificationCode</td>
<td>xsd:string</td>
<td>nonFunctionalProperties</td>
</tr>
<tr>
<td>nonFunctionalProperties</td>
<td>dc:description</td>
<td>hasValue &quot;Code identifying a partys role in the supply chain.&quot;</td>
<td></td>
</tr>
<tr>
<td>concept</td>
<td>partnerDescription</td>
<td>xsd:string</td>
<td>nonFunctionalProperties</td>
</tr>
<tr>
<td>nonFunctionalProperties</td>
<td>dc:description</td>
<td>hasValue &quot;The collection of business properties that describe a business partners identity, their contact information, where they are physically located and their function in a supply chain.&quot;</td>
<td></td>
</tr>
<tr>
<td>nonFunctionalProperties</td>
<td>businessDescription</td>
<td>xsd:string</td>
<td>nonFunctionalProperties</td>
</tr>
<tr>
<td>nonFunctionalProperties</td>
<td>globalPartnerClassificationCode</td>
<td>xsd:string</td>
<td>nonFunctionalProperties</td>
</tr>
<tr>
<td>nonFunctionalProperties</td>
<td>dc:description</td>
<td>hasValue &quot;Code identifying a partners function in the supply chain.&quot;</td>
<td></td>
</tr>
<tr>
<td>concept</td>
<td>globalDocumentFunctionCode</td>
<td>xsd:string</td>
<td>nonFunctionalProperties</td>
</tr>
<tr>
<td>nonFunctionalProperties</td>
<td>dc:description</td>
<td>hasValue &quot;Code identifying the function of a document as either a request or a response.&quot;</td>
<td></td>
</tr>
</tbody>
</table>

**Note:** The above text is a partial representation of the document content, focusing on specific concepts and their properties. The complete document will include additional concepts and properties not listed here.
concept purchaseOrder
  nonFunctionalProperties
    dc: description hasValue "The collection of business properties that describe a buyers offer to purchase a quantity of products at an agreed price and schedule."
  endNonFunctionalProperties
  accountDescription ofType accountDescription
  comments ofType comments
  contractInformation ofType contractInformation
  documentReference ofType documentReference
  financingTerms ofType financingTerms
  generalServicesAdministrationNumber ofType generalServicesAdministrationNumber
  globalgovernmentpriorityratingcode ofType globalGovernmentPriorityRatingCode
  globalpurchaseorderfillprioritycode ofType globalPurchaseOrderFillPriorityCode
  globalpurchaseordertypecode ofType globalPurchaseOrderTypeCode
  governmentcontractidentifier ofType governmentContractIdentifier
  installAt ofType installAt
  isDropShip ofType isDropShip
  ordershippinginformation ofType orderShippingInformation
  productLineItem ofType productLineItem
  proprietaryinformation ofType proprietaryInformation
  requestedevent ofType requestedEvent
  requestedshipfrom ofType requestedShipFrom
  secondarybuyer ofType secondaryBuyer
  shipTo ofType shipTo
  taxexemptstatus ofType taxExemptStatus
  totalamount ofType totalAmount

concept accountDescription
  nonFunctionalProperties
    dc: description hasValue "The collection of business properties that describe a customer or supplier account."
  endNonFunctionalProperties
  accountName ofType accountName
  accountnumber ofType accountNumber
  billTo ofType billTo
  creditcard ofType creditCard
  financedby ofType financedBy
  globalaccountclassificationcode ofType globalAccountClassificationCode
  prepaymentchecknumber ofType prePaymentCheckNumber
  wiretransferidentifier ofType wireTransferIdentifier

concept po: comments
  nonFunctionalProperties
    dc: description hasValue "Free form textual description of a general nature."
  endNonFunctionalProperties
  freeformtext ofType freeFormText

concept accountName
  nonFunctionalProperties
    dc: description hasValue "The name of a bank account."
  endNonFunctionalProperties
  freeformtext ofType freeFormText

concept accountNumber ofType xsd: string
  nonFunctionalProperties
    dc: description hasValue "Identification number of an account."
  endNonFunctionalProperties

