Abstract

This deliverable addresses the description of non-functional properties for Web services. It provides a survey of solutions to this problem for Web services in particular and Software Systems in general. Furthermore, it takes the most relevant non-functional properties models adopted from [O’Sullivan et al., 2005] and provides possible solutions for specifying them in a formal manner, using WSMO/WSML. These solutions can be seen as possible extensions of non-functional properties description support in WSMO/WSML.
Contents

1 Introduction ................................................. 5

2 Survey of Non-functional properties approaches ................. 7
   2.1 NFP and Web services ..................................... 7
      2.1.1 UDDI .................................................. 7
      2.1.2 OWL-S ............................................... 7
      2.1.3 SWSF ................................................ 7
      2.1.4 WSMO/WSML ........................................... 8
      2.1.5 O’Sullivan approach ................................... 8
   2.2 NFP and Software Systems .................................. 9
      2.2.1 Process NFL ............................................ 10
      2.2.2 CQML ................................................ 11
      2.2.3 QML .................................................. 12
      2.2.4 QuO .................................................. 13

3 Non-functional properties tasks .................................. 15

4 Non-functional properties in WSMO/WSML ...................... 17
   4.1 Non-functional properties types ............................. 17
   4.2 Modeling non-functional properties .......................... 18
   4.3 Attaching non-functional properties to WSMO elements ... 18
      4.3.1 Model .................................................. 19
      4.3.2 Syntax ............................................... 20
      4.3.3 Example .............................................. 20
   4.4 Policy-centered NFP meta-model .............................. 21
      4.4.1 The NFP meta-model ................................... 22
   4.5 Using the NFP meta-model in WSMO ......................... 26
      4.5.1 Attaching NFP policies to WSMO elements .............. 26
      4.5.2 Example .............................................. 26
         4.5.2.1 Example of Web Service Description .............. 27
         4.5.2.2 Example of Goal Description ...................... 28
   4.6 Embedding the NFP meta-model in WSMO ..................... 29
      4.6.1 Attaching NFP policies to WSMO elements .............. 30
      4.6.2 Model ................................................ 30
      4.6.3 Syntax ............................................... 31
      4.6.4 Example .............................................. 32
         4.6.4.1 Example of Web Service Description .............. 33
         4.6.4.2 Example of Goal Description ...................... 33

5 Annotations .................................................. 35

6 Ontologies for specifying non-functional properties .......... 36

7 Conclusions and Future Work .................................. 38

A Appendix A .................................................. 39

B Appendix B .................................................. 76

C Appendix C .................................................. 79

D Appendix D .................................................. 82
1 Introduction

Built on current Web services technologies, like WSDL [Christensen et al., 2001], SOAP [W3C, 2003] and UDDI [Bellwood et al., 2002], Semantic Web services are providing a new level of automation for service related tasks like: discovery, composition, selection, negotiation, invocation, etc. Both technologies, Web services as well as Semantic Web services, consider services as fundamental, central entities. How these services are described is crucial for the successful realization of all previous mentioned service related tasks.

Three different aspects must be considered when talking about services: (1) functional, (2) behavioral and (3) non-functional. The functional description contains the formal specification of what exactly the service can do. The behavioral description is about how the functionality of the service can be achieved in terms of interaction with the service (choreography in WSMO) and as well in terms of functionality required from the other Web services (orchestration in WSMO). Finally, the non-functional descriptions captures constraints over the previous two [Chung, 1991]. For example, in case of a train booking service, invoking its functionality (booking a train ticket) might be constrained by using a secure connection (security as non-functional property) or by actually performing the invocation of the services in a certain point in time (temporal availability as non-functional property).

Among the three aspects of a service description, the functional, behavioral aspects are the most investigated aspects. Although the third aspect, non-functional properties, did not capture a very broad attention from the Web service research community, as functional and behavioral descriptions did, one has to recognize the big importance of describing them. This is due to their high relevance for all service related tasks.

Non-functional properties might play an important role in all service related tasks, especially in discovery, selection and substitution of services. It is simple to imagine a scenario in which services which can fulfil a user request and which provide basically the same functionality are selected based on some non-functional properties like price or performance.

The lack of real support (languages, methodologies, tools) for non-functional properties might be due to various factors (Rosa et al., 2002, Eenoo et al., 2005):

- non-functional properties are usually too abstract and most of the time they are stated informally.
- in most of the cases there is no clear delimitation between the functional and non-functional aspects of a service.
- non-functional properties are often considered to be represented after the functional and behavioral have been described.
- non-functional properties very often conflict and compete with each other (e.g. availability and performance).
- complexity of modelling non-functional properties - difficult to formalize.

In this document we investigate some existing solutions for describing non-functional properties of Web services in particular, and Software Systems in general, and we sketch some possible solutions for describing in a more useful manner non-functional properties. We do this by using semantic technologies, more precisely the Web Service Modeling Language (WSML) [de Bruijn et al., 2005] as a formal language to write down the non-functional properties specifications. A semantic approach for describing non-functional properties will basically enable reasoning on these properties, thus supporting the automation of tasks like discovery, selection, composition, etc.
The document is structured as follows: Section 2 provides an overview of some of the existing solutions for describing non-functional properties of Web services in particular and Software Systems in general. Our survey is two fold: on one hand we take a look at conceptual models for non-functional properties and on the other hand we survey languages used to describe them. Section 3 discusses the tasks for which non-functional properties should be used and how they can be used. Section 4 distinguishes between two types of non-functional properties: annotations and service related non-functional properties called non-functional properties (subsection 4.1). Discusses the modeling of non-functional properties by means of ontologies (subsection 4.2) and proposes a simple model to attach non-functional properties to WSMO services and goals (subsection 4.3). Furthermore, a non-functional properties upper-level ontology or meta-model, used to group and structure offered and requested non-functional properties is proposed in subsection 4.4. Two different approaches to integrate the meta-model, one light approach without any changes to the WSML syntax (subsection 4.5) and one embedded approach which implies changes in the WSML syntax (subsection 4.6), are proposed.

Section 5 discusses one of the two types of non-functional properties we have identified namely annotations and Section 6 introduces a set of ontologies for specifying service related non-functional properties. A WSML formalization of these ontologies is available in Appendix A. Finally, in Section 7 we conclude and present our future work.
2 Survey of Non-functional properties approaches

In this section we take a look at some of the most relevant approaches for describing non-functional properties of Web services and Software Components. The intention is to give an overview of: (1) what is the set of relevant non-functional properties considered by each approach, (2) how they are modelled and (3) how they are formally represented. Besides the approaches presented here, other approaches might be analyzed and included in this survey in the future versions of this document.

2.1 NFP and Web services

This section gives an overview of the support provided by different Web services and Semantic Web services approaches in describing the non-functional properties of a service.

2.1.1 UDDI

The Universal Description, Discovery and Integration(UDDI) [Bellwood, 2002] defines a set of non-functional properties for a service provider identified by a businessEntity. The set of non-functional properties contains: the address, the phone numbers, and the email addresses of the service provider. Additionally to non-functional properties some other information (metadata) about the service is available like for example the service category. Schemas like UNSPSC [UNS, 2000] can be used for this purpose. This extra information falls rather in the functional description of the service. The set of non-functional properties and not only can be accessed using the set of UDDI APIs.

2.1.2 OWL-S

The OWL-S [The OWL Services Coalition, 2004] approach consider the following set of non-functional properties: service name, text description, quality rating. Other non-functional properties can be included by using the ServiceParameter from ServiceProfile. These non-functional properties are described in the Service Profile part and explicitly formalized using OWL [McGuinness and van Harmelen, 2004].

2.1.3 SWSF

In Semantic Web Service Ontology(SWSO) [SWSL, 2005c], part of Semantic Web Services Framework(SWSF) [SWSL, 2005a] approach, Service Descriptors are used to provide information about non-functional aspects and/or provenance aspects of a service. The following non-functional properties are considered: service name, service author, service contact information, service contributor, service description, service URL, service identifier, service version, service release date, service language, service trust, service subject, service reliability and service cost. SWSO does not provide a
model for the non-functional properties of a service. The non-functional properties can be described using relations such as name(service, service\_name). These relations can be expressed using SWSL-FOL [SWSL, 2005b] or SWSL-Rules [SWSL, 2005b].

