



## D3.2 WSMO Use Case "Virtual Travel Agency" v0.1

WSMO Working Draft 04 October 2004

**This version:**

<http://www.wsmo.org/2004/d3/d3.2/b2c/20041004/>

**Latest version:**

<http://www.wsmo.org/2004/d3/d3.2/b2c/>

**Previous version:**

<http://www.wsmo.org/2004/d3/d3.2/v0.1/20040719/>

**Editors:**

Michael Stollberg  
Holger Lausen  
Axel Polleres  
Rubén Lara

**Authors:**

Michael Stollberg  
Holger Lausen  
Axel Polleres  
Rubén Lara  
Uwe Keller  
Michal Zaremba  
Armin Haller  
Dieter Fensel  
Michael Kifer

**Reviewer:**

Christoph Bussler

This document is also available in non-normative [PDF](#) version.

Copyright © 2004 [DERI](#)®, All Rights Reserved. [DERI](#) liability, trademark, document use, and software licensing rules apply.

---

### Abstract

This document specifies a concrete Use Case for modeling Semantic Web Services with the Web Service Modeling Ontology WSMO. In accordance to the B2C application scenario described in the [WSMO Use Case Overview document, section 2.1](#), 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. 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 in this use case is the concrete modeling of the top WSMO components, resulting in the specification of WSMO, 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

WSMO Standard: [D2 v1.0 Web Service Modeling Ontology \(WSMO\)](#), last version at: <http://www.wsmo.org/2004/d2/>

WSMO Primer: [D3.1 v0.1 WSMO Primer](#)

Web Service Modeling Language WSML: [D16.0 v0.2 WSMO The WSML Family of Representation Languages](#)

WSMO Discovery: [D5.1 v0.1 WSMO Discovery](#)

---

## Table of Contents

### [1. Introduction](#)

### [2. Use Case Overview](#)

### [3. WSMO Use Case Modeling](#)

#### [3.1 Ontologies](#)

#### [3.2 Goals](#)

#### [3.3 Web Services](#)

#### [3.4 Mediators](#)

##### [3.4.1 OO Mediators](#)

##### [3.4.2 WG Mediators](#)

##### [3.4.3 GG Mediators](#)

##### [3.4.4 WW Mediators](#)

### [4. Web Service Discovery within WSMO](#)

### [5. Conclusions and Future Work](#)

### [References](#)

### [Acknowledgements](#)

### [Appendix: Change Tracking](#)

---

## 1. Introduction

In accordance to the B2C application scenario described in the [WSMO Use Case Overview document, section 2.1](#), 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. We specify the WSMO 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 a goal for an international train connection from Innsbruck to Frankfurt on 17th May 2004, at 16.00 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 use case is the first, initial use case specified of testing, elaborating, and recursively specific WSMO components. The rationale for choosing this first use case is that it allows to showcase and test describing all WSMO components identified in WSMO Standard. The setting is kept simple on purpose; more complex cases to be added at a later stage of this deliverable can build upon this, providing models of more involved scenarios. The main focus of this use case is the test and recursive specification of the WSMO components specification in [Roman et al., 2004], elaboration of the Web Service Modeling Language WSML [de Bruijn, 2004], and development and testing of WSMO Web Service Discovery [Keller et. al., 2004].

This document is structured as follows: [Section 2](#) gives an overview of the use case, identifying the needed WSMO components; [Section 3](#) 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 within this use case; finally, [Section 5](#) concludes the use case. The [Appendix](#) provides a change tracking to previous version of the document.

## 2. Use Case Overview

The following properties have to be covered in our use case modeling. For each of the WSMO top-level components, a separate table describes informally which parts of the use case are concerned.

Table 1. Properties Domain Ontologies

O1	Ontological information are needed on international train itineraries, on notions of date and time, on the purchasing process, as well as on persons, locations, and addresses. This information should be kept in separate re-usable ontologies, following the modularity principle of ontology design.
O2	An itinerary is described by its start and end locations, date and time of departure and arrival, stations which the train passes (particularly, the station where the border is crossed) and is done by some passenger.
O3	An itinerary describes a valid international train connection.
O4	There has to be traveller / customer that does the itinerar/buys a train ticket
O5	There exists a concept that defines whether a location is located at the border between 2 countries
O6	A ticket is valid for exactly 1 itinerary and has a price
O7	A ticket is valid for exactly 1 customer
O9	The purchase ontology has to identify the buyer and seller roles, a product with a price, and valid payment methods
O10	We need to be able to express valid payment methods. The only valid payment method for online tickets is credit card payment
O11	Information on Date and Time should allow axiomatic expressions on dependencies of specific dates and times , i.e. expression that define relationships like 'after' or 'before '

Table 2. Properties Goals

G1	Booking an Online Train Ticket
G1.1	From Innsbruck to Frankfurt
G1.2	Start time: 17th July 2004, at 18.00 local time.

Table 3. Properties Web Services

W1	A National Train Operator, here the Austrian ÖBB, provides an end-user Web Service that offers a search facility for international train connections and a facility for buying international train tickets online.
W2	The search facility takes a start location, an end location, and a departure date as input and returns a set of itineraries.
W3	The facility for buying train tickets online takes a specific itinerary with start location and end location in countries of its coverage, the information of the customer and the number of his (not expired) credit card number as input, and it returns a ticket for this itinerary as the result.
W4	The user interacts with the end-user Web Service which aggregates the search and purchasing Web Services from possibly different providers like ÖBB, DB, etc.

Table 4. Properties Mediators

M1	There need to be OO Mediators that integrate the distinct ontologies used as terminology definitions.
M2	If there are terminological mismatches between the ontologies used in the Goal or the Web Service description, OO Mediators have to be defined to resolve these.
M3	If there are differences between the Goal and the ÖBB-Web Service, a WG Mediator is needed to resolve these.
M4	if there are mismatches between the search facility Web Service and purchase Web Service (which are composed into the end-user Web Service), then a WW Mediator has to be defined which resolves the mismatches.

### 3. WSMO Use Case Modeling

The following exemplifies the specification of this use case within the Web Service Modeling Ontology WSMO. The provided listings use the conceptual model presented in WSMO, final version 1.0 [Roman et al., 2004]. In the listings, we provide the models of each WSMO component according to the syntax of WSML [de Bruijn, 2004], validated by the [WSML Validator](#), version 1.29 dated from 21 September 2004.

#### 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 these ontologies in different settings and use cases - for example, notions of 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.

The "International Train Ticket" Ontology defines a train trip and the surrounding concepts as defined in the WSMO 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 defines travel itineraries for trips by plane. Our ontology reuses the itinerary and flight concepts and adapts 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>, 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 harmonize ontology, as they are likely to include the travelling subdomain.

## Listing 1. Domain Ontology "International Train Ticket"

```

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/d3.2/v0.1/20041004/resources/owlPersonMediator.wsml>>
  loc:<<http://www.wsmo.org/ontologies/location#>>
  geo:<<http://www.wsmo.org/2004/d3/d3.2/v0.1/20041004/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 <<http://www.deri.org/foaf#deri>>
    dc:subject hasValues {"Train", "Itinerary", "Train Connection", "Ticket"}
    dc:description hasValue "International Train Connections"
    dc:publisher hasValue <<http://www.deri.org/foaf#deri>>
    dc:contributor hasValues {<<http://www.deri.org/foaf#stollberg>>,
      <<http://homepage.uibk.ac.at/~C703225/foaf.rdf>>,
      <<http://homepage.uibk.ac.at/~c703240/foaf.rdf>>,
      <<http://www.deri.org/foaf#polleres>>}
    dc:date hasValue "2004-10-04"
    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:source hasValue <<http://www.daml.org/2001/06/itinerary/itinerary-ont>>
    dc:language hasValue "en-US"
    dc:relation hasValues {<<http://www.daml.org/2001/06/itinerary/itinerary-ont>>,
      <<http://daml.umbc.edu/ontologies/ittalks/person>>,
      <<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: 1.10 $"
  endNonFunctionalProperties

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

  usedMediators {<<http://www.wsmo.org/2004/d3/d3.2/v0.1/20041004/resources/owlPersonMediator.wsml>>,
    <<http://www.wsmo.org/2004/d3/d3.2/v0.1/20041004/resources/owlFactBookMediator.wsml>>}

  concept station subConceptOf geo:geographicLocation
    nonFunctionalProperties
      dc:description hasValue "Train station"
    endNonFunctionalProperties
    code ofType xsd:string
    nonFunctionalProperties
      dc:description hasValue "Code of the station"
    endNonFunctionalProperties
    borderToCountry ofType loc:border
    nonFunctionalProperties
      dc:description hasValue "For stations located at the border"
    endNonFunctionalProperties

  concept itinerary
    nonFunctionalProperties
      dc:description hasValue "An itinerary between two locations"
    endNonFunctionalProperties
    passenger ofType prs:person
    nonFunctionalProperties
      dc:description hasValue "prs:person is a subset of vCard (http://www.ietf.org/rfc/rfc2425.txt)"
    endNonFunctionalProperties
    recordLocatorNumber ofType xsd:string
    trip ofType trip

  concept trip
    start ofType loc:location
    end ofType loc:location
    via ofType set loc:location
    departure ofType dt:dateAndTime
    arrival ofType dt:dateAndTime
    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
    and ?Start = ?End.

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}

```

Please notice that the link to large set of instances is missing in WSMO. 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. The inclusion of links to large set of instances will be considered in future versions of WSMO.

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 calendarclock 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.

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/20041004/dt.wsml>>

  nonFunctionalProperties
    dc:title hasValue "Date and Time Ontology"
    dc:creator hasValue "DERI International"
    dc:subject hasValues {"Date", "Time", "Date and Time Algebra"}
    dc:description hasValue "generic representation of data and time including basic 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/damlltime/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: 1.15 $"
  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
    start ofType instant
    end ofType instant

  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

  concept dayOfMonth subConceptOf xsd:integer
    nonFunctionalProperties
      dc:description hasValue "day of a month is represented by an integer"
    endNonFunctionalProperties

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

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

  concept 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 then 9th of March 1924 from the Julian to the Gregorian Calendar)"
    comment: The following dc:source indicates which country changed in which year
    comment: 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 and X is an instant or an
    interval"
  endNonFunctionalProperties
  interval ofType interval
  intervalOrInstant ofType instantOrInterVal

concept instantOrInterVal
  definedBy
    forAll ?x (?x memberOf 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
    dc:description hasValue "integrity constraint for valid dayOfMonths"
  endNonFunctionalProperties
  definedBy
    constraint
      ?X memberOf dayOfMonth and
      (X < 1 or X > 31).

axiom validDate
  nonFunctionalProperties
    dc:description hasValue "Integrity Constraints for date.
      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.

      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.