concept billTo
  nonFunctionalProperties
    dc: description hasValue "The party that will pay the invoice."
  endNonFunctionalProperties
  partnerdescription ofType partnerDescription

concept businessName
  nonFunctionalProperties
    dc: description hasValue "The name of a business entity."
  endNonFunctionalProperties
  freeformtext ofType freeFormText

concept partnerBusinessIdentification
  nonFunctionalProperties
    dc: description hasValue "The collection of business properties that allow for the proprietary identification of a business entity."
  endNonFunctionalProperties
  proprietarybusinessidentifier ofType proprietaryBusinessIdentifier
  proprietarydomainidentifier ofType proprietaryDomainIdentifier
proprietaryidentifierauthority ofType proprietaryIdentifierAuthority

concept proprietaryBusinessIdentifier subConceptOf xsd:string
  nonFunctionalProperties
dc:description hasValue "A unique business identifier assigned and administered by a private authority."
endNonFunctionalProperties

concept proprietaryDomainIdentifier subConceptOf xsd:string
  nonFunctionalProperties
dc:description hasValue "A descriptor that is used to categorize an organization or business entity that is in the Proprietary Business Identifier."
endNonFunctionalProperties

concept proprietaryIdentifierAuthority subConceptOf xsd:string
  nonFunctionalProperties
dc:description hasValue "A unique name that identifies an organization or business entity that is responsible for managing one or more lists of identifiers."
endNonFunctionalProperties

concept physicalLocation
  nonFunctionalProperties
dc:description hasValue "The collection of business properties that identify and describe the actual physical location of an entity as prescribed by local postal authorities, including country identification."
endNonFunctionalProperties

  globallocationidentifier ofType globalLocationIdentifier
  partnerlocationidentification {1 n} ofType partnerLocationIdentification
  physicaladdress ofType physicalAddress

concept globalLocationIdentifier subConceptOf xsd:string
  nonFunctionalProperties
dc:description hasValue "Location uniquely identified by the DUNS +4 number."
endNonFunctionalProperties

concept partnerLocationIdentification
  nonFunctionalProperties
dc:description hasValue "The collection of business properties that allow for the proprietary identification of a business location."
endNonFunctionalProperties

  proprietarydomainidentifier {1} ofType proprietaryDomainIdentifier
  proprietaryidentifierAuthority ofType proprietaryIdentifierAuthority
  proprietarylocationidentifier {1} ofType proprietaryLocationIdentifier

concept proprietaryLocationIdentifier subConceptOf xsd:string
  nonFunctionalProperties
dc:description hasValue "A unique location identifier assigned and administered by a private authority."
endNonFunctionalProperties

concept physicalAddress
  nonFunctionalProperties
dc:description hasValue "The actual physical location of an entity as prescribed by local postal authorities, including country identification as it relates to the party or a product."
endNonFunctionalProperties

  addressline1 ofType addressLine1
  addressline2 ofType addressLine2
  addressline3 ofType addressLine3
  cityname ofType cityName
  globalcountrycode ofType globalCountryCode
  nationalpostalcode ofType nationalPostalCode
  postofficeboxidentifier ofType postOfficeBoxIdentifier
  regionname ofType regionName
  globallocationidentifier ofType globalLocationIdentifier
  partnerlocationidentification {1 n} ofType partnerLocationIdentification

concept addressLine1
  nonFunctionalProperties
dc:description hasValue "Line 1 of the physical address."
endNonFunctionalProperties

  freeformtext {1} ofType freeFormText

concept addressLine2
  nonFunctionalProperties
dc:description hasValue "Line 2 of the physical address."
endNonFunctionalProperties

  freeformtext {1} ofType freeFormText
concept addressLine3
   nonFunctionalProperties
      dc:description hasValue "Line 3 of the physical address."
   endNonFunctionalProperties
   freeformtext {1} ofType freeFormText

concept cityName
   nonFunctionalProperties
      dc:description hasValue "The name of a city."
   endNonFunctionalProperties
   freeformtext {1} ofType freeFormText

concept globalCountryCode subConceptOf xsd:string
   nonFunctionalProperties
      dc:description hasValue "Code identifying the two character country code specified in ISO 3166-1993."
   endNonFunctionalProperties