### 2.1.4 WSMO/WSML

The Web Service Modeling Ontology (WSMO) [Roman et al., 2005] approach recommends a set of non-functional properties for each particular element of a Web service description. For example, the service recommended non-functional properties are: accuracy, contributor, coverage, creator, date, description, financial, format, identifier, language, network-related QoS, owner, performance, publisher, relation, reliability, rights, robustness, scalability, security, source, subject, title, transactional, trust, type, version. The common set of non-functional properties for all elements in WSMO is provided by Dublin Core Metadata Initiative [Weibel et al., 1998]. WSMO does not provide a model for the non-functional properties of a service. Using WSML [de Bruijn et al., 2005], the Web Service Modeling Language, one can assign values to the non-functional properties of a WSMO element. Such a value can be any identifier and thus it can be an IRI, a data value, an anonymous identifier or a list of the former.

A closer look at non-functional properties used in WSML shows that many of them are used to express information about the description itself and not about the service. This point of view is different from the definition of non-functional properties we adopt in this document. For example, the non-functional properties contributor, creator, date, identifier, owner, publisher, subject, title, and version are not constraints on either what a service can do nor how it can do it. They are rather adding extra information about the description.

Another aspect of current approach for non-functional properties in WSML is that these properties are not included in the logical model and thus reasoning on them is not possible. Future versions of WSML might consider this aspect.

### 2.1.5 O’Sullivan approach

In [O’Sullivan et al., 2005] a set of the most relevant non-functional properties for Web services and their modelling are described. This set could be a very good starting point towards a semantically enabled solution for non-functional properties of Web services. The following concepts were identified:

1. **Service Provider**
   The Service Provider model captures information about: the service identifier which can be a UN/SPSC code, the service name and the provider of the service

2. **Temporal Model**
   The Temporal Model provides the temporal concepts that are needed for time related descriptions of a service. These are: Temporal Date, Time, Temporal Interval and Temporal Duration. These concepts can further be refined in more specific concepts like Calendar Date for example.

3. **Locative Model**
   The Locative Model is used to model the location of a service. Concepts like: Address, Region, Route, Point, Street Directory Reference, PhoneNumber, URI, IPAddress and Spectrum are directly related to this model.

4. **Service availability**
   The Service availability combines temporal and locative aspects of the service to describe when and where one can interact with the service.
5. **Obligations**  
The *Obligations* model captures the responsibilities of both service requestor and service provider. Three kinds of obligations were defined: *Pricing obligations, Payment obligations and Relationship obligations.*

6. **Price**  
*Price* and *Payment* are seen as complementary non-functional properties. They represent two views of the same thing but from different perspectives. The payment(cost) is the user’s perspective and the price is the provider’s perspective.

7. **Payment**  
The *Payment* model captures the manner in which a service requestor can fulfil their payment obligations. As stated before, payment and price are complementary.

8. **Discounts**  
Closely related to the notions of price and payment is the notion of *Discount*. Discounts are viewed from the service requestor perspective and are categorized according to the payment method and requestor’s identity.

9. **Penalties**  
The *Penalties* are used by a service provider to specify what exactly will occur if a service requestor does not comply with a specific obligation. The same should hold the other way around. Penalties should be described by both the service provider and service requester and should apply to both.

10. **Rights**  
The *Rights* model captures the permissions granted to service providers and service requestors to perform operations.

11. **Language**  
Three kinds of *Language* support for describing a service are distinguished: *written language* (e.g. English), *spoken language* (e.g. English) and *standard language* (e.g. WSDL).

12. **Trust**  
*Trust* is a notion understood in various ways by different people. The Trust model is directly influenced by other models like endorsement and service inception.

13. **Quality**  
*Quality* is described relative to a standard, an industrial benchmark and/or a ranking schema.

14. **Security**  
The *Security* model is attached to the locative aspect of the service and it’s divided in two dimensions: identification and confidentiality.

These non-functional properties are conceptually defined using ORM Object Role Modelling (ORM) [Halpin, 2001]. ORM is a fact-oriented modelling technique that make no use of attributes and which represents facts in the form of entities playing roles.

### 2.2 NFP and Software Systems

In this section we are going to briefly survey some approaches for specifying non-functional properties for a software system, from which Web services can benefit. The
focus is on the language used to specify the non-functional properties. For the interested reader surveys and comparisons of approaches for specifying non-functional properties in general and quality of service in particular in software systems area can be found in [Aagedal, 2001] (chapter 4.4) and [Jin and Nahrstedt, 2002].

2.2.1 Process NFL

In [Rosa et al., 2002] a language for non-functional properties called Process\(^{NFL}\) is proposed. The language has been designed to consider some specific aspects when modelling non-functional properties like: correlations and conflicts between non-functional properties. An example of non-functional properties that might be in conflict are: security and performance. A high security may probably determine a loss in performance and vice versa. Three abstractions are used to model a non-functional property. These are: NF-Attribute, NF-Property and NF-Action.

NF-Attributes model non-functional characteristics that can be precisely measured (e.g. performance), non-functional characteristics that can not be quantified (e.g. security) and non-functional characteristics that simply can exist or not for a system (e.g. transactional properties like atomicity, consistency, etc.). NF-Attributes can be derived from other NF-Attributes and thus a class hierarchy of NF-Attributes can be created. The root of this hierarchy in the Non-Functional Requirement - NFR class. Process\(^{NFL}\) takes rather an object oriented approach when defining NF-Attributes. A NF-Attribute definition can be seen as a class definition that might have other NF-Attributes as members. The user has the possibility to specify in a NF-Attribute definition the necessity to have the NF-attributes members defined or not (ie. all, any, none, exactly one).

NF-Actions model characteristics that affect the NF-Attributes. These characteristics can be software or hardware characteristics. An example of NF-Action is the encryption algorithm which has a major influence on the security NF-Attribute.

NF-Properties model constraints over the NF-Attributes. These are expressed in terms of different levels for a NF-Attribute. For example the strong performance is a NF-Property which constraints the performance NF-Attribute. The following levels are considered: strong, medium, low.

Templates are used to model all the abstractions in Process\(^{NFL}\). NF-Attributes have for example the following structure:

```plaintext
attribute attributeId1 extends attributeID2{
    primitives primitiveAttributes;
    contribution kindOfContribution;
}
```

Using this template the non-functional property performance can be defined as follows:

```plaintext
attribute performance extends NFR{
    primitives space_performance, time_performance;
    contribution oneX;
}
```

Summary: Process\(^{NFL}\) is a general purpose non-functional properties specification language. It has some nice features like the possibility to express correlations and conflicts between non-functional properties and the possibility to express the compositional aspect of non-functional properties and strengths. However one major drawback of Process\(^{NFL}\) is that it has no formal semantics.
2.2.2 CQML

In [Aagedal and Earl F. Ecklund, 2002] and [Aagedal, 2001] a modeling language for Quality of Services (QoS) [ISO, 1986] called Component Quality Modelling Language is proposed. The following abstractions are used to model the QoS: QoS Characteristic, QoS Statement, QoS Profile and QoS Categories.

QoS Characteristics can be seen as user-defined types. They are the building blocks of QoS specifications. For example delay, which is an important QoS Characteristic for a real time service (e.g. video streaming service) can be modelled as follows:

```plaintext
def quality characteristic delay {
    domain: decreasing numeric milliseconds;
}
```

The general template for describing QoS Characteristics is provided in the listing below:

```plaintext
def quality characteristic <characteristic_name> {
    domain <domain_type>;
}
```

A domain type can be either numeric, set or enumeration. The keyword domain is used to specify the range of possible values that the quality characteristic might have.

One can specify how characteristics influence the quality of the service. In the previous example the characteristic delay is decreasing the quality. QoS Characteristics can be derived from other characteristics (e.g. statisticalDelay:delay).