      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"
    dc:source hasValue <<http://www.tondering.dk/claus/cal/node3.html>>
  endNonFunctionalProperties
  definedBy
    constraint
      ?X memberOf date and (
        (?X.dayOfMonth > 28 and ?X.monthOfYear = 2 and
          not ((modulo(?X.year ,4) = 0 and not modulo(?X.year ,100) = 0)
            or modulo(?X.year ,400) = 0))
        or (?X.dayOfMonth > 29 and ?X.monthOfYear = 2)
        or (?X.dayOfMonth > 30 and ?X.monthOfYear = 4)
        or (?X.dayOfMonth > 30 and ?X.monthOfYear = 6)
        or (?X.dayOfMonth > 30 and ?X.monthOfYear = 9)
        or (?X.dayOfMonth > 30 and ?X.monthOfYear = 11)).

axiom invalidHourOfDay
  nonFunctionalProperties
    dc:description hasValue "integrity constraint for valid hourOfDay:"
  endNonFunctionalProperties
  definedBy
    constraint
      ?X memberOf hourOfDay and
      (?X < 0 or ?X >= 24).

axiom invalidMinuteOfHour
  nonFunctionalProperties
    dc:description hasValue "integrity constraint for valid minuteOfHour:"
  endNonFunctionalProperties
  definedBy
    constraint
      ?X memberOf minuteOfHour and
      (?X < 0 or ?X >= 60).

axiom invalidSecondOfMinute
  nonFunctionalProperties
    dc:description hasValue "integrity constraint for valid secondOfMinute:"
  endNonFunctionalProperties
  definedBy
    constraint
      ?X memberOf secondOfMinute and
      (?X < 0 or ?X >= 60).

axiom invalidInterval
  definedBy
    constraint
      ?X memberOf interval and ?X.start >= ?X.end.

axiom equalityDate
  nonFunctionalProperties
    dc:description hasValue "computes equality of a date"
  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.

**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**

?X > ?Y <- ?Y < ?X.

**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 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**

?JDN = ?C + ?D + ?E + ?F - 1524.

**axiom** daysBetweenDates

**nonFunctionalProperties**

dc:description **hasValue** "the difference in days between 2 dates"

**endNonFunctionalProperties**

**definedBy**

daysBetween[instant1 **hasValue** ?D1, instant2 **hasValue** ?D2, result **hasValue** ?X] <-

?D1 **memberOf** date **and** ?D2 **memberOf** date **and**

?X = julianDayNumber(?D1) - julianDayNumber(?D2).

**axiom** equalityTime

**nonFunctionalProperties**

```

    dc:description hasValue "computes if two given times are the same"
endNonFunctionalProperties
definedBy
    ?X = ?Y <-
        ?X memberOf time and ?Y memberOf time and
        ?X.secondOfMinute = ?Y.secondOfMinute and
        ?X.minuteOfHour = ?Y.minuteOfHour and
        ?X.hourOfDay = ?Y.hourOfDay.

axiom beforeTime
nonFunctionalProperties
    dc:description hasValue "computes if a given time ?X is before another time ?Y"
endNonFunctionalProperties
definedBy
    ?X < ?Y <-
        ?X memberOf time and ?Y memberOf time and
        ((?X.secondOfMinute < ?Y.secondOfMinute and ?X.minuteOfHour = ?Y.minuteOfHour and ?X.hourOfDay = ?Y.h
        (?X.minuteOfHour < ?Y.minuteOfHour and ?X.hourOfDay = ?Y.hourOfDay) or
        (?X.hourOfDay < ?Y.hourOfDay)).

axiom afterTime
nonFunctionalProperties
    dc:description hasValue "defined as inverse of beforeTime"
endNonFunctionalProperties
definedBy
    ?X > ?Y <- ?Y < ?X.

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

axiom secondsBetweenTimes
nonFunctionalProperties
    dc:description hasValue "the difference in seconds between 2 times"
endNonFunctionalProperties
definedBy
    secondsBetween[instant1 hasValue ?T1, instant2 hasValue T2, result hasValue ?X] <-
        ?T1 memberOf time and ?T2 memberOf time and
        ?X = secondsFromMidnight(?T1) - secondsFromMidnight(?T2).

axiom equalityDateAndTime
nonFunctionalProperties
    dc:description hasValue "computes if Date and Time are equal"
endNonFunctionalProperties
definedBy
    ?X = ?Y <-
        ?X memberOf dateAndTime and ?Y memberOf dateAndTime and
        ?X.date = ?Y.date and
        ?X.time = ?Y.time.

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

axiom afterDateAndTime
nonFunctionalProperties
    dc:description hasValue "defined as inverse of beforeDateAndTime"
endNonFunctionalProperties
definedBy
    ?X > ?Y <- ?X
        memberOf dateAndTime and ?Y memberOf 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).

```

The "Purchase" ontology defines general concepts to make a purchase order request. The ontology is an WSMO 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 [\[EDIFACT\]](#).

## Listing 3. Domain Ontology "Purchase"

```

namespace
  <<http://www.wsmo.org/ontologies/purchase#>>
  po: <<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.wsmi#>>
  dt: <<http://www.wsmo.org/ontologies/dateTime#>>
  targetnamespace: <<http://www.wsmo.org/ontologies/purchase#>>

ontology <<http://wsmo.org/ontologies/purchase/20041004/po.wsmi#>>

  nonFunctionalProperties
    dc:title hasValue "Purchase Ontology"
    dc:creator hasValue "DERI International"
    dc:subject hasValues {"Purchase Order Request", "Buyer", "Seller", "Product Line Item", "Price", "Payment method", "Delive
    dc:description hasValue "general purchase order request ontology based on the 3A4 PIP of RosettaNet"
    dc:publisher hasValue "DERI International"
    dc:contributor hasValues {"Armin Haller, Maciej Zaremba"}
    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:relation hasValues {<<http://www.daml.ecs.soton.ac.uk/ont/currency.daml>>, <<http://www.wsmo.org/ontologies/dateTim
    dc:rights hasValue <<http://www.deri.org/privacy.html>>
    version hasValue "$Revision: 1.14 $"
  endNonFunctionalProperties

  importedOntologies
    <<http://www.wsmo.org/ontologies/dateTime#>>

  usedMediators
    ooMediator <<http://www.wsmo.org/2004/d3/d3.2/v0.1/20041004/resources/owlCurrencyMediator.wsmi#>>

  comment: conceptDefinitions
    concept pip3A4Purchase
      nonFunctionalProperties
        dc:description hasValue "Rosettanets Purchase Order Request Document"
      endNonFunctionalProperties
      buyer ofType buyer
      globaldocumentfunctioncode ofType globalDocumentFunctionCode
      purchaseorder ofType purchaseOrder
      thisdocumentgenerationdatetime ofType thisDocumentGenerationDateTime
      thisdocumentidentifier ofType thisDocumentIdentifier
      seller ofType seller

    concept buyer
      nonFunctionalProperties
        dc:description hasValue "The role initiating a business document exchange."
      endNonFunctionalProperties
      partnerroledescription ofType partnerRoleDescription

    concept seller
      nonFunctionalProperties
        dc:description hasValue "The role receiving the document in a business document exchange."
      endNonFunctionalProperties
      partnerroledescription 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 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

```

dc:description **hasValue** "Name of the contact person(s) within the organization."  
**endNonFunctionalProperties**  
 freeformtext **ofType** freeFormText

**concept** freeFormText **subConceptOf** xsd:string  
**nonFunctionalProperties**  
 dc:description **hasValue** "Unformatted text."  
**endNonFunctionalProperties**

**concept** emailAddress **subConceptOf** xsd:string  
**nonFunctionalProperties**  
 dc:description **hasValue** "Electronic mail address."  
**endNonFunctionalProperties**

**concept** facsimileNumber  
**nonFunctionalProperties**  
 dc:description **hasValue** "The numerical schema designed to achieve contact via facsimile."  
**endNonFunctionalProperties**  
 communicationsnumber **ofType** communicationsNumber

**concept** communicationsNumber **subConceptOf** xsd:string  
**nonFunctionalProperties**  
 dc:description **hasValue** "The electro-technical communication number,  
 e.g., telephone number, facsimile number, pager number."  
**endNonFunctionalProperties**

**concept** telephoneNumber  
**nonFunctionalProperties**  
 dc:description **hasValue** "The numerical schema designed to achieve contact via telephone."  
**endNonFunctionalProperties**  
 communicationsnumber **ofType** communicationsNumber

**concept** globalPartnerRoleClassificationCode **subConceptOf** xsd:string  
**nonFunctionalProperties**  
 dc:description **hasValue** "Code identifying a partys role in the supply chain."  
**endNonFunctionalProperties**

**concept** partnerDescription  
**nonFunctionalProperties**  
 dc:description **hasValue** "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."  
**endNonFunctionalProperties**  
 businessdescription **ofType** businessDescription  
 globalpartnerclassificationcode **ofType** globalPartnerClassificationCode  
 contactinformation **ofType** contactInformation  
 physicallocation **ofType** physicalLocation

**concept** businessDescription  
**nonFunctionalProperties**  
 dc:description **hasValue** "The collection of business properties that  
 describe a business identity **and** location."  
**endNonFunctionalProperties**  
 globalbusinessidentifier **ofType** globalBusinessIdentifier  
 globalsupplychaincode **ofType** globalSupplyChainCode  
 businessname **ofType** businessName  
 partnerbusinessidentification **ofType** set partnerBusinessIdentification  
 nationalbusinessstaxidentifier **ofType** nationalBusinessTaxIdentifier

**concept** globalBusinessIdentifier **subConceptOf** xsd:integer  
**nonFunctionalProperties**  
 dc:description **hasValue** "A unique business identifier."  
**endNonFunctionalProperties**

**concept** globalSupplyChainCode **subConceptOf** xsd:string  
**nonFunctionalProperties**  
 dc:description **hasValue** "Code identifying the supply chain for the partners **function**."  
**endNonFunctionalProperties**

**concept** globalPartnerClassificationCode **subConceptOf** xsd:string  
**nonFunctionalProperties**  
 dc:description **hasValue** "Code identifying a partners **function** in the supply chain."  
**endNonFunctionalProperties**

**concept** globalDocumentFunctionCode **subConceptOf** xsd:string  
**nonFunctionalProperties**  
 dc:description **hasValue** "Code identifying the **function** of  
 a document as either a request **or** a response."  
**endNonFunctionalProperties**

**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 set** contractInformation  
 documentreference **ofType set** documentReference  
 financingterms **ofType set** financingTerms  
 generalservicesadministrationnumber **ofType** generalServicesAdministrationNumber  
 globalgovernmentpriorityratingcode **ofType** globalGovernmentPriorityRatingCode  
 globalpurchaseorderfillprioritycode **ofType** globalPurchaseOrderFillPriorityCode  
 globalpurchaseordertypecode **ofType set** globalPurchaseOrderTypeCode  
 governmentcontractidentifier **ofType** governmentContractIdentifier  
 installat **ofType** installAt  
 isdropship **ofType** isDropShip  
 ordershippinginformation **ofType** orderShippingInformation  
 productlineitem **ofType set** productLineItem  
 proprietaryinformation **ofType** proprietaryInformation  
 requestedevent **ofType** requestedEvent  
 requestedshipfrom **ofType set** 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 **subConceptOf** 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 ofType set 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 ofType proprietaryDomainIdentifier
  proprietaryidentifierauthority ofType proprietaryIdentifierAuthority
  proprietarylocationidentifier 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 ofType set partnerLocationIdentification