concept nationalPostalCode subConceptOf xsd:string
   nonFunctionalProperties
      dc:description hasValue "Code identifying geographic location as specified by a national postal code."
   endNonFunctionalProperties

concept postOfficeBoxIdentifier
   nonFunctionalProperties
      dc:description hasValue "The proprietary identity of a physical address, located at a post office, designed solely to accept and receive mail."
   endNonFunctionalProperties
   freeformtext {1} ofType freeFormText

concept regionName
   nonFunctionalProperties
      dc:description hasValue "The name of a state or province within a country."
   endNonFunctionalProperties
   freeformtext {1} ofType freeFormText

concept creditCard
   nonFunctionalProperties
      dc:description hasValue "A collection of business properties that describe information about a credit card."
   endNonFunctionalProperties
   cardholdername {1} ofType cardHolderName
   creditcardidentifier {1} ofType creditCardIdentifier
   expirydate {1} ofType expiryDate
   globalcreditcardclassificationcode {1} ofType globalCreditCardClassificationCode
   proprietarycididentifier ofType proprietaryCIDIdentifier

concept cardHolderName
   nonFunctionalProperties
      dc:description hasValue "The name of the owner of a credit card."
   endNonFunctionalProperties
   freeformtext {1} ofType freeFormText

concept creditCardIdentifier
   nonFunctionalProperties
      dc:description hasValue "The unique number that identifies a credit card."
   endNonFunctionalProperties
   proprietaryreferenceidentifier {1} ofType proprietaryReferenceIdentifier

concept proprietaryReferenceIdentifier subConceptOf xsd:string
   nonFunctionalProperties
      dc:description hasValue "A unique reference identifier for goods, services or business documents."
   endNonFunctionalProperties

concept expiryDate
   nonFunctionalProperties
      dc:description hasValue "The date that a contractual agreement expires."
   endNonFunctionalProperties
   expMonth {1} ofType dt:monthOfYear
   expYear {1} ofType dt:year

concept globalCreditCardClassificationCode subConceptOf xsd:string
   nonFunctionalProperties
      dc:description hasValue "Code identifying the type of credit card used."
   endNonFunctionalProperties
concept proprietaryCIDIdentifier subConceptOf xsd:string
  nonFunctionalProperties
dc:description hasValue "Unique identifier for credit card purchase activity utilized by American Express."
endNonFunctionalProperties

concept financedBy
  nonFunctionalProperties
dc:description hasValue "The party who is the financier."
endNonFunctionalProperties
partnerdescription ofType partnerDescription

concept globalAccountClassificationCode subConceptOf xsd:string
  nonFunctionalProperties
dc:description hasValue "Account classification indicating its functionality, e.g., credit card account, debit card account."
endNonFunctionalProperties

concept prePaymentCheckNumber
  nonFunctionalProperties
dc:description hasValue "The check number issued to prepay a monetary amount for an account."
endNonFunctionalProperties
checkNumber {1} ofType checkNumber

concept checkNumber subConceptOf xsd:string
  nonFunctionalProperties
dc:description hasValue "The identification code of a bank cheque."
endNonFunctionalProperties

concept wireTransferIdentifier subConceptOf xsd:string
  nonFunctionalProperties
dc:description hasValue "A unique identity of a wire transfer used for reference."
endNonFunctionalProperties

concept contractInformation
  nonFunctionalProperties
dc:description hasValue "The collection of business properties that represent a business arrangement for the supply of goods or services at an agreed price."
endNonFunctionalProperties
contractidentifier {1} ofType contractIdentifier
primarycontractwith ofType primaryContractWith
secondarycontractwith ofType secondaryContractWith

concept contractIdentifier
  nonFunctionalProperties
dc:description hasValue "The unique number than identifies a contract."
endNonFunctionalProperties
proprietarydocumentidentifier {1} ofType proprietaryDocumentIdentifier

concept proprietaryDocumentIdentifier subConceptOf xsd:string
  nonFunctionalProperties
dc:description hasValue "Unique identifier, i.e. a numeric value or alphanumeric value, for a business document."
endNonFunctionalProperties