QoS Statements are used to specify restrictions of the values of QoS Characteristics. Taking the previous example a low delay can be defined as a delay less than four using CQML as:

```plaintext
def quality low delay {
    delay < 4;
}
```

QoS Profiles are used to bind QoS statements with component specifications. In the example the QoS statement low delay is bound to a component specification called component specification:

```plaintext
def profile goodComponent for component specification {
    provides low delay;
}
```

All the previous mentioned abstractions: QoS Characteristics, QoS Statements and QoS Profiles can be grouped in QoS Categories. In the example below two QoS Characteristics: delay and output are grouped together:

```plaintext
def quality category timelines {
    delay;
    output;
}
```

The output in the previous listing is another QoS Characteristic that is used to model the quality of service for a real time service, like a video streaming service. This QoS Characteristic can be further specialized to output frequency which represents the frequency of the video streaming service output.

**Summary:** CQML is a language for specifying QoS at different levels of abstraction. It can be easily integrated with UML. CQML provides the means for precise specification of QoS characteristics and QoS measurements. It allows QoS characteristics to be refined and also to be aggregated. However, as pointed out in [Röttger and Zschaler, 2003], CQML has some drawbacks. For example CQML is only based on QoS-relations specification between components and there is no possibility to specify the demand on resources for components. Also dependencies between QoS characteristics can not be easily specified.
2.2.3 QML

In [Frølund and Koistinen, 1998], a QoS specification language for distributed object systems is presented. This language, called Quality of service Quality Language (QML), is not restricted to any particular domain (e.g. real-time or multimedia systems) or to any particular quality of service. QML provides the following main abstractions for modeling QoS: contract type, contract and profile.

A contract type represents a specific QoS category such as performance or security. There are no predefined contract types in QML. Users can define their own contract types by specifying a set of constraints over a number of QoS dimensions. A QoS dimension characterizes a QoS category and has a domain of values. There are three kinds of domains: set domains, enumerated domains and numeric domains. An example of contract type for performance is given below.

```
type Performance = contract{
    delay: decreasing numeric msec;
    throughput: increasing numeric MB/sec;
}
```

This performance contract type contains two dimensions, delay and throughput that have both a numeric domain. Just like CQML, QML provides keywords ("decreasing" and "increasing") that allow users to specify how dimensions or characteristics influence a QoS. In the example the keyword "decreasing" indicates that a small delay is better than a large one. The general template for a contract type is provided in the listing below:

```
type <contract_type_name> = contract{
    <dimensionName_1>: <dimensionType_1>;
    ...
    <dimensionName_n>: <dimensionType_n>;
}
```

A `dimensionType` contains information about the domain type (set, enumerated, numeric), order constraints if needed, the unit measure and relations like increasing/decreasing.

Using contract types, one can define concrete contracts. A contract in QML is an instance of a contract type and represents a particular QoS specification. Given the performance contract type from the previous example a contract can be defined as follows:

```
systemPerformance = Performance contract{
    delay < 10 msec;
    throughput > 10 MB/sec;
}
```

Statistical properties can be attached to QoS dimensions. This can be done by defining aspects of QoS dimensions. QML includes four generally applicable aspects: percentile, mean, variance and frequency. To the previous example we can attach two percentile aspects to delay dimension as follows:

```
systemPerformance = Performance contract{
    delay {
        percentile 60 < 5 msec;
        percentile 100 < 10 msec;
    };
    throughput > 10 MB/sec;
}
```

Thus, from a set of delay measurements sorted by the ascending value of their values the first 60% of the measurements should have a value lower then 5 msec all of them should have a value lower then 10 msec.

Contracts can be refined using the `refined by` construct. If $A$ and $B$ are two contracts than we can specify that $A$ is a refinement of $B$ as follows "$A = B$ refined by ...". There
are some restrictions with respect to refinement: (1) the two contracts must have the same contract type and (2) the new contract, specified as a refinement, must introduce stronger constraints that in the initial contract or new constraints. For example, one can refine the previous defined systemPerformance contract as follows:

```
refinementSystemPerformance = systemPerformance refined by {
    delay < 5 msec;
    throughput > 15 MB/sec
}
```

The third main abstraction in QML is the profile. A profile describes the associations between contracts and interface elements for a particular interface. It describes a set of requirements, each of which specifies one or more contracts for one or more interface entities. If a requirement from profile is not associated with an interface entity then that requirement is called default requirement and applies to all entities from the interface. Multiple profiles can be defined for an interface. Let’s consider a service, myService, that has an interface, myServiceInterface, with two operations: myOperation1 and myOperation2. This interface is described in CORBA IDL. A profile associated with this interface might have the following specification:

```
myServiceProfile for myServiceInterface = profile {
    require systemSecurity;
    from myOperation1 require Performance contract {
        delay < 100 msec;
    }
}
```

The profile myServiceProfile associates contracts with entities in the myServiceInterface interface. The first requirement clause states that the service should satisfy a systemSecurity contract, defined in a similar manner as with the systemPerformance contract. Since the clause is not associated with a particular operation from the interface, this clause applies to both operations and it is called default clause or default contract. The second requirement clause refers only to myOperation1 operation. Clause that are not default clauses must be stronger than the default clause/contract. Profiles can be refined in a similar manner as contracts.

Summary: QML is a general purpose QoS specification language. It is not restricted to any particular domain or to any particular quality of service. QML separates specification of QoS aspects from functional specification aspects. A useful mechanism in QML is the refinement mechanism that allows QoS aspects to be defined as refinements of existing ones. However QML has an important drawback. It only covers the design time aspects of QoS, but does not address the problem of what actions to take at runtime if the QoS requirements can not be satisfied.

2.2.4 QuO

In [Zinky et al., 1997] and [Loyall et al., 1998], the Quality Objects framework (QuO) which supports the development of QoS enabled distributed applications is presented. As part of QuO framework a set of QoS description languages called QDL languages were proposed. The QDL suite contains a Contract Description Language (CDL), a Structure Description Language (SDL) and a Resource Description Language (RDL). However, only the Contract Description Language is entirely specified.

The CDL language is used for specifying QoS contracts between clients and CORBA objects. Such a contract specifies the level of service desired by the client, the level of service the object expects to provide, actions to take when the QoS level changes and what QoS characteristics are observable. SDL describes the internal structure of remote

---

1 http://www.omg.org/corba/
objects implementations and how they consume resources. Finally, RDL abstracts the physical resource used by the objects.

The CDL language has four major modeling abstractions: nested regions each representing a possible state of a QoS, transitions for each level of regions specifying behavior to trigger when the active region changes, system controlling objects for measuring and controlling QoS and callbacks for notifying the client or object. Within a nested region two different levels of region exists: negotiated regions or reality regions. Negotiated regions define the system conditions within the client and the service will try to operate. Within a negotiated region there may be many reality regions that are defined in terms of measured system conditions. A contract example in CDL is provided in the following listing:

```plaintext
contract repl_contract{
  negotiated regions are
  region Low_Cost : when ClientExpectedReplicas == 1 =>
    reality regions are
    region Low : when MeasuredNumberReplicas < 1 =>
    region High : when MeasuredNumberReplicas >= 1 =>
    transitions are
      transition High => Low : ClientCallback.availability_degraded();
    end transitions;
    end reality regions;
  region Available : when ClientExpectedReplicas >= 2 =>
    reality regions are
    ... transitions are
    ... end transitions;
  end reality regions;
  end negotiated regions;
  end repl_contract;
}
```

The previous example shows a client contract that can operate in two possible operating modes: Low_Cost and Available. These two modes are represented as negotiated regions. The decision on which operating mode is active depends on ClientExpectedReplicas that is a system condition object. It indicates the client expected values for a particular aspect, provided replicas, in this case. Another system condition object called MeasuredNumberReplicas is used to discriminate between reality regions. In case of Low_Cost negotiated region, the reality regions are Low and High. The reality regions are used to notify the user about changes in the number of replicas. If there is a decrease of replicas during measurement then a callback availability_degraded() will be triggered.