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

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

concept addressLine3
nonFunctionalProperties
  dc:description hasValue "Line 3 of the physical address."
endNonFunctionalProperties

```

```

freeformtext ofType freeFormText

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

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

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

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

concept creditCardIdentifier
  nonFunctionalProperties
    dc:description hasValue "The unique number that identifies a credit card."
  endNonFunctionalProperties
  proprietaryreferenceidentifier 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 ofType dt:monthOfYear
  expYear 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 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 ofType contractIdentifier
primarycontractwith ofType primaryContractWith
secondarycontractwith ofType secondaryContractWith

concept contractIdentifier
nonFunctionalProperties
  dc:description hasValue "The unique number than identifies a contract."
endNonFunctionalProperties
proprietarydocumentidentifier 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 ofType partnerDescription

concept secondaryContractWith
nonFunctionalProperties
  dc:description hasValue "The subsequent party in a binding
  agreement between two or more persons or parties."
endNonFunctionalProperties
partnerdescription 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
timestamp ofType dt:dateAndTime
globaldocumentreferencetypecode ofType globalDocumentReferenceTypeCode
globalpartnerroleclassificationcode ofType globalPartnerRoleClassificationCode
linenumber ofType lineNumber
proprietarydocumentidentifier ofType proprietaryDocumentIdentifier
revisionnumber ofType revisionNumber

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 ofType set 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 ofType set 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 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 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 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 **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 **ofType** countableAmount

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

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

**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**  
 proprietarydocumentidentifier **ofType** proprietaryDocumentIdentifier

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

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

**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**  
 carrierinformation **ofType** carrierInformation  
 globalfreeonboardcode **ofType** globalFreeOnBoardCode  
 globalshipmenttermscode **ofType** globalShipmentTermsCode  
 globalshippingservicelevelcode **ofType** globalShippingServiceLevelCode  
 globalspecialfulfillmentrequestcode **ofType** **set** globalSpecialFulfillmentRequestCode  
 packlistrequirements **ofType** packListRequirements  
 specialhandlinginstruction **ofType** specialHandlingInstruction

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

**endNonFunctionalProperties**  
 accountIdentifier **ofType** accountIdentifier  
 globalCarrierCode **ofType** globalCarrierCode

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

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

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

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

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

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

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

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

**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 **ofType** set contractInformation  
 countryOfOrigin **ofType** countryOfOrigin  
 customerInformation **ofType** set customerInformation  
 documentReference **ofType** set documentReference  
 expediteReferenceIdentifier **ofType** expediteReferenceIdentifier  
 globalProductUnitOfMeasureCode **ofType** globalProductUnitOfMeasureCode  
 globalPurchaseOrderFillPriorityCode **ofType** globalPurchaseOrderFillPriorityCode  
 installAt **ofType** installAt  
 isDropShip **ofType** isDropShip  
 lineNumber **ofType** lineNumber  
 orderQuantity **ofType** orderQuantity

ordershippinginformation **ofType** orderShippingInformation  
 productidentification **ofType** productIdentification  
 productsublineitem **ofType set** productSubLineItem  
 proprietaryinformation **ofType** proprietaryInformation  
 requestedevent **ofType** requestedEvent  
 requestedshipfrom **ofType set** 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 **ofType** globalCountryCode

**concept** customerInformation

**nonFunctionalProperties**

dc:description **hasValue** "The collection of business properties that describe an end user."

**endNonFunctionalProperties**

customerprojectidentifier **ofType** customerProjectIdentifier

globalcustomertypecode **ofType** globalCustomerTypeCode

partnerdescription **ofType** partnerDescription

**concept** customerProjectIdentifier

**nonFunctionalProperties**

dc:description **hasValue** "The unique identification number that identifies a project for a given customer."

**endNonFunctionalProperties**

proprietaryreferenceidentifier **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 **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 **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**

requestedquantity **ofType** requestedQuantity

**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 **ofType set** partnerProductIdentification

**concept** requestedQuantity

**nonFunctionalProperties**

dc:description **hasValue** "The quantity of product requested."

**endNonFunctionalProperties**

productquantity **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 **ofType** globalPartnerClassificationCode  
 proprietaryproductidentifier **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 **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 **ofType set** customerInformation  
 expeditereferenceidentifier **ofType** expediteReferenceIdentifier  
 globalproductunitofmeasurecode **ofType** globalProductUnitOfMeasureCode  
 globalpurchaseorderfillprioritycode **ofType** globalPurchaseOrderFillPriorityCode  
 installat **ofType** installAt  
 isdropship **ofType** isDropShip  
 orderquantity **ofType** orderQuantity  
 ordershippinginformation **ofType** orderShippingInformation  
 proprietaryinformation **ofType** proprietaryInformation  
 requestedevent **ofType** requestedEvent  
 requestedshipfrom **ofType set** 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 **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  
 datestamp **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**

**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  
 invoicechargestypecode **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 **ofType** partnerDescription

**concept** subLineItem  
**nonFunctionalProperties**  
 dc:description **hasValue** "Information contained within a subline."  
**endNonFunctionalProperties**  
 linenumber **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 **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 ofType partnerDescription
  secondarybuyerpurchaseorderid ofType secondaryBuyerPurchaseOrderIdentifier

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

comment: axiomDefinitions
axiom validBuyer
  nonFunctionalProperties
    dc:description hasValue "Exactly one buyer."
  endNonFunctionalProperties
  definedBy
    forAll ?x (exists ?y (?x memberOf pip3A4Purchase -> ?x[buyer hasValue ?y])).

axiom validPurchaseorder
  nonFunctionalProperties
    dc:description hasValue "Exactly one purchaseorder."
  endNonFunctionalProperties
  definedBy
    forAll ?x (exists ?y (?x memberOf pip3A4Purchase -> ?x[purchaseorder hasValue ?y])).

axiom validThisdocumentgenerationdatetime
  nonFunctionalProperties
    dc:description hasValue "Exactly one thisdocumentgenerationdatetime."
  endNonFunctionalProperties
  definedBy
    forAll ?x (exists ?y (?x memberOf pip3A4Purchase -> ?x[thisdocumentgenerationdatetime hasValue ?y])).

axiom validThisdocumentidentifier
  nonFunctionalProperties
    dc:description hasValue "Exactly one thisdocumentidentifier."
  endNonFunctionalProperties
  definedBy
    forAll ?x (exists ?y (?x memberOf pip3A4Purchase -> ?x[thisdocumentidentifier hasValue ?y])).

axiom validSeller

```

```

    nonFunctionalProperties
      dc:description hasValue "Exactly one seller."
    endNonFunctionalProperties
  definedBy
    forAll ?x (exists ?y (?x memberOf pip3A4Purchase -> ?x[seller hasValue ?y])).

axiom validPartnerroledescription
  nonFunctionalProperties
    dc:description hasValue "Exactly one partnerroledescription."
  endNonFunctionalProperties
  definedBy
    forAll ?x (exists ?y (?x memberOf buyer or ?x memberOf seller or ?x memberOf toRole
      -> ?x[partnerroledescription hasValue ?y])).

axiom validGlobalpartnerroleclassificationcode
  nonFunctionalProperties
    dc:description hasValue "Exactly one globalpartnerroleclassificationcode."
  endNonFunctionalProperties
  definedBy
    forAll ?x (exists ?y (?x memberOf partnerRoleDescription or ?x memberOf documentReference
      -> ?x[globalpartnerroleclassificationcode hasValue ?y])).

/*****
the rest of these axioms are commented out, since they where not done in a proper wsml syntax yet

axiom validPartnerdescription
  nonFunctionalProperties
    dc:description hasValue "Exactly one partnerdescription."
  endNonFunctionalProperties
  definedBy
    "forAll ?x exists ?y ((?x memberOf partnerRoleDescription hasValue ?x[partnerdescription hasValue ?y]) a
      (?x memberOf billTo hasValue ?x[partnerdescription hasValue ?y]) and
      (?x memberOf financedBy hasValue ?x[partnerdescription hasValue ?y]) and
      (?x memberOf primaryContractWith hasValue ?x[partnerdescription hasValue ?y]) and
      (?x memberOf secondaryContractWith hasValue ?x[partnerdescription hasValue ?y]) and
      (?x memberOf installAt hasValue ?x[partnerdescription hasValue ?y]) and
      (?x memberOf secondaryBuyer hasValue ?x[partnerdescription hasValue ?y]) and
      (?x memberOf customerInformation hasValue ?x[partnerdescription hasValue ?y]) and
      (?x memberOf shipTo hasValue ?x[partnerdescription hasValue ?y]))"

axiom validFreeformtext
  nonFunctionalProperties
    dc:description hasValue "Exactly one freeformtext."
  endNonFunctionalProperties
  definedBy
    "forAll ?x exists ?y ((?x memberOf contactName hasValue ?x[freeformtext -> ?y]) and
      (?x memberOf comments hasValue ?x[freeformtext hasValue ?y]) and
      (?x memberOf accountName hasValue ?x[freeformtext hasValue ?y]) and
      (?x memberOf businessName hasValue ?x[freeformtext hasValue ?y]) and
      (?x memberOf addressLine1 hasValue ?x[freeformtext hasValue ?y]) and
      (?x memberOf addressLine2 hasValue ?x[freeformtext hasValue ?y]) and
      (?x memberOf addressLine3 hasValue ?x[freeformtext hasValue ?y]) and
      (?x memberOf cityName hasValue ?x[freeformtext hasValue ?y]) and
      (?x memberOf postOfficeBoxIdentifier hasValue ?x[freeformtext hasValue ?y]) and
      (?x memberOf regionName hasValue ?x[freeformtext hasValue ?y]) and
      (?x memberOf cardHolderName hasValue ?x[freeformtext hasValue ?y]) and
      (?x memberOf packListRequirements hasValue ?x[freeformtext hasValue ?y]) and
      (?x memberOf specialHandlingText hasValue ?x[freeformtext hasValue ?y]) and
      (?x memberOf revisionIdentifier hasValue ?x[freeformtext hasValue ?y]) and
      (?x memberOf proprietaryInformation hasValue ?x[freeformtext hasValue ?y]))"

axiom validCommunicationsnumber
  nonFunctionalProperties
    dc:description hasValue "Exactly one communicationsnumber."
  endNonFunctionalProperties
  definedBy
    "forAll ?x exists ?y ((?x memberOf facsimileNumber -> ?x[communicationsnumber -> ?y]) and
      (?x memberOf telephone -> ?x[communicationsnumber -> ?y]))"

axiom validBusinessdescription
  nonFunctionalProperties
    dc:description hasValue "Exactly one businessdescription."
  endNonFunctionalProperties
  definedBy
    "forAll ?x exists ?y (?x memberOf partnerDescription -> ?x[businessdescription -> ?y])"

axiom validGlobalpurchaseordertypecode
  nonFunctionalProperties
    dc:description hasValue "One or more globalpurchaseordertypecode."
  endNonFunctionalProperties

```