concept primaryContractWith
  nonFunctionalProperties
dc:description hasValue "The principal party in a binding agreement between two or more persons or parties."
endNonFunctionalProperties
partnerdescription {1} ofType partnerDescription

concept secondaryContractWith
  nonFunctionalProperties
dc:description hasValue "The subsequent party in a binding agreement between two or more persons or parties."
endNonFunctionalProperties
partnerdescription {1} ofType partnerDescription

concept documentReference
  nonFunctionalProperties
dc:description hasValue "The collection of business properties that allows the description of multiple proprietary documents and applicable line number references."
endNonFunctionalProperties
datetimestamp ofType dt:dateAndTime
globaldocumentreferencetypecode {1} ofType globalDocumentReferenceTypeCode
globalpartnerroleclassificationcode {1} ofType globalPartnerRoleClassificationCode
concept globalDocumentReferenceTypeCode subConceptOf xsd:string
  nonFunctionalProperties
dc:description hasValue "Code identifying the type of business document used for referencing within another business document."
endNonFunctionalProperties

concept lineNumber subConceptOf xsd:string
  nonFunctionalProperties
dc:description hasValue "Number of the line in the document."
endNonFunctionalProperties

concept revisionNumber subConceptOf xsd:string
  nonFunctionalProperties
dc:description hasValue "An incremental number used to identify changes."
endNonFunctionalProperties

concept financingTerms
  nonFunctionalProperties
dc:description hasValue "The collection of business properties that describe financing terms."
endNonFunctionalProperties
globalFinanceTermsCode ofType globalFinanceTermsCode
paymentTerms { 1 n } ofType paymentTerms

concept globalFinanceTermsCode subConceptOf xsd:string
  nonFunctionalProperties
dc:description hasValue "Code identifying the terms that govern financing."
endNonFunctionalProperties

concept paymentTerms
  nonFunctionalProperties
dc:description hasValue "The collection of business properties that describe payment terms."
endNonFunctionalProperties
discounts { 1 n } ofType discounts
globalPaymentConditionCode ofType globalPaymentConditionCode
netTermsDay ofType netTermsDay
netTermsDays ofType netTermsDays
percentDue ofType percentDue

concept discounts
  nonFunctionalProperties
dc:description hasValue "The collection of business properties that describe payment discounts."
endNonFunctionalProperties
discountDay ofType discountDay
discountDays ofType discountDays
discountPercent ofType discountPercent

concept discountDay
  nonFunctionalProperties
dc:description hasValue "The specific day of the month a payment is required in order to receive a discount."
endNonFunctionalProperties
dayOfMonth { 1 } ofType dt:dayOfMonth

concept discountDays
  nonFunctionalProperties
dc:description hasValue "The number of days within which a payment is required in order to receive a discount."
endNonFunctionalProperties
countableAmount { 1 } ofType countableAmount

concept countableAmount subConceptOf xsd:integer
  nonFunctionalProperties
dc:description hasValue "Dimensionless magnitude, e.g. number of products."
endNonFunctionalProperties

concept discountPercent
  nonFunctionalProperties
dc:description hasValue "The financial percent representing a reduction to the total amount due."
endNonFunctionalProperties
percentAmount { 1 } ofType percentAmount

concept percentAmount subConceptOf xsd:float
  nonFunctionalProperties
dc:description hasValue "A real number representing a percentage value, e.g. 75.125 represents 75 1/8 percent."
endNonFunctionalProperties

concept globalPaymentConditionCode subConceptOf xsd:string
  nonFunctionalProperties
dc:description hasValue "A code identifying the conditions under which payment will be made."
endNonFunctionalProperties

concept netTermsDay
  nonFunctionalProperties
dc:description hasValue "The specific day of the month a payment is due without incurring late charges."
endNonFunctionalProperties
dayOfMonth {1} ofType dt:dayOfMonth

concept netTermsDays
  nonFunctionalProperties
dc:description hasValue "The number of days within which a payment is due without incurring late charges."
endNonFunctionalProperties
countableAmount {1} ofType countableAmount