Using SDL language, different behaviors to invoke for method calls/returns based on the current regions of contracts can be specified. From QDL specifications (CDL and SDL), the QuO code generator generates code that helps application developers in making their application QoS-aware.

Summary: The QuO framework with its QDL languages suite (CDL, SDL and RDL) provides support for QoS specifications of distributed applications. QuO is very much a contract-based approach, specifying what actions must be taken in case QoS requirements are not satisfied. QuO/QDL has a set of drawbacks. For example there is no special facility for specification reuse. Specification of contract types is not supported. Also QuO mixes declarative specifications of QoS with implementation specification which makes it quite complex.
3 Non-functional properties tasks

As mentioned in Section 1 three different types of properties must be considered when talking about services: (1) functional, (2) behavioral and (3) non-functional. In this section we discuss the last type of properties. More precisely we investigate what are the tasks for which non-functional properties might be relevant.

It is important to introduce briefly what types of non-functional properties exist since the tasks that can be performed with these properties depend on them. We distinguish between two classes of non-functional properties: (1) annotations - which provide metadata about any type of element description (service, goal, ontology, etc.) and (2) non-functional properties - the set of properties strictly belonging to a service other then functional and behavioral. A detailed discussion on what types of non-functional properties exist is provided in Section 4.1.

The set of tasks relevant for the first category of non-functional properties, namely annotations are:

1. Discovery
   Although discovery based on annotations might not be relevant for the discovery process from the end user perspective (it is unlikely that a user will ask for service descriptions that were created at a specific date), it might be however relevant in the preparatory steps of discovery process. For example a discovery engine could first collect the latest version of service description available online.

2. Selection
   Considering the same settings described above, selection based on annotations might be relevant task as well.

3. Versioning
   The version property from a service description or any of its elements is intended to be used and should be used to keep track of different versions of description.

The set of tasks relevant for the second category of non-functional properties strictly related to a service which we call simply non-functional properties are:

1. Discovery
   Discovery is a process for which non-functional properties of a service might be quite relevant. This process depends on how services are modelled and implicitly on their descriptions [Keller et al., 2004]. As discussed in Section 1 a service description has three distinct aspects: functional, behavioral and non-functional. All these descriptions might be considered when a user request is matched against services. However in most cases, discovery based on non-functional properties is performed after functional-based is performed. For example on a set of discovered services that offer online television new matching process can be performed by considering constraints on non-functional properties specified by the requestor (e.g. delay should be lower than a specified value, etc.)

2. Selection
   Selection depends heavily on the discovery process taking as input the set of discovered services which can fulfil the requested functionality (e.g. train booking services). In the end, the requestor is only looking for one service which she/he will select from the set of services that can fulfill the requested functionality. Selection can be based on some non-functional properties like availability, price, etc. (e.g. from a set of services that offer online television, the requester will
select one service which is available in that moment in time and which has the lowest delay).

3. **Negotiation and Agreement**
   For the previous tasks we have considered that the values for the non-functional properties of services are somehow fixed. But this is not always the case in the real world, in business settings. Very often the execution of the service itself is preceded by a negotiation and agreement process. Different alternatives for non-functional properties values can be negotiated between service provider and service consumer which try both to promote their preferences. Non-functional properties like price, payment method, security, trust, and most notably quality of service are often the basis of such negotiation. In the end parties can reach an agreement or not.

4. **Monitoring**
   Once an agreement is reached the parties involved need to know if the other partners comply with the agreement or not. As mentioned above, such an agreement is usually constructed by negotiating on non-functional properties of the service. The monitoring of the agreement will be based on the non-functional properties.

5. **Substitution**
   It is often the case that service go offline or they simply do not provide the quality of service agreed before with the requestor. In this case the first service is replaced by another service or combination of services that provide the same functionality with the requested quality of service. This process is called substitution and non-functional properties could be used to determine if a service should be substitute, why it has to be substitute and what could be its replacement.

Semantic Web services aim at automating most of the tasks mentioned above. Such a vision can only be realized if semantic explicit formalizations of the Web services descriptions, including non-functional properties, are provided.
4 Non-functional properties in WS-MO/WSML

This section investigates how non-functional properties could be better supported in WS-MO/WSML. First, Section 4.1 talks about the types of non-functional properties. The modeling of these properties in WS-MO/WSML is discussed afterwards in Section 4.2 and a concrete approach on how to attach non-functional properties descriptions to services, goals and other elements in WS-MO/WSML is proposed in Section 4.3. The initial model to attach/specify non-functional properties of services is further developed in Section 4.5 and Section 4.6 by proposing a meta-model (see Section 4.4) that gives structure to specifications. The meta-model takes into account the WS-MO entities (services and goals) to provide a solution that captures the client and provider perspectives. In particular, services can be associated with conditioned policies to express composed offers; and goals can associate properties with relevance indicators to express user preferences. In Section 4.5, a straightforward way to attach the proposed meta-model to WS-MO service descriptions is presented. In Section 4.6, a sounder proposal for embedding the meta-model in WS-MO is presented. An example about how the same non-functional property specification is attached to a Web service description according to the approaches proposed in Section 4.3, Section 4.5 and Section 4.6 is proposed in Appendix B.

4.1 Non-functional properties types

In this section we discuss what are the types of non-functional properties. More precisely: (1) we distinguish between two types of non-functional properties and (2) we investigate what are properties that belong to each of these two categories.

A closer look at non-functional properties shows that there are two categories in which these properties can be divided: (1) annotations - which provide metadata about any type of element description (service, goal, ontology, etc.) and (2) non-functional properties - which are properties that strictly belong to a service, properties other than functional and behavioral.

1. Annotations, the first category of non-functional properties, are properties which can apply to all elements descriptions e.g services, goals, mediators, ontologies, etc. They are simply providing a way to annotate, to provide metadata about any type of element description.

   In this category we can include the following properties: contributor, coverage, creator, date, format, identifier, language, owner, publisher, rights, source, and version.

   Properties like subject, title, type, description can be used to add extra information about the service description and the service itself. Additional they can contain information about the functionality of the service (e.g. service category)

2. The second category of non-functional properties are those properties which strictly belong to a service and which are not functional and behavioral. We call these properties simply non-functional properties.

   This category includes properties that are describing the following aspects of a service: locative, temporal, availability, obligation, price, payment, discounts,
rights, trust, quality of service, security, intellectual property, rewards, provider, reliability, robustness, scalability, performance, transactional.

It is important to mention that the set of non-functional properties from both categories is extensible. Service providers and requesters might add other properties to annotations or non-functional properties categories except the ones mentioned above.

4.2 Modeling non-functional properties

An important challenge towards better supporting non-functional properties in WSMO/WSML is the modeling of these properties. For this problem the proposed solution is to define ontologies for specifying non-functional properties. These ontologies are used afterwards when the non-functional properties of services are specified. Ontologies, which describe the non-functional properties domain, can be imported and concepts referring to non-functional properties can be instantiated and used in the service descriptions.

An initial set of ontologies for specifying non-functional properties is provided in Appendix [A]. These ontologies provide a formal model for non-functional properties like availability, security, etc.

In the context of modeling non-functional properties some questions might arise e.g. which aspects of the service description are considered to be functional and which non-functional? The border between these aspects becomes very much dependent on the context. For example the same aspect could be seen sometimes as functional sometimes as non-functional (e.g. coverage).

4.3 Attaching non-functional properties to WSMO elements

Given any ontology model for non-functional properties, a second challenge that has to be addressed is how to attach non-functional properties descriptions to services, goals or any other WSMO element. This section provides a concrete solution for this problem.

The solution consists in modeling non-functional descriptions of services or goals in a way similar to which capabilities are currently modelled in WSMO/WSML. A service is an entity which provides a functionality (e.g. given a date, a start location, a destination and information about a client, a service can book a ticket for the desired trip), but at the same time a service can be seen as an entity which exhibits one or more non-functional properties (e.g. given a particular type of client a service charges a particular price, etc.). A simplified model of a WSMO service following this approach is:

```
webService
  capability idCapability
  precondition definedBy axiomPrecond
  postcondition definedBy axiomPostcond
  assumption definedBy axiomAssumption
  effect definedBy axiomEffect
  nonFunctionalProperty
    idNFP hasValue valuelistnfp definedBy axiomNFP
```

This approach has the following advantages:

- The set of non-functional properties is neither explicit nor finite. Therefore, users of WSMO/WSML can define and attach an open set of non-functional properties
to a goal or a service.