```

    definedBy
      "forAll ?x exists ?y (?x memberOf purchaseOrder -> ?x[globalpurchaseordertypecode ->> ?y])"

axiom validIsdropship
  nonFunctionalProperties
    dc:description hasValue "Exactly one isdropship."
  endNonFunctionalProperties
  definedBy
    "forAll ?x exists ?y ((?x memberOf purchaseOrder -> ?x[isdropship -> ?y]) and
      (?x memberOf productLineItem -> ?x[isdropship -> ?y]) and
      (?x memberOf productSubLineItem -> ?x[isdropship -> ?y]))"

axiom validProductlineitem
  nonFunctionalProperties
    dc:description hasValue "One or more productlineitem."
  endNonFunctionalProperties
  definedBy
    "forAll ?x exists ?y (?x memberOf purchaseOrder -> ?x[productlineitem ->> ?y])"

axiom validRequestedshipfrom
  nonFunctionalProperties
    dc:description hasValue "One or more requestedshipfrom."
  endNonFunctionalProperties
  definedBy
    "forAll ?x exists ?y (?x memberOf purchaseOrder -> ?x[requestedshipfrom ->> ?y])"

axiom validAccountname
  nonFunctionalProperties
    dc:description hasValue "Exactly one accountname."
  endNonFunctionalProperties
  definedBy
    "forAll ?x exists ?y (?x memberOf accountDescription -> ?x[accountname -> ?y])"

axiom validProprietarybusinessidentifier
  nonFunctionalProperties
    dc:description hasValue "Exactly one proprietarybusinessidentifier."
  endNonFunctionalProperties
  definedBy
    "forAll ?x exists ?y (?x memberOf partnerBusinessIdentification -> ?x[proprietarybusinessidentifier -> ?y])"

axiom validProprietarydomainidentifier
  nonFunctionalProperties
    dc:description hasValue "Exactly one proprietarydomainidentifier."
  endNonFunctionalProperties
  definedBy
    "forAll ?x exists ?y ((?x memberOf partnerBusinessIdentification -> ?x[proprietarydomainidentifier -> ?y]) and
      (?x memberOf partnerLocationIdentification -> ?x[proprietarydomainidentifier -> ?y]))"

axiom validProprietarydomainidentifier
  nonFunctionalProperties
    dc:description hasValue "Exactly one proprietarydomainidentifier."
  endNonFunctionalProperties
  definedBy
    "forAll ?x exists ?y (?x memberOf partnerbusinessidentification -> ?x[proprietarydomainidentifier -> ?y])"

axiom validProprietarylocationidentifier
  nonFunctionalProperties
    dc:description hasValue "Exactly one proprietarylocationidentifier."
  endNonFunctionalProperties
  definedBy
    "forAll ?x exists ?y (?x memberOf partnerLocationIdentification -> ?x[proprietarylocationidentifier -> ?y])"

axiom validCardholdername
  nonFunctionalProperties
    dc:description hasValue "Exactly one cardholdername."
  endNonFunctionalProperties
  definedBy
    "forAll ?x exists ?y (?x memberOf creditCard -> ?x[cardholdername -> ?y])"

axiom validCreditcardidentifier
  nonFunctionalProperties
    dc:description hasValue "Exactly one creditcardidentifier."
  endNonFunctionalProperties
  definedBy
    "forAll ?x exists ?y (?x memberOf creditCard -> ?x[creditcardidentifier -> ?y])"

axiom validExpirydate
  nonFunctionalProperties
    dc:description hasValue "Exactly one expirydate."
  endNonFunctionalProperties

```

```

    definedBy
      "forAll ?x exists ?y (?x memberOf creditCard -> ?x[expirydate -> ?y])"

axiom validGlobalcreditcardclassificationcode
  nonFunctionalProperties
    dc:description hasValue "Exactly one globalcreditcardclassificationcode."
  endNonFunctionalProperties
  definedBy
    "forAll ?x exists ?y (?x memberOf creditCard -> ?x[globalcreditcardclassificationcode -> ?y])"

axiom validProprietaryreferenceidentifier
  nonFunctionalProperties
    dc:description hasValue "Exactly one proprietaryreferenceidentifier."
  endNonFunctionalProperties
  definedBy
    "forAll ?x exists ?y ((?x memberOf creditCardIdentifier -> ?x[Proprietaryreferenceidentifier -> ?y]) and
      (?x memberOf accountIdentifier -> ?x[proprietaryreferenceidentifier -> ?y]) and
      (?x memberOf customerProjectIdentifier -> ?x[proprietaryreferenceidentifier -> ?y]) and
      (?x memberOf expediteReferenceIdentifier -> ?x[proprietaryreferenceidentifier -> ?y]))"

axiom validExpmonth
  nonFunctionalProperties
    dc:description hasValue "Exactly one expmonth."
  endNonFunctionalProperties
  definedBy
    "forAll ?x exists ?y (?x memberOf expiryDate -> ?x[expmonth -> ?y])"

axiom validExpyear
  nonFunctionalProperties
    dc:description hasValue "Exactly one expyear."
  endNonFunctionalProperties
  definedBy
    "forAll ?x exists ?y (?x memberOf expiryDate -> ?x[expyear -> ?y])"

axiom validChecknumber
  nonFunctionalProperties
    dc:description hasValue "Exactly one checknumber."
  endNonFunctionalProperties
  definedBy
    "forAll ?x exists ?y (?x memberOf prePaymentCheckNumber -> ?x[checknumber -> ?y])"

axiom validContractidentifier
  nonFunctionalProperties
    dc:description hasValue "Exactly one contractidentifier."
  endNonFunctionalProperties
  definedBy
    "forAll ?x exists ?y (?x memberOf contractInformation -> ?x[contractidentifier -> ?y])"

axiom validProprietarydocumentidentifier
  nonFunctionalProperties
    dc:description hasValue "Exactly one proprietarydocumentidentifier."
  endNonFunctionalProperties
  definedBy
    "forAll ?x exists ?y ((?x memberOf contractIdentifier -> ?x[proprietarydocumentidentifier -> ?y]) and
      (?x memberOf documentReference -> ?x[proprietarydocumentidentifier -> ?y]) and
      (?x memberOf generalServicesAdministrationNumber -> ?x[proprietarydocumentidentifier -> ?y]) and
      (?x memberOf governmentContractIdentifier -> ?x[proprietarydocumentidentifier -> ?y]) and
      (?x memberOf secondaryBuyerPurchaseOrderIdentifier -> ?x[proprietarydocumentidentifier -> ?y]) and
      (?x memberOf thisDocumentIdentifier -> ?x[proprietarydocumentidentifier -> ?y]))"

axiom validGlobaldocumentreferencetypecode
  nonFunctionalProperties
    dc:description hasValue "Exactly one globaldocumentreferencetypecode."
  endNonFunctionalProperties
  definedBy
    "forAll ?x exists ?y (?x memberOf documentReference -> ?x[globaldocumentreferencetypecode -> ?y])"

axiom validDiscounts
  nonFunctionalProperties
    dc:description hasValue "One or more discounts."
  endNonFunctionalProperties
  definedBy
    "forAll ?x exists ?y (?x memberOf paymentTerms -> ?x[discounts ->> ?y])"

axiom validDayofmonth
  nonFunctionalProperties
    dc:description hasValue "Exactly one dayofmonth."
  endNonFunctionalProperties
  definedBy
    "forAll ?x exists ?y ((?x memberOf discountDay -> ?x[dayofmonth -> ?y]) and

```

```

        (?x memberOf netTermsDay -> ?x[dayofmonth -> ?y]))"

axiom validCountableamount
  nonFunctionalProperties
    dc:description hasValue "Exactly one countableamount."
  endNonFunctionalProperties
  definedBy
    "forAll ?x exists ?y ((?x memberOf discountDays -> ?x[countableamount-> ?y]) and
    (?x memberOf netTermsDays -> ?x[countableamount -> ?y]))"

axiom validPercentamount
  nonFunctionalProperties
    dc:description hasValue "Exactly one percentamount."
  endNonFunctionalProperties
  definedBy
    "forAll ?x exists ?y ((?x memberOf discountPercent -> ?x[percentamount -> ?y]) and
    (?x memberOf percentDue -> ?x[percentamount -> ?y]))"

axiom validAffirmationindicator
  nonFunctionalProperties
    dc:description hasValue "Exactly one affirmationindicator."
  endNonFunctionalProperties
  definedBy
    "forAll ?x exists ?y ((?x memberOf isDropShip -> ?x[affirmationindicator -> ?y]) and
    (?x memberOf isTaxExempt -> ?x[affirmationindicator -> ?y]))"

axiom validGlobalcarriercode
  nonFunctionalProperties
    dc:description hasValue "Exactly one globalcarriercode."
  endNonFunctionalProperties
  definedBy
    "forAll ?x exists ?y (?x memberOf carrierInformation -> ?x[globalcarriercode -> ?y])"

axiom validGlobalproductunitofmeasurecode
  nonFunctionalProperties
    dc:description hasValue "Exactly one globalproductunitofmeasurecode."
  endNonFunctionalProperties
  definedBy
    "forAll ?x exists ?y (?x memberOf productLineItem -> ?x[globalproductunitofmeasurecode -> ?y])"

axiom validLinenumbr
  nonFunctionalProperties
    dc:description hasValue "Exactly one linenumbr."
  endNonFunctionalProperties
  definedBy
    "forAll ?x exists ?y ((?x memberOf productLineItem -> ?x[linenumbr -> ?y]) and
    (?x memberOf subLineItem -> ?x[linenumbr -> ?y]))"

axiom validOrderquantity
  nonFunctionalProperties
    dc:description hasValue "Exactly one orderquantity."
  endNonFunctionalProperties
  definedBy
    "forAll ?x exists ?y ((?x memberOf productLineItem -> ?x[orderquantity -> ?y]) and
    (?x memberOf productSubLineItem -> ?x[orderquantity -> ?y]))"

axiom validProductidentification
  nonFunctionalProperties
    dc:description hasValue "Exactly one productidentification."
  endNonFunctionalProperties
  definedBy
    "forAll ?x exists ?y (?x memberOf productLineItem -> ?x[productidentification -> ?y])"

axiom validRequestedevent
  nonFunctionalProperties
    dc:description hasValue "Exactly one requestedevent."
  endNonFunctionalProperties
  definedBy
    "forAll ?x exists ?y ((?x memberOf productLineItem -> ?x[requestedevent -> ?y]) and
    (?x memberOf productSubLineItem -> ?x[requestedevent -> ?y]))"

axiom validGlobalcountrycode
  nonFunctionalProperties
    dc:description hasValue "Exactly one globalcountrycode."
  endNonFunctionalProperties
  definedBy
    "forAll ?x exists ?y (?x memberOf countryOfOrigin -> ?x[globalcountrycode -> ?y])"

axiom validGlobalcustomertypecode
  nonFunctionalProperties

```

```

        dc:description hasValue "Exactly one globalcustomertypecode."
    endNonFunctionalProperties
    definedBy
        "forAll ?x exists ?y (?x memberOf customerInformation -> ?x[globalcustomertypecode -> ?y])"

axiom validBusinessstaxidentifier
    nonFunctionalProperties
        dc:description hasValue "Exactly one businessstaxidentifier."
    endNonFunctionalProperties
    definedBy
        "forAll ?x exists ?y (?x memberOf nationalBusinessTaxIdentifier -> ?x[businessstaxidentifier -> ?y])"