concept percentDue
  nonFunctionalProperties
dc:description hasValue "The amount owed expressed as a percentage."
endNonFunctionalProperties

concept generalServicesAdministrationNumber
  nonFunctionalProperties
dc:description hasValue "Identifying number relating to a pre-established end-user pricing agreement."
endNonFunctionalProperties

concept globalGovernmentPriorityRatingCode subConceptOf xsd:string
  nonFunctionalProperties
dc:description hasValue "If a contract number exists and PO type is Government, a priority rating code is required."
endNonFunctionalProperties

concept globalPurchaseOrderFillPriorityCode subConceptOf xsd:string
  nonFunctionalProperties
dc:description hasValue "Code identifying fill priority for manufacturing in a constrained condition. Value to be determined by Trading Partner Agreement."
endNonFunctionalProperties

concept globalPurchaseOrderTypeCode subConceptOf xsd:string
  nonFunctionalProperties
dc:description hasValue "Code identifying category specification for a purchase order."
endNonFunctionalProperties

concept governmentContractIdentifier
  nonFunctionalProperties
dc:description hasValue "The unique number that identifies a government contract."
endNonFunctionalProperties

concept installAt
  nonFunctionalProperties
dc:description hasValue "The partner and/or location to which the product must be set up for use or service."
endNonFunctionalProperties

concept isDropShip
  nonFunctionalProperties
dc:description hasValue "Indicates whether the order is a drop shipment."
endNonFunctionalProperties

concept affirmationIndicator subConceptOf xsd:string
  nonFunctionalProperties
dc:description hasValue "Used to indicate "Yes", "No" statements."
endNonFunctionalProperties

concept orderShippingInformation
  nonFunctionalProperties
dc:description hasValue "The collection of business properties that describe information relating to shipping a product."
endNonFunctionalProperties
carrierginformation ofType carrierInformation

globalfreonboardcode ofType globalFreeOnBoardCode
globalshipmenttermsscode ofType globalShipmentTermsCode
globalshippingservicelevelcode ofType globalShippingServiceLevelCode
globalspecialfulfillmentrequestcode [1 n] ofType globalSpecialFulfillmentRequestCode
packlistrequirements ofType packListRequirements
specialhandlinginstruction ofType specialHandlingInstruction

concept carrierInformation
  nonFunctionalProperties
    dc:description hasValue "The collection of business properties that describe a carriers identification."
  endNonFunctionalProperties
endNonFunctionalProperties

concept accountIdentifier
  nonFunctionalProperties
    dc:description hasValue "The unique identifier that identifies an account."
  endNonFunctionalProperties
endNonFunctionalProperties

concept globalCarrierCode subConceptOf xsd:string
  nonFunctionalProperties
    dc:description hasValue "A unique carrier identification code, based on Standard Carrier Alpha Code(s) (SCAC)."
  endNonFunctionalProperties
endNonFunctionalProperties

concept globalFreeOnBoardCode subConceptOf xsd:string
  nonFunctionalProperties
    dc:description hasValue "Code identifying a specified point where a product is delivered or placed on board a carrier without charge."
  endNonFunctionalProperties
endNonFunctionalProperties

concept globalShipmentTermsCode subConceptOf xsd:string
  nonFunctionalProperties
    dc:description hasValue "Code identifying the terms under which a product is shipped."
  endNonFunctionalProperties
endNonFunctionalProperties

concept globalShippingServiceLevelCode subConceptOf xsd:string
  nonFunctionalProperties
    dc:description hasValue "Code identifying the shipping service level, e.g., overnight, same day."
  endNonFunctionalProperties
endNonFunctionalProperties

concept globalSpecialFulfillmentRequestCode subConceptOf xsd:string
  nonFunctionalProperties
    dc:description hasValue "Code identifying a special fulfillment request, e.g. delivery options."
  endNonFunctionalProperties
endNonFunctionalProperties

concept packListRequirements
  nonFunctionalProperties
    dc:description hasValue "Free form textual description, on the pack list, of requirements relating to the packing of the product."
  endNonFunctionalProperties
endNonFunctionalProperties