- Non-functional property models are attached to services in the same way as capabilities are.

and the following disadvantages:

- The WSML syntax has to be extended.

The rest of this section details the approach in terms of changes to WSMO conceptual model in Section 4.3.1 changes to WSML language syntax in Section 4.3.2. Finally an example is provided in Section 4.3.3.

4.3.1 Model

Following this approach the WSMO conceptual model is modified as follows:

Listing 4.1: WSMO conceptual model extensions

<table>
<thead>
<tr>
<th>Class</th>
<th>annotations</th>
</tr>
</thead>
<tbody>
<tr>
<td>hasContributor</td>
<td>type dc:contributor</td>
</tr>
<tr>
<td>hasCoverage</td>
<td>type dc:coverage</td>
</tr>
<tr>
<td>hasCreator</td>
<td>type dc:creator</td>
</tr>
<tr>
<td>hasDate</td>
<td>type dc:date</td>
</tr>
<tr>
<td>hasDescription</td>
<td>type dc:description</td>
</tr>
<tr>
<td>hasFormat</td>
<td>type dc:format</td>
</tr>
<tr>
<td>hasIdentifier</td>
<td>type dc:identifier</td>
</tr>
<tr>
<td>hasLanguage</td>
<td>type dc:language</td>
</tr>
<tr>
<td>hasOwner</td>
<td>type owner</td>
</tr>
<tr>
<td>hasPublisher</td>
<td>type dc:publisher</td>
</tr>
<tr>
<td>hasRelation</td>
<td>type dc:relation</td>
</tr>
<tr>
<td>hasRights</td>
<td>type dc:rights</td>
</tr>
<tr>
<td>hasSource</td>
<td>type dc:source</td>
</tr>
<tr>
<td>hasSubject</td>
<td>type dc:subject</td>
</tr>
<tr>
<td>hasTitle</td>
<td>type dc:title</td>
</tr>
<tr>
<td>hasType</td>
<td>type dc:type</td>
</tr>
<tr>
<td>hasVersion</td>
<td>type version</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Class</th>
<th>nonFunctionalProperty</th>
</tr>
</thead>
<tbody>
<tr>
<td>hasAnnotations</td>
<td>type annotations</td>
</tr>
<tr>
<td>hasDefinition</td>
<td>type logicalExpression</td>
</tr>
</tbody>
</table>

The modified WSMO model contains two classes for non-functional properties:

- A class which covers the annotations aspects of service, goal, mediators, ontologies descriptions and any of their elements. This class is called annotations and can be used not only for service descriptions but for any elements of WSMO model. The recommended set of annotations is based on Dublin Core Metadata Initiative [Weibel et al., 1998]. Non-functional properties that are represented by this class are for example author and subject. The set of annotations is not restricted to the properties listed in Listing 4.6.2. Additionally other annotations could be defined.

- A class which strictly covers the service related non-functional properties. This class is called nonFunctionalProperty and groups properties such as price and availability.

The principle behind this proposed extension is that functional and non-functional aspects of services could be modeled in a similar way. Thus, we model the nonFunctionalProperty class in a way similar to which a capability is currently modelled in WSMO/WSML. Non functional properties are defined using logical expressions same as pre/post-conditions, assumptions and effects are being defined in a capability. Modeling non-functional properties of a service or goal using
the nonFunctionalProperty construct will require a terminology provided by ontologies for specifying non-functional properties (c.f. Appendix A).

A similar approach which models conditions over the non-functional properties was proposed in [Hauswirth et al., 2005].

### 4.3.2 Syntax

Extending the WSMO conceptual model, as mentioned above, will generate a set of changes to WSML syntax as well. The following listing provides the WSML syntax extensions based on the conceptual model extensions previously presented.

**Listing 4.2: WSML syntax extensions**

```xml
annotations = 'annotations' attributevalue∗ 'endAnnotations'
nfp = 'nfp' attributevalue|nfp
attributevalue = id 'hasValue' valuelistnfp annotations?
log, definition?
log definition = 'definedBy' log expr+
header = annotations | importsontology | usesmediator
```

### 4.3.3 Example

In the following listing a concrete example on how to model non-functional properties of a service based on the proposed extensions to WSMO conceptual model and WSML syntax is presented. We consider a service which charges a specific price (lower than 10 euro) if the client is older than 60 or younger than 10 years old. This condition is expressed as a logical expression. Please note that nonFunctionalProperty, capability and interface are all constructs at the same level.