axiom validGlobalcountrycode
    nonFunctionalProperties
        dc:description hasValue "Exactly one globalcountrycode."
    endNonFunctionalProperties
    definedBy
        "forAll ?x exists ?y (?x memberOf nationalBusinessTaxIdentifier -> ?x[globalcountrycode -> ?y])"

axiom validRequestedquantity
    nonFunctionalProperties
        dc:description hasValue "Exactly one requestedquantity."
    endNonFunctionalProperties
    definedBy
        "forAll ?x exists ?y (?x memberOf orderQuantity -> ?x[requestedquantity -> ?y])"

axiom validProductquantity
    nonFunctionalProperties
        dc:description hasValue "Exactly one productquantity."
    endNonFunctionalProperties
    definedBy
        "forAll ?x exists ?y (?x memberOf requestedQuantity -> ?x[productquantity -> ?y])"

axiom validGlobalpartnerclassificationcode
    nonFunctionalProperties
        dc:description hasValue "Exactly one globalpartnerclassificationcode."
    endNonFunctionalProperties
    definedBy
        "forAll ?x exists ?y (?x memberOf partnerProductIdentification -> ?x[globalpartnerclassificationcode -> ?y])"

axiom validProprietaryproductidentifier
    nonFunctionalProperties
        dc:description hasValue "Exactly one proprietaryproductidentifier."
    endNonFunctionalProperties
    definedBy
        "forAll ?x exists ?y (?x memberOf partnerProductIdentification -> ?x[proprietaryproductidentifier -> ?y])"

axiom validGlobalproductunitofmeasurecode
    nonFunctionalProperties
        dc:description hasValue "Exactly one globalproductunitofmeasurecode."
    endNonFunctionalProperties
    definedBy
        "forAll ?x exists ?y (?x memberOf productSubLineItem -> ?x[globalproductunitofmeasurecode -> ?y])"
    *****/

```

The "Locations Ontology" defines an concepts for locations, including cities and states, as well as postal addresses. This ontology the [DAML ontology for geographical locations](#), an ontology describing a wide variety of locations and geographical areas. The concept country is extended using the [OWL-Factbook ontology](#). The concept address reuses the [DAML address ontology](#).

## Listing 4. Domain Ontology "Locations"

```

namespace <<http://www.wsmo.org/ontologies/location#>>
  dc:<<http://purl.org/dc/elements/1.1#>>
  wsm1:<<http://www.wsmo.org/d2/#>>
  cnt:<<http://www.wsmo.org/2004/d3/d3.2/v0.1/20041004/resources/owlFactbookMediator.wsm1#>>
  geo:<<http://www.wsmo.org/2004/d3/d3.2/v0.1/20041004/resources/owlGeoMediator.wsm1#>>
  ad:<<http://www.wsmo.org/2004/d3/d3.2/v0.1/20041004/resources/owlAddressMediator.wsm1#>>
  xsd:<<http://www.w3.org/2001/XMLSchema#>>

ontology <<http://www.wsmo.org/ontologies/location>>

  nonFunctionalProperties
    dc:title hasValue "Locations Ontology"
    dc:creator hasValue <<http://www.deri.org/foaf#deri>>
    dc:subject hasValues {"Location", "Country", "State", "City", "Address"}
    dc:description hasValue "Ontology for representing locations in the current political/social system"
    dc:publisher hasValue <<http://www.deri.org/foaf#deri>>
    dc:contributor hasValues {<<http://www.deri.org/foaf#lara>>,
      <<http://www.deri.org/foaf#polleres>>}
    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:identifier hasValue <<http://www.wsmo.org/ontologies/location>>
    dc:source hasValue <<http://www.daml.org/2001/02/geofile/geofile-ont>>
    dc:language hasValue "en-US"
    dc:relation hasValues {<<http://www.daml.org/2001/09/countries/fips-10-4-ont>>,
      <<http://www.daml.org/2001/09/countries/iso-3166-ont>>,
      <<http://www.daml.org/2003/09/factbook/factbook-ont>>,
      <<http://www.daml.org/2001/02/geofile/geofile-ont>>,
      <<http://daml.umbc.edu/ontologies/ittalks/address>>}
    dc:coverage hasValue "ID:7029392 Name:World"
    dc:rights hasValue <<http://www.deri.org/privacy.html>>
    version hasValue "$Revision: 1.12 $"
  endNonFunctionalProperties

  usedMediators {<<http://www.wsmo.org/2004/d3/d3.2/v0.1/20041004/resources/owlFactbookMediator.wsm1#>>,
    <<http://www.wsmo.org/2004/d3/d3.2/v0.1/20041004/resources/owlAddressMediator.wsm1#>>,
    <<http://www.daml.org/2004/d3/d3.2/v0.1/20041004/resources/owlGeoMediator.wsm1#>>}

  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 than- for 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 moreThanDistance
  nonFunctionalProperties
    dc:description hasValue "Computes -more than- for 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).

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 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

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

```

## 3.2. Goals

Goals denote what a user wants as the result of the Web Service. For modeling the goal, we describe 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 a 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 5), a concrete Goal wherein a user wants to buy a train itinerary from Innsbruck to Frankfurt on a certain date (Listing 6), and a GG Mediator that restricts the generic Goal to tickets for traintrips within Austria and Germany (in the section of [Mediators](#) defined for this Use Case).

Listing 5 shows the generic Goal with the following description elements:

- **postcondition:** A itinerary for a with a customer that is of type person.
- **effect:** there shall be a trade for the itinerary bought, to be payed with a defined paymentmethod method, the and the customer must be of type Buyer as defined in the Purchase Ontology.

## Listing 5: Goal - buying a ticket online

```

namespace <<http://www.wsmo.org/2004/d3/d3.2/v0.1/20041004/resources/goal1#>>
  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/d3.2/v0.1/20041004/resources/goal1.wsm|>>

  nonFunctionalProperties
    dc:title hasValue "Buying a train ticket online"
    dc:creator hasValue <<http://www.deri.org/foaf#deri>>
    dc:subject hasValues {"Tickets", "Online Ticket Booking", "trip"}
    dc:description hasValue "Express the goal of buying a ticket for a train trip"
    dc:publisher hasValue <<http://www.deri.org/foaf#deri>>
    dc:contributor hasValues {<<http://www.deri.org/foaf#stollberg>>,
      <<http://www.deri.org/foaf#ara>>,
      <<http://www.deri.org/foaf#ausen>>,
      <<http://www.deri.org/foaf#polleres>>}
    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/d3.2/v0.1/20041004/resources/goal1.wsm|>>
    dc:language hasValue "en-US"
    dc:relation hasValues {<<http://www.wsmo.org/ontologies/trainConnection>>,
      <<http://www.wsmo.org/ontologies/purchase>>,
      <<http://www.wsmo.org/ontologies/location>>}
    dc:coverage hasValue "ID:7029392 Name:World"
    dc:rights hasValue <<http://deri.at/privacy.html>>
    version hasValue "$Revision: 1.6 $"
  endNonFunctionalProperties

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

  postcondition
    axiom havingItineraryForTrip
    nonFunctionalProperties
      dc:description hasValue "This goal expresses the general desire of buying a ticket for
        any kind of trip. Thus, the goal postcondition defines that there shall be an
        itinerary for a trip for a passenger that is a person."
    endNonFunctionalProperties
    definedBy
      ?SomeItinerary memberOf tc:itinerary[
        tc:trip hasValue ?SomeTrip,
        tc:passenger hasValue ?SomePassenger]
      and ?SomeTrip memberOf tc:trip
      and ?SomePassenger memberOf loc:person .

  effect
    axiom havingTradeForTrip
    nonFunctionalProperties
      dc:description hasValue "The goal effect is to have a trade with a provider (not specified)
        for the itinerary given; the buyer and the payment method are specified according
        to the Purchase ontology."
    endNonFunctionalProperties
    definedBy
      ?someTrade memberOf po:trade[
        items hasValues {?someTrip},
        buyer hasValue ?someBuyer,
        payment hasValue ?somePayment
      ]
      and ?someBuyer memberOf po:buyer
      and ?somePayment memberOf po:paymentMethod .

```

The concrete Goal states that the desire is to get the description of the itinerary bought, and that the effect of the Web Service has to be a trade between the train company and the requester for the desired itinerary. Listing 6 shows this Goal with the following elements:

- **postcondition:** A itinerary for a train trip from Innsbruck to Frankfurt on July, 17th 2004, valid for the customer Tim Berners-Lee.
- **effect:** there shall be a trade for the itinerary, to be payed with a given creditcard and, the bill of the trade will be sent to Tim's address, and the actual ticket will also be sent to

Tim's address.

Listing 6: Goal - buying a train ticket from Innsbruck to Frankfurt online

```

namespace <<http://www.wsmo.org/2004/d3/d3.2/v0.1/20041004/resources/goal#>>
  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#>>
  wsml:<<http://www.wsmo.org/2004/d2/#>>
  xsd:<<http://www.w3.org/2001/XMLSchema#>>

goal <<http://www.wsmo.org/2004/d3/d3.2/v0.1/20041004/resources/goal.wsml>>

  nonFunctionalProperties
    dc:title hasValue "Buying a train ticket from Innsbruck to Frankfurt on..."
    dc:creator hasValue <<http://www.deri.org/foaf#deri>>
    dc:subject hasValues {"Train Tickets", "Online Ticket Booking", "Train trip"}
    dc:description hasValue "Express the goal of buying a concrete ticket for a train trip"
    dc:publisher hasValue <<http://www.deri.org/foaf#deri>>
    dc:contributor hasValues{<<http://www.deri.org/foaf#stollberg>>,
      <<http://www.deri.org/foaf#lara>>,
      <<http://www.deri.org/foaf#lausen>>,
      <<http://www.deri.org/foaf#polleres>>,
      <<http://www.deri.org/foaf#haller>>}
    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/d3.2/v0.1/20041004/resources/goal.wsml>>
    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 "ID:7029392 Name:World"
    dc:rights hasValue <<http://deri.at/privacy.html>>
    version hasValue "$Revision: 1.16 $"
  endNonFunctionalProperties

  usedMediators
    <<http://www.wsmo.org/2004/d3/d3.2/v0.1/20041004/resources/ggm1.wsml>>
    //all other mediators and ontologies are inherited from the GG Mediator