concept specialHandlingInstruction
  nonFunctionalProperties
    dc:description hasValue "The collection of business properties that describe product packaging or shipping handling instructions."
  endNonFunctionalProperties
endNonFunctionalProperties

concept specialHandlingCode subConceptOf xsd:string
  nonFunctionalProperties
    dc:description hasValue "Code identifying special handling or packaging requirements for the product."
  endNonFunctionalProperties
endNonFunctionalProperties

concept specialHandlingText
  nonFunctionalProperties
    dc:description hasValue "Free form textual description for how specified goods, packages or containers should be handled."
  endNonFunctionalProperties
endNonFunctionalProperties

concept productLineItem
  nonFunctionalProperties
    dc:description hasValue "The collection of business properties that..."
describe a business document entry for a product.

endNonFunctionalProperties
comments ofType comments
contractinformation {1 n} ofType contractInformation
countryOfOrigin ofType countryOfOrigin
customerInformation {1 n} ofType customerInformation
documentReference {1 n} ofType documentReference
expediteReferenceIdentifier ofType expediteReferenceIdentifier
globalProductUnitOfMeasureCode {1} ofType globalProductUnitOfMeasureCode
globalPurchaseOrderFillPriorityCode ofType globalPurchaseOrderFillPriorityCode
installAt ofType installAt
isdropsip {1} ofType isDropShip
linenumber {1} ofType lineNumber
orderquantity {1} ofType orderQuantity
ordershippingInformation ofType orderShippingInformation
productIdentification {1} ofType productIdentification
productSublineItem {1 n} ofType productSubLineItem
proprietaryInformation ofType proprietaryInformation
requestedEvent {1} ofType requestedEvent
requestedShipFrom {1 n} ofType requestedShipFrom
requestedUnitPrice ofType requestedUnitPrice
shipTo ofType shipTo
taxExemptStatus ofType taxExemptStatus
totalLineItemAmount ofType totalLineItemAmount

concept countryOfOrigin
  nonFunctionalProperties
dc:description hasValue "Country where product originates."
endNonFunctionalProperties
globalCountryCode {1} ofType globalCountryCode

concept customerInformation
  nonFunctionalProperties
dc:description hasValue "The collection of business properties that describe an end user."
endNonFunctionalProperties
customerProjectIdentifier ofType customerProjectIdentifier
globalCustomerTypeCode {1} ofType globalCustomerTypeCode
partnerDescription {1} ofType partnerDescription

concept customerProjectIdentifier
  nonFunctionalProperties
dc:description hasValue "The unique identification number that identifies a project for a given customer."
endNonFunctionalProperties
proprietaryReferenceIdentifier {1} ofType proprietaryReferenceIdentifier

concept globalCustomerTypeCode subConceptOf xsd:string
  nonFunctionalProperties
dc:description hasValue "Code identifying the type of end user."
endNonFunctionalProperties

concept nationalBusinessTaxIdentifier
  nonFunctionalProperties
dc:description hasValue "The national tax identification number assigned to a business."
endNonFunctionalProperties
businessTaxIdentifier {1} ofType businessTaxIdentifier
globalCountryCode ofType globalCountryCode

concept businessTaxIdentifier subConceptOf xsd:string
  nonFunctionalProperties
dc:description hasValue "Identifying number for Tax Information Field."
endNonFunctionalProperties

concept expediteReferenceIdentifier
  nonFunctionalProperties
dc:description hasValue "The unique identification number for the expediting of a product."
endNonFunctionalProperties
proprietaryReferenceIdentifier {1} ofType proprietaryReferenceIdentifier

concept globalProductUnitOfMeasureCode subConceptOf xsd:string
  nonFunctionalProperties
dc:description hasValue "Code identifying a product unit of measure."
endNonFunctionalProperties

concept orderQuantity
  nonFunctionalProperties
dc:description hasValue "The collection of business properties that describe various types of product quantity used in an ordering process."
endNonFunctionalProperties
concept productIdentification
   nonFunctionalProperties
      dc:description hasValue "The collection of business properties that describe proprietary and global identifier information regarding a product."
      endNonFunctionalProperties
      globalproductidentifier ofType globalProductIdentifier
      partnerproductidentification { 1 n } ofType partnerProductIdentification