**Listing 4.3: Example of NFPs descriptions based on the proposed extensions**

```xml
webService "http://example.org/ws"
  annotations
dc#title hasValue "WSML example service with non-functional properties" endAnnotations
  nonFunctionalProperty
    price hasValue ?price
    annotations
dc#description hasValue "If the client is older than 60 or younger than 10 years old the invocation price is lower than 10 euro" endAnnotations
  importsOntology {?client[age hasValue ?age] memberOf hu#human and ?age[amount hasValue ?years, units hasValue hu#YearsDuration] memberOf hu#age and (greaterEqual(?years, 60) or lessEqual(?years,10)) implies ?price[hasAmount hasValue ?amount, hasCurrency hasValue cur#Euro] memberOf po#AbsolutePrice and lessEqual(?amount, 10).
    capability wsCapability
    interface wsInterface```
4.4 Policy-centered NFP meta-model

The general goal of a meta-model is to provide a frame for addressing the issues related to the tasks that need to be considered. In our context, there are three aspects of non-functional properties representation that are relevant for the tasks identified in Section 3.

1. Some non-functional properties may conflict with each other due to either technological constraints or business concerns. As an example, consider a Web service providing sport videos. As for conflicts related to technological constraints, the more is the resolution of a video the more is the bandwidth required. As for business related conflicts, consider the correlation between resolution and price, where the same functional outputs can be offered at a different price according to resolution level. More in general terms, a web service can provide the same functionality with different non-functional property levels. Several real world services provide evidence for this consideration. As an example, from the Italian railway company Trenitalia web site it is possible to book tickets with different fares such as Flexi, Standard, Amica, Blu card rail plus, and so on. Each of these fares is characterized by a set of non-functional properties such as price, discount and extra advantages; some of these properties such as allowed trains specify the term for the fare applicability. From the same web site is possible to buy a transit pass for a destination under different conditions (e.g., weekly or monthly one-way or return pass for IC/EC trains). Another example scenario is provided by the logistic operator domain. This scenario is fully described in Appendix D and it will be used as a running example in the following sections. Therefore, there should be the possibility of the explicit clustering of related non-functional properties. Moreover, the non-functional property ontology should take into account the conditioned provision of such offer clusters. This topic is related to the tasks of discovery, selection and negotiation and agreement.

2. Non-functional property descriptions are involved in both non-functional property offerings and non-functional property requests; this perspective is embedded in the WSMO model that distinguishes between services and goals, the former dealing with providers that offer non-functional properties and the latter dealing with clients that require non-functional properties. It is likely that different domain ontologies of non-functional properties could be used in offers and requests that are expressed by different providers and requesters. Therefore a meta-model that accommodates the two perspectives is advisable.

3. Non-functional property descriptions need to be compared and matched to address any of the tasks mentioned in Section 3. Therefore, the representation of non-functional property requirements should be enough expressive to accommodate different constraint specifications from the requester; conversely, non-functional property offers should take into account different ways of expressing offers; in particular, the possibility to state range-based offers (e.g. response time is between 10 and 30 ms) and the possibility to express dynamic non-functional properties, i.e., non-functional properties whose values need to be defined at runtime or on the basis of data fetched through provider-requester interaction (e.g. the price for shipment depending on the shipped object’s weight) need to be considered.

Typical Nfp-based Web service models are based on attribute-value non-functional property specifications. An attribute identifies the involved non-functional property and

\[ \text{http://www.ferroviedellostato.it/ferrovie/util/inglese.jsp} \]
\[ \text{The description of the several fares offered by Trenitalia can be found at http://www.trenitalia.com/en/oraribiglietti/infotariffe.html} \]
the value specifies its value, which is the value offered by the associated service. The shortcomings of the attribute-value approaches with respect to the three requirements discussed above can be synthesized as follows:

1. The description of non-functional properties clusters is possible if and only if axioms to define non-functional properties values are exploited. Moreover, axioms need to state the very same conditions for each non-functional property of the group. However, this approach makes the identification of the clusters quite difficult since it does not provide for an explicit representation of the joint offers.

2. The attribute-value approach can be adequate to describe non-functional property offers, but its expressiveness is very poor when non-functional property requirements are considered. Typically, non-functional property requirements make use of constraints expressions involving constraints operators (such as “≤”, “≥”, range specifications, and so on) that are not usually considered. In order to address such kinds of requests, new language constructs for non-functional property requests should be introduced.

3. Attribute-value approaches show limitations even for describing offers. Range-based offers are not easily expressed; dynamic non-functional properties can be described if and only if variables are supported and defined as part of axiom specifications. However, this solution is weak since a dynamic non-functional property would not be explicitly represented; moreover, data fetching would end up to be executed by general programs that cannot be referred in the axioms. The problem is that most of the existing semantic languages do not directly support variables and ranges.

For the reasons highlighted above, we propose a non-functional property model based on:

- The explicit representation of non-functional properties joint offers through the concept of Policy.
- The explicit representation of non-functional properties request in Goal descriptions through the concept of RequestedPolicy.
- A metamodel of offered and requested non-functional property specifications which is explicitly targeted at supporting Nfp-based matching. The abstract metamodel defines a specification structure, which can be used to implement non-functional property domain ontologies; it also leaves a high level of freedom in the specification of offered and requested non-functional properties to the user.

4.4.1 The NFP metamodel

This section has the aim to propose a metamodel that can be used for policy specification in Web services and Goal descriptions. The model defines an upper-level ontology; therefore it is independent from any specific language or framework (see [Comerio et al., 2007]). However, since the primary goal is to enable the automatic accomplishments of the tasks mentioned in Section 3, it is advisable to embed the model in the WSML language and the WSMO framework. The model has been tested in both WSML and OWL; compliance with other framework specifications (e.g., SAWSDL) has also been tested.

Figure 4.2 shows the top-level concepts of the proposed metamodel, that are: Policy, PolicyCondition, PolicyNfp, PolicyExpression, Unit. Policy is the concept that represents the main class of entities of the metamodel. This concept is characterized
The reference to the URI of a functional description can point to a WSML or to a wsdl web service description. This reference supports the management of policies when these are stored separately from the web service descriptions they refer to (see the examples in Section 4.5.2).
The aim of applicability condition is to define which clients can apply for policies (e.g., the premium policy can be accessed only by clients older than 60). The process of verifying the applicability conditions of policies may need hard reasoning capabilities and large background knowledge. The applicability condition for a policy can be described both by a text description and by instances of class PolicyCondition. The text description can be used to support a user in selecting the policy when the verification process cannot be completely automated. The instances of class PolicyCondition specified the arbitrary axioms that logically define the condition. However, in order to support the specification of such axioms in a structured way, the meta-model ontology provides also the concept of Client and two new relations:

- satisfiedFor (ofType PolicyCondition, ofType Client)
- satisfied (ofType PolicyCondition)

Such relations can be used in an atom representing the head of axiom rules defining the applicability condition. The first (binary) relation can be used to relate the applicability of a policy to a client, e.g., identified by variable appearing in the atom and in the rule body. The second (unary) relation can be used to avoid a specific reference to the client. An example of exploitation of the binary relation is provided in Appendix B, in Listing B.2.

The key element in the Policy description is the specification of a set of PolicyNfp. A PolicyNfp is basically characterized by a PolicyExpression. Each PolicyExpression is defined by a ConstraintOperator and by a set of different attributes that depends on the constraint operator used. Consequently, starting from a classification of ConstraintOperator, a classification of PolicyExpression and PolicyNfp is defined. Figure 4.2 shows the logical relationships between the three classifications.

PolicyNfps are classified as follows:

- QuantitativeNfp: represents non-functional properties that assume numeric values. These non-functional properties can assume a single value (i.e., SingleValueNfp) or a range of values (i.e., RangeNfp).

- QualitativeNfp: represents non-functional properties that assume values specified in an ontology. These non-functional properties can be expressed with logical constraint operators (i.e., ListNfp) or with user defined constraint operators (i.e., UserDefinedNfp).

PolicyExpressions are classified in QuantitativeExpression and QualitativeExpression. SingleValueExpression and RangeExpression are disjoint subclasses of QuantitativeExpression. ListExpression and UserDefinedExpression are disjoint subclasses of QualitativeExpression.

A SingleValueExpression is defined by a binary constraint operator, a value and a unit:

```
concept SingleValueExpression subConceptOf QuantitativeExpression
hasOperator ofType (1 1) BinaryOperator
hasParameter ofType (1 1) float
hasUnit ofType (1 1) Unit
```

A RangeExpression is defined by a ternary constraint operator, a minimum value, a maximum value and a unit:

```
concept RangeExpression subConceptOf QuantitativeExpression
hasOperator ofType (1 1) TernaryOperator
hasMinParameter ofType (1 1) float
hasMaxParameter ofType (1 1) float
hasUnit ofType (1 1) Unit
```

A ListExpression is defined by a logical constraint operator, a set of values and a unit:
A UserDefinedExpression is defined by a user defined constraint operator, a set of values and a unit:

concept UserDefinedExpression subConceptOf QualitativeExpression
hasOperator ofType (1 1) UserDefinedOperator
hasParameters ofType (1 *) <anyType>

Figure 4.3: Required NFP meta-model

Figure 4.3 shows how the described meta-model is extended to allow the specification of required non-functional properties into a Goal description. This extension is performed into three directions: (i) the introduction of Request as a subclass of PolicyNfp that extends this one with the property hasRelevance (i.e., assumes a value in the range [0..1], where 0 means irrelevant and 1 means required); (ii) the introduction of particular types of required non-functional property as subclasses of Request and of the corresponding particular subclass of PolicyNfp. As an example, the SingleValueRequest is introduced as a subclass of Request and SingleValueNfp. It inherits the hasRelevance property from Request and hasOperator, hasParameter and hasUnit from SingleValueNfp; (iii) the introduction of the class RequestedPolicy as a subclass of Policy. This class is characterized by a set of Request. For the complete axiomatization of these logical dependencies see Appendix C. Observe that for RequestedPolicy (i) the attribute referenceFunctionalDescription should point to a WSMO Goal and (ii) the specification of conditions (textual and logical) is not significant.

In this model offered and requested non-functional properties are coupled by inheritance to deliver a stratified model. A simpler version of the model (a flat model) that detaches offered and requested non-functional properties can be defined to facilitate editing. Such a model does not make the concept of PolicyExpression explicit; it provides a flat frame-based representation of the basic classes of offered non-functional properties (SingleValueNfp, RangeNfp, ListNfp, UserDefinedNfp) and requested non-functional properties (SingleValueRequest, RangeRequest, ListRequest, UserDefinedRequest).
4.5 Using the NFP meta-model in WSMO

In this section a straightforward way to attach the proposed meta-model to Web service and Goal descriptions is shown. This approach does not require any change in the WSMO syntax. Section 4.5.1 shows the approach and Section 4.5.2 provides examples.

4.5.1 Attaching NFP policies to WSMO elements

The solution consists in modeling non-functional properties of Web services introducing the concept of policy (importing the proposed non-functional property meta-model) and defining it. Policy is a name and therefore an identifier (see WSMO D16.1).

In the above listing, "http://www.siti.disco.unimib.it/research/ontologies/OntoNFPwsml#" is the URI of the ontology representing the meta-model and "http://www.example.com/PolicyOntology#" is the URI of an ontology where policies are stored and described (the policy instance ontology). Observe that the axiom definition following the policy declaration can be omitted; it can be assumed that the membership of policy instances to the class OntoNFPwsml#Policy is stated in the policy instance ontology.

Non-functional properties are modelled in Goal specification in a similar way. The concept of RequestedPolicy is used as a requester-side counterpart of Policy.

4.5.2 Example

The following section provides a concrete example about how to use the non-functional property meta-model to define Policy and RequestedPolicy and how to attach them to Web service and Goal descriptions.

We consider that a user is looking for a shipment service offered by a logistic operator. Among the several non-functional properties that characterize this service (see Appendix D), in this example the following ones are considered: (i) accepted payment methods; (ii) base price and (iii) maximum hours to perform the delivery.
4.5.2.1 Example of Web Service Description

The non-functional properties offered by a Web service are stated by policies associated with Web service descriptions. In this section, the description of the WSOmni-Transport service that offers a policy with specified values of Payment Method, Base Price and Hours To Delivery is proposed. The policy is applicable for users that request repetitive shipments. The policy is associated to the Web service description using an external link. This solution allows the specification of the complex policy in a separate ontology.

Listing 4.4: Example of Web Service descriptions based on the proposed extensions