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

  postcondition
    axiom havingItineraryForTrip
      nonFunctionalProperties
        dc:description hasValue "The goal postcondition is represented as a fact, in this case the fact is
          only specified partly, e.g. for the time of departure the minute and seconds
          are not specified.
          It represents that !Tim Berners-Lee! wants to go from innsbruckHbf to frankfurtHbf departing
          from innsbruckHbf at 17.07.2004 18h"
      endNonFunctionalProperties
      definedBy
        ?someItinerary memberOf tc:itinerary[
          tc:trip hasValue ?someTrip,
          tc:passenger hasValue ?somePassenger
        ]
        and ?someTrip memberOf tc:trainTrip[
          tc:start hasValue tc:innsbruckHbf,
          tc:end hasValue tc:frankfurtHbf,
          tc:departure hasValue ?someDeparture
        ]
        and ?somePassenger memberOf loc:person[
          tc:firstName hasValue "Tim"^^xsd:string,
          tc:lastName hasValue "Berners-Lee"^^xsd:string,
          tc:email hasValue "timbl@w3.org"^^xsd:string
        ]
        and ?someDeparture memberOf dt:dateAndTime[
          dt:date hasValue ?someDate,
          dt:time hasValue ?someTime
        ]
        and ?someDate memberOf dt:date[
          dt:dayOfMonth hasValue 17^^xsd:integer,
          dt:monthOfYear hasValue 7^^xsd:integer,
          dt:year hasValue 2004^^xsd:integer
        ]
        and ?someTime memberOf dt:time[
          dt:hourOfDay hasValue 18^^xsd:integer
        ]
      ]
    ]
  
```

**effect****axiom** havingTradeForTrip**nonFunctionalProperties**

dc:description **hasValue** "The **goal effect** is represented as a fact  
 It represents that Tim Berners-Lee wants to have a trade  
 with a provider (**not** specified) for the itinerary given;  
 the ticket should be delivered to his address **and** he wants  
 to pay by creditcard"

**endNonFunctionalProperties****definedBy**

```
?sometrade memberOf po:purchaseOrderRequest[
  po:buyer hasValue ?buyer,
  po:thisdocumentgenerationdatetime hasValue ?time,
  po:purchaseorder hasValue ?po,
  po:financedby hasValue ?financed,
  po:globalpurchaseordertypecode hasValue "Standard",
  po:isdropship hasValue ?isDropShip,
  po:productlineitem hasValue ?prodLine
]
and ?buyer memberOf po:buyer[
  po:partnerroledescription hasValue ?prole
]
and ?prole memberOf po:partnerRoleDescription[
  po:contactinformation hasValue ?contactinfo,
  po:globalpartnerroleclassificationcode hasValue po:customer ,
  po:partnerdescription hasValue ?partnerDesc
]
and ?contactinfo memberOf po:contactInformation[
  po:contactname hasValue "Tim Berners-Lee"^^xsd:string,
  po:emailaddress hasValue "tim@w3.org"^^xsd:string,
  po:telephonenumber hasValue ?phnno
]
and ?phnno memberOf po:telephoneNumber[
  po:communicationsnumber hasValue "0123456789" ^^xsd:string
]
and ?partnerDesc memberOf po:partnerDescription[
  po:businessdescription hasValue ?busdesc ,
  po:globalpartnerclassificationcode hasValue "End User"^^xsd:string,
  po:physicallocation hasValue ?loc
]
and ?loc memberOf po:physicalLocation[
  po:physicaladdress hasValue ?addr
]
and ?addr memberOf po:physicalAddress[
  po:addressline hasValue ?addrline,
  po:cityname hasValue ?city,
  po:globalcountrycode hasValue "US"^^xsd:string,
  po:nationalpostalcode hasValue "02103"^^xsd:string,
  po:regionname hasValue ?region
]
and ?region memberOf po:regionName[
  po:freeformtext hasValue "Massachusetts"^^xsd:string
]
and ?city memberOf po:cityName[
  po:freeformtext hasValue "Boston"^^xsd:string
]
and ?addrline memberOf po:addressLine[
  po:freeformtext hasValue "Tims street 3"^^xsd:string
]
and ?busdesc memberOf po:businessDescription[
  po:businessname hasValue ?busname
]
and ?busname memberOf po:businessname[
  po:freeformtext hasValue "Tim Berners-Lee"^^xsd:string
]
and ?time memberOf of:thisDocumentGenerationDateTime[
  po:datestamp hasValue ?dt
]
and ?dt memberOf dt:dateAndTime[
  dt:date hasValue ?dtD,
  dt:time hasValue ?dtT
]
and ?dtD memberOf dt:date[
  dt:dayOfMonth hasValue 18^^xsd:integer,
  dt:monthOfYear hasValue 07^^xsd:integer,
  dt:year hasValue 2004^^xsd:integer
]
and ?dtT memberOf dt:time[
  dt:hourOfDay hasValue 21^^xsd:integer,
```

```

    dt:minuteOfHour hasValue 05^^xsd:integer,
    dt:secondOfMinute hasValue 00^^xsd:integer
  ]

  and ?po memberOf po:purchaseorder[
    po:accountdescription hasValue ?accDesc
  ]
  and ?accDesc memberOf po:accountDescription[
    po:creditcard hasValue ?CC
  ]
  and ?CC memberOf po:creditCard[
    po:cardholdername hasValue ?ccHolder,
    po:creditcardidentifier hasValue ?ccID,
    po:expirydate hasValue ?ccDate,
    po:globalcreditcardclassificationcode hasValue "Master Card"^^xsd:string
  ]
  and ?ccHolder memberOf po:cardHolderName[
    po:freeformtext hasValue "Tim Berners-Lee"^^xsd:string
  ]
  and ?ccID memberOf po:creditCardIdentifier[
    po:proprietaryreferenceidentifier hasValue "5535 4464 6686 7747"^^xsd:string
  ]
  and ?ccDate memberOf po:expiryDate[
    po:expmonth hasValue 09^^xsd:integer,
    po:expyear hasValue 2007^^xsd:integer
  ]
  and ?isDropShip memberOf po:isDropShip[
    po:affirmationindicator hasValue "No"^^xsd:string
  ]
  and ?financed memberOf po:financedBy[
    po:partnerdescription hasValue ?partnerDesc
  ]
  and ?prodLine memberOf po:productLineItem[
    po:contractinformation hasValue ?contractI,
    po:globalproductunitofmeasurecode hasValue "Piece",
    po:isdropship hasValue ?isDropShip,
    po:linenumber hasValue ?line,
    po:orderquantity hasValue ?qty,
    po:productidentification hasValue ?prodId
  ]
  and ?prodId memberOf po:productIdentification[
    po:globalproductIdentifier hasValue ""^^xsd:string,
    po:requestedevent hasValue ?rEv
  ]
  and ?rEv memberOf po:requestedevent[
    po:transportationevent hasValue ?tEv,
    dt:globaltransporteventcode hasValue "Ship"^^xsd:string
  ]
  and ?tEv memberOf po:transportationEvent[
    dt:datestamp hasValue ?dt
  ]
  and ?dt memberOf dt:dateAndTime[
    dt:date hasValue ?d,
    dt:time hasValue ?t
  ]
  and ?d memberOf dt:date[
    dt:dayOfMonth hasValue 20^^xsd:integer,
    dt:monthOfYear hasValue 07^^xsd:integer,
    dt:year hasValue 2004^^xsd:integer
  ]
  and ?t memberOf dt:time[
    dt:hourOfDay hasValue 09^^xsd:integer,
    dt:minuteOfHour hasValue 00^^xsd:integer,
    dt:secondOfMinute hasValue 00^^xsd:integer
  ]
  and ?contractI memberOf po:contractInformation[
    po:contractidentifier hasValue ?contractID
  ]
  and ?contractID memberOf po:contractIdentifier[
    po:proprietarydocumentidentifier hasValue "POR123456"^^xsd:string
  ]
  and ?qty memberOf po:orderQuantity[
    po:requestedquantity hasValue ?rqty
  ]
  and ?rqty memberOf po:requestedQuantity[
    po:productquantity hasValue 1.00^^xsd:float
  ]
  ]

```

Notice that an instance of the concept 'Itinerary' is used as the value of the property 'Items' of the concept 'Trade'. In the ontologies defined above, Itinerary is not defined in tc.wsml as a subconcept of po:product. This subclassing should be done by an OO-mediator that imports the terminology required for the goal and takes care of this operation. Such a mediator will be included in the next version of this deliverable.

### 3.3 Web Services

As explained above, 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, which is 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 7 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, and the purchase intention of the buyer has to be a train itinerary with start and end locations in Austria or Germany. Furthermore, 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:** describes the possible trips the service can return wherefore the start and end location have to be either in Austria or Germany.
- **effect:** a trade is performed for the train itinerary given as a postcondition.

## Listing 7: ÖBB Web Service for Booking Online Train Tickets for Austria and Gern

```

namespace <<http://www.wsmo.org/2004/d3/d3.2/v0.1/20041004/resources/ws#>>
  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/d3.2/v0.1/20041004/resources/>>
  targetnamespace: <<http://www.wsmo.org/2004/d3/d3.2/v0.1/20041004/resources/ws#>>

webservice <<http://www.wsmo.org/2004/d3/d3.2/v0.1/20041004/resources/ws.wsml>>

  nonFunctionalProperties
    dc:title hasValue "OEBB 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: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 hasValues {tc:austria, tc:germany}
    dc:rights hasValue <<http://deri.at/privacy.html>>
    version hasValue "$Revision: 1.7 $"
  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>>}

  capability _#
    precondition
      axiom _#
        nonFunctionalProperties
          dc:description hasValue "the input has to be a buyer with a purchase intention for
            an itinerary 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.
            A credit card as payment method is expected."
        endNonFunctionalProperties
        definedBy
          ?Buyer memberOf po:buyer and
          ?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).
        assumption
          axiom _#
            nonFunctionalProperties
              dc:description hasValue "the credit card has to be valid, i.e. not expired.
                The current date is provided as a built-in functionality
                (currently defined explicitly as built-in function is not available)."
            endNonFunctionalProperties
            definedBy
              ?CreditCard memberOf po:creditCard[
                po:cardholdername hasValue ?someHolder,
                po:creditcardidentifier hasValue ?someIdentifier,
                po:expirydate hasValue ?someExpiration,
                po:globalcreditcardclassificationcode hasValue "Master Card"^^xsd:string
              ]
              and ?someHolder memberOf po:cardHolderName[
                po:freeformtext hasValue "Tim Berners-Lee"^^xsd:string
              ]
              and ?someIdentifier memberOf po:creditCardIdentifier[
                po:proprietaryreferenceidentifier hasValue "5535 4464 6686 7747"^^xsd:string
              ]
              and ?someExpiration memberOf po:expiryDate[
                po:expmonth hasValue "09"^^xsd:integer,
                po:expyear hasValue "2007"^^xsd:integer
              ]
            ]
          and (currentDate date year < ?CreditCard expirvdate exnyear or

```

```

(currentDate.date.monthOfYear <= ?CreditCard.expirydate.expmonth and currentDate.date.year = ?CreditCard.expir

postcondition
  axiom_#
    nonFunctionalProperties
      dc:description hasValue "the output of the service is a train trip 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
      ?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).

effect
  axiom_#
    nonFunctionalProperties
      dc:description hasValue "there shall be a trade for the train trip of the postcondition"
    endNonFunctionalProperties
    definedBy
      ?someTrade memberOf po:trade[
        po:items hasValues {?Trip},
        po:payment hasValue ?acceptedPayment
      ]
      and ?acceptedPayment memberOf po:creditCard .