concept requestedQuantity
   nonFunctionalProperties
      dc:description hasValue "The quantity of product requested."
      endNonFunctionalProperties
   productquantity { 1 } ofType productQuantity

concept productQuantity subConceptOf xsd:float
   nonFunctionalProperties
      dc:description hasValue "A quantity specifying the number of product units."
      endNonFunctionalProperties

concept partnerProductIdentification
   nonFunctionalProperties
      dc:description hasValue "The collection of business properties that describe proprietary part information."
      endNonFunctionalProperties
      globalpartnerclassificationcode { 1 } ofType globalPartnerClassificationCode
      proprietaryproductidentifier { 1 } ofType proprietaryProductIdentifier
      revisionidentifier ofType revisionIdentifier

concept globalProductIdentifier subConceptOf xsd:string
   nonFunctionalProperties
      dc:description hasValue "Global unique product identifier, expressed by the Global Trade Identification Number (GTIN)."
      endNonFunctionalProperties

concept proprietaryProductIdentifier subConceptOf xsd:string
   nonFunctionalProperties
      dc:description hasValue "An internal identifier used to identify a product."
      endNonFunctionalProperties

concept revisionIdentifier
   nonFunctionalProperties
      dc:description hasValue "Free form text that identifies a revision to a proprietary serial number."
      endNonFunctionalProperties
   freeformtext { 1 } ofType freeFormText

concept productSubLineItem
   nonFunctionalProperties
      dc:description hasValue "The collection of business properties that describe a part of a product line item."
      endNonFunctionalProperties
      comments ofType comments
      contractinformation ofType contractInformation
      countryoforigin ofType countryOfOrigin
      customerinformation { 1 n } ofType customerInformation
      expediterreferencerecidentifier ofType expediteReferenceIdentifier
      globalproductunitofmeasurecode { 1 } ofType globalProductUnitOfMeasureCode
      globalpurchaseorderfillprioritycode ofType globalPurchaseOrderFillPriorityCode
      installat ofType installAt
      isdropship { 1 } ofType isDropShip
      orderquantity { 1 } ofType orderQuantity
      ordershippinginformation ofType orderShippingInformation
      proprietaryinformation ofType proprietaryInformation
      requestedevent { 1 } ofType requestedEvent
      requestedshipfrom { 1 n } ofType requestedShipFrom
      requestedunitprice ofType requestedUnitPrice
      shipto ofType shipTo
      sublineitem ofType subLineItem

concept proprietaryInformation
   nonFunctionalProperties
      dc:description hasValue "A free form textual description of information, relating to a product."
      endNonFunctionalProperties
   freeformtext { 1 } ofType freeFormText

concept requestedEvent
   nonFunctionalProperties
      dc:description hasValue "The date a transportation event is requested to occur."
      endNonFunctionalProperties
   transportationevent ofType transportationEvent
concept transportationEvent
  nonFunctionalProperties
dc:description hasValue "The collection of business properties that describe the occurrence of the public conveyance of goods as a commercial enterprise."
endNonFunctionalProperties
beginTime ofType beginTime
datetimestamp ofType dt:dateAndTime
endtime ofType endTime
globalTransportEventCode ofType globalTransportEventCode

concept beginTime
  nonFunctionalProperties
dc:description hasValue "The start time of a time period."
endNonFunctionalProperties
timestamp ofType dt:dateAndTime

concept endTime
  nonFunctionalProperties
dc:description hasValue "The end time of a time period."
endNonFunctionalProperties
timestamp ofType dt:dateAndTime

concept globalTransportEventCode subConceptOf xsd:string
  nonFunctionalProperties
dc:description hasValue "Code identifying an event during the transportation of a shipment."
endNonFunctionalProperties
physicaladdress ofType physicalAddress

concept requestedShipFrom
  nonFunctionalProperties
dc:description hasValue "The location where the product is requested to be shipped from."
endNonFunctionalProperties
physicaladdress ofType physicalAddress