```xml
<wsmlVariant _http://www.wsmo.org/wsml/wsml – syntax/wsml – flight
namespace {
  "http://www.siti.disco.unimib.it/research/LOWS#",
  dc "http://purl.org/dc/elements/1.1",
  pol3 "http://www.siti.disco.unimib.it/research/ontologies/LGPolicy3#"
}

webService WSOmniTransport
  annotations
dcCreator hasValue ("Marco Comerio")
endAnnotations

nonFunctionalProperty
  annotations
dcDescription hasValue ("The WSOmniTransport offers the NFP specified in multiTravelPolicy ")
endAnnotations

importsOntology {
  "http://www.siti.disco.unimib.it/research/ontologies/LOPolicy3#PolicyInstanceLOOntology3"
}

policy pol3#policyForRepetitiveTravel

capability wsCapability
interface wsInterface
```

Listing 4.5: Example of Policy specification

```xml
<wsmlVariant _http://www.wsmo.org/wsml/wsml – syntax/wsml – flight
namespace {
  "http://www.siti.disco.unimib.it/research/ontologies/LLOntology#",
  ontoNFP "http://www.siti.disco.unimib.it/research/ontologies/OntoNFPwsml#",
  dc "http://purl.org/dc/elements/1.1",
  lo "http://www.siti.disco.unimib.it/research/ontologies/LOOntology#",
  nfpo "http://www.siti.disco.unimib.it/research/ontologies/NFPOntology#"
}

ontology policyInstanceLOOntology3
  importsOntology {
    "http://www.siti.disco.unimib.it/research/ontologies/OntoNFPwsml#OntoNFPwsml",
    "http://www.siti.disco.unimib.it/research/ontologies/LOOntology#LOOntology",
    "http://www.siti.disco.unimib.it/research/ontologies/NFPOntology#NFPOntology"
  }

instance policyForRepetitiveTravel memberOf ontoNFP#Policy
  annotations
dcDescription hasValue ("Policy Instance for Logistic Operator")
endAnnotations

ontoNFP#referenceFunctionalDescription hasValue ("http://www.siti.disco.unimib.it/research/LOWS#WSOmniTransport")

ontoNFP#hasConditionDescription hasValue ("This policy is applicable for client that required multiple travels")

ontoNFP#hasCondition hasValue policyCondition

ontoNFP#hasNfp hasValue offeredPaymentMethod
ontoNFP#hasNfp hasValue offeredBasePrice
ontoNFP#hasNfp hasValue offeredHoursToDelivery

axiom policyCondition
definedBy
  applicable(policyForRepetitiveTravel) :- #([numberOfTravelsSubscribed hasValue ?n and ?n > 1]
    memberOf lo#LogisticOperatorClient.
```
4.5.2.2 Example of Goal Description

The required non-functional properties for the shipment service selection are: (i) payment method equal to carriage paid; (ii) service base price lower than 50.0 Euros; (iii) hours to perform the delivery lower than 72. The user also specifies that the non-functional properties Payment Method and Base Price are required (i.e., relevance=0.8) and the non-functional property Hours To Delivery is preferred (i.e., relevance=0.6).

The requested policy is associated to the Goal description using an external link. This solution allows the specification of the complex requested policy in a separate ontology.

---

Listing 4.6: Example of Goal descriptions based on the proposed extensions

```wsml
wsmlVariant "http://www.wsmo.org/wsml/wsml−syntax/wsml−flight"
namespace { "http://www.siti.disco.unimib.it/research/LOGoal#",
    dc: "http://purl.org/dc/elements/1.1/",
    req1: "http://www.siti.disco.unimib.it/research/ontologies/LORequest1#"
}
goal ordinaryTransport
  importsOntology { _http://www.siti.disco.unimib.it/research/ontologies/RequestInstanceLOOntology# requestInstanceLOOntology#}
  annotations
dc#creator hasValue { "Marco Comerio" }
endAnnotations
nonFunctionalProperty
  annotations
dc#description hasValue { "The ordinaryTransport Goal requires the NFPs specifies in LOrequest1" }
endAnnotations
importsOntology { _http://www.siti.disco.unimib.it/research/ontologies/LOPolicy1# PolicyInstanceLOOntology1#}
policy hasValue { req1#LOrequest1#}
capability wsCapability
interface wsInterface
```

Listing 4.7: Example of RequestedPolicy specification
4.6 Embedding the NFP meta-model in WSMO

In this section we propose to embed the non-functional property meta-model proposed in Section 4.4.1 in WSMO. Section 4.6.1 makes a proposal on how to attach Policy and RequestedPolicy to Web service and Goal descriptions. Sections 4.6.2 and 4.6.3 show how the WSMO conceptual model and syntax must be extended. Finally, Section 4.6.4 provides examples.
4.6.1 Attaching NFP policies to WSMO elements

An alternative to the approach to attach non-functional property policies to WSMO elements proposed in Section 4.5.1 consists in embedding the proposed meta-model and using it in Web service and Goal descriptions.

The solution consists in modeling non-functional properties of Web services introducing the concept of Policy. Each Policy is characterized by an applicability condition, its description and by a set of offered non-functional properties.

```plaintext
webService
  annotation idAnnotation
  policy idPolicy
  policyConditionDescription idPolicyCondDescr
  policyCondition idPolicyCondition
  policyNfp idNFP1
  policyNfp idNFP2
  ...
  policyNfp idNFPn
```

Non-functional properties are modelled in Goal specification in the same way. Each `requestedPolicy` is characterized by a set of required non-functional properties.