interface_#
  nonFunctionalProperties
    dc:description hasValue "describes the Interface of Web Service (not specified yet)"
  endNonFunctionalProperties
  choreography ***
  orchestration ***

```

As in the modeling in the Goal, here an instance of the concept 'Itinerary' is used as the value of the property 'Items' of the concept 'Trade', while Itinerary is not defined in tc.wsml as a subconcept of po:product. The OO-mediator that imports the terminology required for the capability and performs this operation will be included in the next version of this deliverable.

## 3.4 Mediators

### 3.4.1 OO Mediators

OO Mediators "connect" ontologies with other ontologies or OO Mediators for refining ontologies, as well as for importing ontologies as the terminology definitions into other WSMO components. As the Goal and the Web Service specified above have homogeneous information spaces, we only have to specify OO Mediators for the existing ontologies used for the domain ontologies defined in this use case. Here, we have to define the following OO Mediators, specified in the Listings below:

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

Listing 7: OO-Mediator "importing the OWL Address Ontology to the Location Ontology"

```

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

ooMediator <<http://www.wsmo.org/2004/d3/d3.2/v0.1/20041004/resources/owlAddressMediator.wsmo>>
  nonFunctionalProperties
    dc:title hasValue "OO Mediator importing the OWL Factbook ontology to WSML"
    dc:creator hasValue <<http://www.deri.org/foaf#deri>>
    dc:subject hasValues {"ooMediator", "Address", "Locations"}
    dc:description hasValue "Mediator to import an OWL address ontology into a WSML locations ontology"
    dc:publisher hasValue <<http://www.deri.org/foaf#deri>>
    dc:contributor hasValue <<http://www.deri.org/foaf#polleres>>
    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/d3.2/v0.1/20041004/resources/owlAddressMediator.wsmo>>
    dc:language hasValue "en-us"
    dc:relation hasValues {<<http://daml.umbc.edu/ontologies/ittalks/address/>>,
      <<http://www.wsmo.org/ontologies/location>>}
    dc:coverage hasValue "ID:7029392 Name:World"
    dc:rights hasValue <<http://www.deri.org/privacy.html>>
    version hasValue "$Revision: 1.6 $"
  endNonFunctionalProperties

  source <<http://daml.umbc.edu/ontologies/ittalks/address/>>

  target <<http://www.wsmo.org/ontologies/location>>

  // useService http://138.232.65.151:8080/TranslatorService/OWL2WSML/

  mediationService <<http://138.232.65.151:8080/TranslatorService/OWL2WSML/>>

```

Listing 8: OO-Mediator "importing the OWL Currency Ontology into the Purchase Ontology"

```

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

ooMediator <<http://www.wsmo.org/2004/d3/d3.2/v0.1/20041004/resources/owlCurrencyMediator.wsmo>>
  nonFunctionalProperties
    dc:title hasValue "OO Mediator importing the OWL Currency ontology to WSML"
    dc:creator hasValue <<http://www.deri.org/foaf#deri>>
    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 <<http://www.deri.org/foaf#deri>>
    dc:contributor hasValue <<http://www.deri.org/foaf#lausen>>
    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/d3.2/v0.1/20041004/resources/owlCurrencyMediator.wsmo>>
    dc:language hasValue "en-us"
    dc:relation hasValues{<<http://www.daml.ecs.soton.ac.uk/ont/currency.daml>>,
      <<http://www.wsmo.org/ontologies/purchaseOrder>>}
    dc:coverage hasValue "ID:7029392 Name:World"
    dc:rights hasValue <<http://www.deri.org/privacy.html>>
    version hasValue "$Revision: 1.5 $"
  endNonFunctionalProperties

  source <<http://www.daml.ecs.soton.ac.uk/ont/currency.daml>>

  target <<http://www.wsmo.org/ontologies/purchaseOrder>>

  useService http://138.232.65.151:8080/TranslatorService/OWL2WSML/

```

Listing 9: OO-Mediator "importing the OWL Factbook into the Location Ontology"

```

namespace
dc:<<http://purl.org/dc/elements/1.1#>>
wsm1:<<http://www.wsmo.org/2004/d2/#>>

ooMediator <<http://www.wsmo.org/2004/d3/d3.2/v0.1/20041004/resources/owlFactbookMediator.wsm1>>
nonFunctionalProperties
dc:title hasValue "OO Mediator importing the OWL Factbook ontology to WSML"
dc:creator hasValue <<http://www.deri.org/foaf#deri>>
dc:subject hasValues {"ooMediator", "Factbook", "Locations"}
dc:description hasValue "Mediator to import an OWL factbook ontology into a WSML locations ontology"
dc:publisher hasValue <<http://www.deri.org/foaf#deri>>
dc:contributor hasValue <<http://www.deri.org/foaf#polleres>>
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/d3.2/v0.1/20041004/resources/owlFactbookMediator.wsm1>>
dc:language hasValue "en-us"
dc:relation hasValues {<<http://www.daml.org/2003/09/factbook/factbook-ont/>>,
  <<http://www.wsmo.org/ontologies/locations>>}
dc:coverage hasValue "ID:7029392 Name:World"
dc:rights hasValue <<http://www.deri.org/privacy.html>>
version hasValue "$Revision: 1.6 $"
endNonFunctionalProperties

source <<http://www.daml.org/2003/09/factbook/factbook-ont/>>

target <<http://www.wsmo.org/ontologies/locations>>

useService http://138.232.65.151:8080/TranslatorService/OWL2WSML/

```

Listing 10: OO-Mediator "importing the OWL Person Ontology into the Train Connection Ontology"

```

namespace
dc:<<http://purl.org/dc/elements/1.1#>>
wsm1:<<http://www.wsmo.org/2004/d2/#>>

ooMediator <<http://www.wsmo.org/2004/d3/d3.2/v0.1/20041004/resources/owlPersonMediator.wsm1>>
nonFunctionalProperties
dc:title hasValue "OO Mediator importing the OWL Person ontology to WSML"
dc:creator hasValue <<http://www.deri.org/foaf#deri>>
dc:subject hasValues {"ooMediator", "Person", "Train Connections"}
dc:description hasValue "Mediator to import an OWL person ontology into a WSML train connections ontology"
dc:publisher hasValue <<http://www.deri.org/foaf#deri>>
dc:contributor hasValue <<http://www.deri.org/foaf#polleres>>
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/d3.2/v0.1/20041004/resources/owlPersonMediator.wsm1>>
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: 1.5 $"
endNonFunctionalProperties

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

target <<http://www.wsmo.org/ontologies/trainConnection>>

useService http://138.232.65.151:8080/TranslatorService/OWL2WSML/

```

Notice that the mediation services are not specified. For importing an OWL ontology into a WSML ontology, it is obvious that such mediation services are required. The terminology to express the capability of mediation services as well as the requester goals is not defined at the moment. This terminology, modeling the ontology mediation domain, has to be included in future versions of the deliverable and the necessary goals and capabilities have to be defined using such terminology.

### 3.4.2 WG Mediators

A WG Mediator links a Web Service to a Goal, resolves terminological mismatches, and states the functional difference (if any) between both. The main application of WG Mediators is handling of partial matches within Web Service discovery. For resolving terminological mismatches, OO Mediators are applied, similar to the ones specified above. The functional difference is stated in the reduction which restricts the set of valid ontology objects to be passed between the Web Service and the Goal.

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 not terminology mismatches), and there is no functional differences between the Goal and the Capability. An WG Mediator with a reduction would be needed if the Web Service Capability specifies that train tickets as well as plane tickets are sold: Therefore, the reduction would restrict the valid set of information to train tickets, as requested by the Goal.

### 3.4.3 GG Mediators

A GG Mediator connects Goals, specifying the possible functionality reduction. For example, a GG Mediator would connect a Goal "buy a ticket" with another Goal "buy a train ticket" by stating the ontological correspondance between the Goals as a reduction. If 'train ticket' is a subclass of 'ticket', than the reduction in the GG Mediator would specify that valid instances for the second Goal have to be 'train ticket **subclassof** ticket'.

In our use case, we have defined a generic Goal for buying a ticket for any kind of trip (Listing 5), a concrete Goal wherein a user wants to buy a train itinerary from Innsbruck to Frankfurt on a certain date (Listing 6). The GG Mediator restricts the generic Goal to buying train tickets for itineraries in Austria and Germany; this GG Meditaor is used within the concrete Goal, wherein only specific values are defined for the ontological notions provided by the GG Mediator. Therefore, the GG Mediator has the following description elements:

- **sourcecomponent:** this is the generic Goal defined in Listing 5.
- **targetcomponent:** the concrete Goal defined in Listing 6.
- **usedMeditators:** this element is intended to import further terminology definition (Ontologies or OO Mediators) needed for defining the reduction. Here, no additional terminology is needed. Note that the terminology definitions of the source- and target component is inherited into the Mediator - which goes for all types of WSMO Mediators.

Listing 11: GG Mediator that restricts the generic Goal

```

namespace <<http://www.wsmo.org/2004/d3/d3.2/v0.1/20041004/resources/ggm1>>
  dc:<<http://purl.org/dc/elements/1.1#>>
  wsm1:<<http://www.wsmo.org/2004/d2/#>>

ggMediator <<http://www.wsmo.org/2004/d3/d3.2/v0.1/20041004/resources/ggm1.wsm1>>
  nonFunctionalProperties
    dc:title hasValue "GG Mediator that links the general Goal for buying tickets with the concrete Goal for buying a train ticket fr
    dc:creator hasValue <<http://www.deri.org/foaf#deri>>
    dc:subject hasValues {"Tickets", "Online Ticket Booking", "train trip"}
    dc:description hasValue "Restricts the trip to train trips within Austria and Germany"
    dc:publisher hasValue <<http://www.deri.org/foaf#deri>>
    dc:contributor hasValue <<http://www.deri.org/foaf#stollberg>>
    dc:date hasValue "2004-10-04"
    dc:type hasValue <<http://www.wsmo.org/2004/d2/#ggMediator>>
    dc:format hasValue "text/html"
    dc:identifier hasValue <<http://www.wsmo.org/2004/d3/d3.2/v0.1/20041004/resources/ggm1.wsm1>>
    dc:language hasValue "en-us"
    dc:relation hasValues {<<http://www.wsmo.org/2004/d3/d3.2/v0.1/20041004/resources/goal1.wsm1>>,
      <<http://www.wsmo.org/2004/d3/d3.2/v0.1/20041004/resources/goal.wsm1>>}
    dc:coverage hasValue "ID:7029392 Name:World"
    dc:rights hasValue <<http://deri.at/privacy.html>>
    version hasValue "$Revision: 1.2 $"
  endNonFunctionalProperties