concept requestedUnitPrice
  nonFunctionalProperties
dc:description hasValue "The price requested for a unit of product."
endNonFunctionalProperties
financialamount ofType financialAmount

concept financialAmount
  nonFunctionalProperties
dc:description hasValue "The collection of business properties that describe the monetary amount defined by a specified currency."
endNonFunctionalProperties
globalcurrencycode ofType cu:currency
globalmonetaryamounttypecode ofType globalMonetaryAmountTypeCode
invoicechargetypecode ofType invoiceChargeTypeCode
monetaryamount ofType monetaryAmount

concept globalMonetaryAmountTypeCode subConceptOf xsd:string
  nonFunctionalProperties
dc:description hasValue "Code identifying whether the monetary amount is a debit or credit."
endNonFunctionalProperties

concept invoiceChargeTypeCode subConceptOf xsd:string
  nonFunctionalProperties
dc:description hasValue "Code identifying the values for the types of charges contained in an invoice."
endNonFunctionalProperties

concept monetaryAmount subConceptOf xsd:float
  nonFunctionalProperties
dc:description hasValue "Magnitude of currency amount."
endNonFunctionalProperties

concept shipTo
  nonFunctionalProperties
dc:description hasValue "The partner and/or location to which the product must be delivered."
endNonFunctionalProperties
partnerdescription {1} ofType partnerDescription

concept subLineItem
  nonFunctionalProperties
dc:description hasValue "Information contained within a subline."
endNonFunctionalProperties
linenumber {1} ofType lineNumber

concept taxExemptStatus
  nonFunctionalProperties
dc:description hasValue "The collection of business properties that describe tax exemption conditions."
endNonFunctionalProperties
istaxexempt ofType isTaxExempt
taxexemption ofType taxExemption

concept isTaxExempt
nonFunctionalProperties
dc:description hasValue "Indicates whether a product is exempt from taxation."
endNonFunctionalProperties
affirmationindicator {1} ofType affirmationIndicator

concept taxExemption
nonFunctionalProperties
dc:description hasValue "The collection of business properties that describe tax exemption type and identification information."
endNonFunctionalProperties
globaltaxexemptioncode ofType globalTaxExemptionCode
taxexemptioncertificationidentifier ofType taxExemptionCertificationIdentifier

concept globalTaxExemptionCode subConceptOf xsd:string
nonFunctionalProperties
dc:description hasValue "Code identifying the type of tax exemption for a product or service."
endNonFunctionalProperties

concept taxExemptionCertificationIdentifier
nonFunctionalProperties
dc:description hasValue "The unique identifier that represents the authorization for a product or services tax exempt status."
endNonFunctionalProperties
proprietaryreferenceidentifier ofType ProprietaryReferenceIdentifier

concept totalLineItemAmount
nonFunctionalProperties
dc:description hasValue "The monetary total associated with a line item."
endNonFunctionalProperties
financialamount ofType financialAmount

concept secondaryBuyer
nonFunctionalProperties
dc:description hasValue "The collection of business properties that describe an alternate buyer."
endNonFunctionalProperties
partnerdescription {1} ofType partnerDescription
secondarybuyerpurchaseorderidentifier ofType secondaryBuyerPurchaseOrderIdentifier

concept secondaryBuyerPurchaseOrderIdentifier
nonFunctionalProperties
dc:description hasValue "Unique number that identifies a purchase order issued by the secondary buyer."
endNonFunctionalProperties
proprietarydocumentidentifier {1} ofType proprietaryDocumentIdentifier

concept totalAmount
nonFunctionalProperties
dc:description hasValue "Total price for an entire invoice."
endNonFunctionalProperties
financialamount ofType financialAmount

concept thisDocumentGenerationDateTime
nonFunctionalProperties
dc:description hasValue "The date-time stamp indicating when this business document was generated."
endNonFunctionalProperties
datetimestamp ofType dt:dateAndTime

concept thisDocumentIdentifier
nonFunctionalProperties
dc:description hasValue "The information that identifies the business document being sent. This identifier is used to represent the specific business document associated with the defined business process."
endNonFunctionalProperties
proprietarydocumentidentifier {1} ofType proprietaryDocumentIdentifier