  source <<http://www.wsmo.org/2004/d3/d3.2/v0.1/20041004/resources/goal1.wsm1>>

  target <<http://www.wsmo.org/2004/d3/d3.2/v0.1/20041004/resources/goal.wsm1>>

```

### 3.4.4 WW Mediators

A WW Mediator connects Web Services used by another Web Service in their Orchestration, resolving heterogeneities at all levels (data, process, protocol). Also, WW Mediators are applied to resolve heterogeneities on the data, process, and protocol level between the Choreographies of Web Services that are ought to interact in a global interaction model.

There is no WW Mediator in this use case at this point in time, since the Orchestration of the ÖBB end-user Web Service is not specified in this version. 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

### THIS PART NEEDS TO BE REWORKED, ACCORDING TO WSMO D5.1

Web Service Discovery is concerned with inference-based mechanisms that detect suitable Web Service for a given Goal. This means that the discovery mechanism inspects available Web Service descriptions and determines whether these can be used to fulfill a certain Goal. The overall structure of WSMO supports Web Service discovery explicitly by introducing the notions of Goals and Web Services as top level building blocks. The requirements and the approach for Web Service Discovery in WSMO is exhaustively discussed in [\[Keller et al., 2004\]](#). Here, we shortly summarize the most important aspects and explain the realization of Web Service Discovery within FLORA-2 on basis of the use case models as specified above.

In general, Web Service Discovery is separated into three major aspects:

#### 1. The Core: Goal-Capability-Matching

Goal-Capability matching determines whether the Capability of a Web Service Description can satisfy the given Goal, i.e. if the Web Service can be used for solving the Goal. Therefore, it basically has to be proven that the Capability logically entails the Goal with the premise that the conditions for successful usage of the Web Service are fulfilled at invocation time.

Goal-Capability-Matching is considered as the heart of Web Service Discovery, as it

determines whether some Web Service can satisfy the Goal at all. Upon this, different "Discoverers" can be build that modify the discovery results with regard to specific requirements for the application. We discuss this in more detail below.

## 2. Handling partial matches in Goal-Capability Matching

Goal-Capability matching is only successful, i.e. it returns a set of suitable Web Services to solve a given Goal, if the Goal and the Web Service Capability match perfectly. This means that for all states where the postconditions and effects of the Web Service are fulfilled the Goal is satisfied. This might not hold for many cases, as there might be semantic differences between the Goal and the Capability; but a Web Service might be usable for solving Goal when the valid set of outputs (as well as the set of halting states) of the Web Service is restricted – also vice versa, i.e. the Web Service can be used to solve a Goal within certain restrictions on the Goal.

In WSMO, these differences are explicitly stated in a WG Mediator. This restricts the valid values between a Goal and a Web Service Capability, thereby ensures Goal-Capability-Matching between Goals and Capabilities that only match partly, and thus broadens the set of possible usable Web Services for solving a Goal. For determining the required reduction in a WG-Mediator, specific heuristics can be applied.

## 3. Filter mechanisms for improving discovery results

Goal-Capability matching returns a set of suitable Web Services for solving a Goal. In order to improve the quality of the results set, additional filter mechanism can be applied. These can be based on the non-functional properties of Web Services, or can take some preferences of the customer into account.

The theory of Web Service Discovery is exhaustively discussed in [\[Keller et al., 2004\]](#). Therein, the general proof obligation is defined, and different approaches by means of logic programming are presented as discussed in detail. For Web Service Discovery in this use case, we implemented the approach "Goal as ground facts", see section 4.2.1 in [\[Keller et al., 2004\]](#). Summarizing, the matchmaking works as follows:

- The Goal is defined as a ground fact, meaning it is a logical expression without variables. The Goal is only specified partially, meaning that concrete values are only specified for the subset of the properties of the ontological concepts that is needed to express the desire; e.g., for the itinerary that a ticket is desired for, no information is specified for 'arrival' (see Listing 5) .
- The description elements of the Web Service Capability are modeled as rules. Here also, only the subset of properties of the ontological notions that is needed to describe the information structure and conditions is specified; besides, the body of each rule contains conditions that restrict the range of valid values.

The approach for matchmaking is to check whether the ground facts specified in the Goal satisfy the rules of the Capability. Obviously, this is facilitated by the construction of Goals and Capabilities. Considering the Goal as specified in Listing 5, the Goal postcondition is fact that satisfies the body of the Capability postcondition in Listing 6: the Goal Postcondition is a specific itinerary (from Innsbruck to Frankfurt with a certain departure), and the body of the Capability postcondition requires an itinerary with start- and endlocation in Austria or Germany, respectively, and with a departure that has to be later than the current date (the current date is modeled as an instance of dateandtime from the Date and Time Ontology as there is no built-in function yet). The same matching holds for the Goal Effect in correlation to the Capability Effect. Thus, the following discovery query returns the identifier of the ÖBB end-user service, as a service that can satisfy the Goal. The result of successful discovery in Flora is shown in the screenshot in Figure 7.

```
?- Webservice:webservice[
  capability->_C:capability[postcondition, effect]
].
```

```

/cygdrive/c/_projects/_WSMO/_cvs/wsmo/d3/d32/resources
[FLORA: Compiling c:\_projects\_WSMO\_cvs\wsmo\d3\d32\resources\match.flr]
[Preprocessing c:\_projects\_WSMO\_cvs\wsmo\d3\d32\resources\match.flr]
[FLORA: Compile time 0.2800 seconds]
[FLORA: Dependency checking time: 0.1100 seconds]
[FLORA: [Coder] Generating P file 0.0200 seconds]
[FLORA: [Coder] Generating DB file 0.0100 seconds]
[FLORA: Done! CPU time used: 0.4500 seconds]
[FLORA: Loading c:\_projects\_WSMO\_cvs\wsmo\d3\d32\resources\match.flr into mod
ule main]
[Compiling c:\_projects\_WSMO\_cvs\wsmo\d3\d32\resources\match]
[Preprocessing c:\_projects\_WSMO\_cvs\wsmo\d3\d32\resources\match.P]
[match compiled, cpu time used: 2.5640 seconds]
[match_main loaded]
[FLORA: Dynamically loading c:\_projects\_WSMO\_cvs\wsmo\d3\d32\resources\match.
fdb into module main]
[Preprocessing c:\_projects\_WSMO\_cvs\wsmo\d3\d32\resources\match.fdb]
[FLORA: Done! CPU time used: 0.0200 seconds]
[FLORA: Dynamically loading c:\_projects\_WSMO\_cvs\wsmo\d3\d32\resources\match.
fld into module main]
[Preprocessing c:\_projects\_WSMO\_cvs\wsmo\d3\d32\resources\match.fld]
[FLORA: Done! CPU time used: 0.0000 seconds]

Testing if the KB is inconsistent:
No

Querying for Web Services for the goal in goal.flr:
Webservice = oebbWS

1 solution(s) in 0.0100 seconds on NB-HL

Yes

Yes

flora2 ?-

```

Figure 7. Successful Web Service Discovery in FLORA-2

The transformation of the WSMO models as specified in the Listings to FLORA-2 compilable F-Logic is trivial, as WSML relies on F-Logic. The complete models for the use case as FLORA-compilable resources is provided in [Appendix A](#). However, there are some aspects that have to be considered for the transformation of WSML definitions to FLORA-2 syntax:

1. **Signature Checking** Class Signatures (single and multi-valued attributes only) are checked via a set of rules, that can be found in the source file flr.flr in the appendix. Note that there is yet no agreed syntax for f-logic to define the signatures of relation or method parameters, therefore we currently do not check those signatures. We refer to the ongoing discussion in the f-logic syntax consortium recently established by Ontoprise, Michael Kifer and DERI.
2. **Integrity Constraints** In WSML they are modelled as rules with empty head, flora does not have a build in support to check them, therefore we use the two valued relation `invalid` in the head of the rule, the first parameter is the instance that is violating some constraint, the second is a String that identifies the constraint which is violated. The integrity of the knowledge base is checked the query `"?-invalid(X,Y)"`, only if this query returns no values the knowledge base has a model.
3. **Mediators** are not implemented yet, all flora files are simply imported in one single file and then evaluated.
4. **Namespaces** are not implemented.
5. **Non Functional Properties** are not translated directly into the ontologies represented in flora2 syntax. They are only present as inline comments.
6. **Evaluation of Arithmetic Expressions** is not always done in flora2 unless the "is" operator is used, therefore some "=" in the WSML listings have been changes to "is".
7. **Function Symbols** have not been defined directly as function symbols in the flora files, but rewritten as relations with an arity of n+1 (w.r.t. arity of the function), see

secondsAfterMidnight relation in the dt.flr for an example.

8. **Precision of Floating Point Operations** Due to the imprecise handling of floating point arithmetics in the XSB engine underlying flora2, some axiomatizations given in the wsml file have been implemented differently. See for example the julianDayNumber function.
9. **Variable Declarations** have been omitted into the flora2 files, a variable is simply identified as an upper case term.
10. **Redefinition of Build-In Predicates** is not allowed in flora2, therefore new relations are defined, for example equal(X,Y) which is equivalent to X=Y in the wsml files.

In conclusion, the approach for Goal-Capability-Matching presented here seems to be a promising solution because it is heading towards the right direction. For further development of the Goal-Capability-Matching technology within WSMO as the heart of Web Service Discovery, very complex and challenging efforts have to be faced, e.g. definition of a decidable subset for the specifications of logical expression in WSML as well as implementation of logical entailment for the reasoner to be supported 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, according to the application scenario for Semantic Web Services described in the [WSMO Use Case Overview document, section 2.1](#). 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 Cases are gathered in the [WSMO Use Case Overview document](#).

## References

**[de Bruijn, 2004]** de Bruijn, J. (ed): *D16.0 v0.2 WSMO The WSML Family of Representation Languages ebXML*, WMSL Working Draft 26 September 2004, available at: <http://www.wsmo.org/2004/d16/v0.2/20040926/>.

**[ebXML]** ebXML, available at: <http://www.ebxml.org>.

**[EDIFACT]** UN/EDIFACT, available at: <http://www.unece.org/trade/untdid/welcome.htm>.

**[Fensel & Bussler, 2002]** D. Fensel and C. Bussler: *The Web Service Modeling Framework WSMF*, Electronic Commerce Research and Applications, 1(2), 2002.

**[Keller et al., 2004]** Keller, U.; Lara, R.; Polleres, A. (eds.): *WSMO Discovery*, D5.1 v0.1, WMSL Working Draft 13 September 2004. available at <http://www.wsmo.org/2004/d5/d5.1/v0.1/20040913/>

**[Pan and Hobbs]** F. Pan and R. Hobbs: *Time in OWL-S*, available at: <http://www.isi.edu/~pan/damlttime/AAAsymp2004.pdf>.

**[Roman et al., 2004]** D. Roman, U. Keller, H. Lausen (eds.): *Web Service Modeling Ontology (WSMO)*, WSMO Working Draft D2, final version 1.0, 20 September 2004, available at <http://www.wsmo.org/2004/d2/v1.0/20040920/>.

**[RosettaNet]** RosettaNet, available at: <http://www.rosettanet.org>.

## Acknowledgements

The work is funded by the European Commission under the projects [DIP](#), [Knowledge Web](#), [SEKT](#), [SWWS](#), [Esperanto](#), and [h-TechSight](#); by [Science Foundation Ireland](#) under the [DERI-Lion](#) project; and by the Vienna city government under the [CoOperate program](#).

The editors would like to thank to all the [members of the WSMO working group](#) for their advice and input into this document.

---

## Appendix: Change Tracking

To facilitate retracing of changes inbetween 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.

**Version: 04 October 2004** <http://www.wsmo.org/2004/d3/d3.2/b2c/20041004/>

- 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

**Version: 19 July 2004** <http://www.wsmo.org/2004/d3/d3.2/v0.1/20040719/>

- 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)

**Version: 28 June 2004** <http://www.wsmo.org/2004/d3/d3.2/v0.1/20040628/>

- 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

---



\$Date

webmaster