```plaintext
goal
  annotation idAnnotation
  requestedPolicy idReqPolicy
  policyNfp idNFP1
  policyNfp idNFP2
  ...
  policyNfp idNFPn
```

4.6.2 Model

Following this approach the WSMO conceptual model must be extended as follows:

```plaintext
Listing 4.8: WSMO conceptual model extensions

Class Policy
  hasAnnotations type annotations
  importsOntology type ontology
  usesMediator type ooMediator
  hasReferenceFunctionalDescription type iri
  hasConditionDescription type string
  hasCondition type policyCondition
  hasNfp type PolicyNfp

Class RequestedPolicy sub Class Policy
  hasNfp type Request

Class PolicyNfp
  hasExpression type PolicyExpression multiplicity = single–valued

Class QualitativeNfp sub Class PolicyNfp
  hasExpression type QualitativeExpression multiplicity = single–valued

Class ListNfp sub Class QualitativeNfp
  hasExpression type ListExpression multiplicity = single–valued

Class UserDefinedNfp sub Class QualitativeNfp
  hasExpression type UserDefinedExpression multiplicity = single–valued

Class QuantitativeNfp sub Class PolicyNfp
  hasExpression type QuantitativeExpression multiplicity = single–valued

Class SingleValueNfp sub Class QuantitativeNfp
  hasExpression type SingleValueExpression multiplicity = single–valued

Class RangeNfp sub Class QuantitativeNfp
  hasExpression type RangeExpression multiplicity = single–valued
```
4.6.3 Syntax

Extending the WSMO conceptual model, as mentioned above, will generate a set of changes to WSML syntax as well. The following listings provides the WSML syntax extensions based on the conceptual model extensions previously presented.

Concerning offered non-functional properties a new element called Policy is introduced. A Web service can expose several policies according to different quality levels. Thus, for example, a basic policy with a standard set of non-functional properties and

The complete OntoNFP WSML ontology is provided in Appendix C.
a gold policy which manages more non-functional properties and requests the payment of a fee could be defined for a Web service. In order to be used by user, each policy is characterized by a logical condition that has to be satisfied. According with the WSML conventions, we define the following production rules:

### Listing 4.9: WSML syntax extensions

```
webservice = 'webService' nonanonymous? header* capability? policy* interface*
policy = 'policy' id header* policycondition* offerednfps*
policycondition = conditionTextualDescription? conditionLogicalDescription*
  conditionTextualDescription = 'conditionDescription' t hasvalue string
  conditionLogicalDescription = 'condition' expr
  offerednfps = 'policyNfp' id memberof? nfpExpr
  nfpExpr = SingleValueExpression | RangeExpression | ListExpression
  | UserDefinedExpression
  SingleValueExpression = 'constraintOperator' t hasvalue
  BinaryOperator 'parameter' t hasvalue number 'unit' t hasvalue value
  RangeExpression = 'constraintOperator' t hasvalue TernaryOperator
  'minParameter' t hasvalue number 'maxParameter' t hasvalue number
  'unit' t hasvalue value
  ListExpression = 'constraintOperator' t hasvalue ListOperator
  'parameter' t hasvalue valuelist
  UserDefinedExpression = 'constraintOperator' t hasvalue
  UserDefinedOperator 'parameter' t hasvalue valuelist
  BinaryOperator = 'atLeast' | 'atMost'
  TernaryOperator = 'interval'
  ListOperator = 'exact' | 'exist'
  UserDefinedOperator = value
```

Concerning the requested non-functional properties, a new element called RequestedPolicy is introduced in the Goal description to collect a set of required non-functional properties. The new proposed WSML production rules are:

### Listing 4.10: WSML syntax extensions

```
goal = 'goal' nonanonymous? header* capability? requestedPolicy* interface*
requestedPolicy = 'requestedPolicy' id header* requirednfps*
requirednfps = 'policyNfp' id memberof? nfpExpr 'relevance' number
```

### 4.6.4 Example

In the following listing a concrete example on how to model non-functional properties of a service based on the proposed extensions to WSMO conceptual model and WSML syntax is presented.
The examples described in Section 4.5.2 are rewritten according to the new proposal.

### 4.6.4.1 Example of Web Service Description

An example of non-functional property policy description that offers a service with specified values of Payment Method, Base Cost and Hours To Delivery is the following:

Listing 4.11: Example of Web Service descriptions based on the proposed extensions

```xml
wsmlVariant "http://www.wsmo.org/wsml/wsml--syntax/wsml--flight"
namespace {
    http://www.siti.disco.unimib.it/research/LOWS#,
    dc "http://purl.org/dc/elements/1.1/",
    lo "http://www.siti.disco.unimib.it/research/ontologies/LOOntology#",
    nfpo "http://www.siti.disco.unimib.it/research/ontologies/NFPOntology#"
}
webService WSOmniTransport
  annotations
    dc#creator hasValue "Marco Comerio"
  endAnnotations
importsOntology {
    http://www.siti.disco.unimib.it/research/ontologies/LOOntology#LOOntology,
    http://www.siti.disco.unimib.it/research/ontologies/NFPOntology#NFPOntology
}
policy policyForRepetitiveTravel
  annotations
    dc#description hasValue "Policy Instance for Logistic Operator"
  endAnnotations
  conditionDescription hasValue "This policy is applicable for client that required multiple travels"
  condition repetitiveTravelCondition definedBy
    appliable(policyForRepetitiveTravel) :-
    numberOfTravelsSubscribed hasValue ?n and ?n > 1
    memberOf lo#LogisticOperatorClient.

policyNfp offeredPaymentMethod memberOf nfpo#ListPaymentMethod
  constraintOperator hasValue exact
  parameters hasValue nfpo#carriageForward

policyNfp offeredBasePrice memberOf nfpo#SingleValueBasePrice
  constraintOperator hasValue equal
  parameter hasValue 20
  unit hasValue nfpo#euro

policyNfp offeredHoursToDelivery memberOf nfpo#RangeHoursToDelivery
  constraintOperator hasValue interval
  minParameter hasValue 24
  maxParameter hasValue 48
  unit hasValue nfpo#hour

capability wsCapability
  interface wsInterface
```

### 4.6.4.2 Example of Goal Description

An example of requested policy description that required (i) payment method equal to carriage paid; (ii) service base price lower than 50.0 Euros and (iii) hours to perform the delivery lower than 72 is the following:

Listing 4.12: Example of Goal descriptions based on the proposed extensions

```xml
wsmlVariant "http://www.wsmo.org/wsml/wsml--syntax/wsml--flight"
namespace {
    http://www.siti.disco.unimib.it/research/LOGoal#,
    dc "http://purl.org/dc/elements/1.1/",
    lo "http://www.siti.disco.unimib.it/research/ontologies/LOOntology#",
    nfpo "http://www.siti.disco.unimib.it/research/ontologies/NFPOntology#"
}
webService WSOmniTransport
  importsOntology {
    http://www.siti.disco.unimib.it/research/ontologies/LOOntology#LOOntology,
    http://www.siti.disco.unimib.it/research/ontologies/NFPOntology#NFPOntology
}
policy policyForRepetitiveTravel
  condition repetitiveTravelCondition definedBy
    appliable(policyForRepetitiveTravel) :-
    numberOfTravelsSubscribed hasValue ?n and ?n > 1
    memberOf lo#LogisticOperatorClient.
```

```xml
policyNfp offeredPaymentMethod memberOf nfpo#ListPaymentMethod
  constraintOperator hasValue exact
  parameters hasValue nfpo#carriageForward

policyNfp offeredBasePrice memberOf nfpo#SingleValueBasePrice
  constraintOperator hasValue equal
  parameter hasValue 20
  unit hasValue nfpo#euro

policyNfp offeredHoursToDelivery memberOf nfpo#RangeHoursToDelivery
  constraintOperator hasValue interval
  minParameter hasValue 24
  maxParameter hasValue 48
  unit hasValue nfpo#hour
```
lo _"http://www.siti.disco.unimib.it/research/ontologies/LOOntology#", nfpo _"http://www.siti.disco.unimib.it/research/ontologies/NFPOntology#" }

**goal** ordinaryTransport

annotations

dc#creator hasValue {"Marco Comerio"}
endAnnotations

**importsOntology** { _"http://www.siti.disco.unimib.it/research/ontologies/LOOntology#LOOntology", _"http://www.siti.disco.unimib.it/research/ontologies/NFPOntology#NFPOntology"

requestedPolicy loRequest

annotations

dc#description hasValue {"The ordinaryTransport Goal requires the NFPs specifies in loRequest"}
endAnnotations

policyNfp requestedPaymentMethod memberOf nfpo#ListPaymentMethodRequest

constraintOperator hasValue exact

parameters hasValue nfpo#carriagePaid

relevance hasValue 0.8

policyNfp requestedBasePrice memberOf nfpo#SingleValueBasePriceRequest

constraintOperator hasValue lessEqual

parameter hasValue 50

unit hasValue nfpo#euro

relevance hasValue 0.8

policyNfp requestedHoursToDelivery memberOf nfpo#SingleValueHoursToDeliveryRequest

hasOperator hasValue lessEqual

hasParameter hasValue 72

hasUnit hasValue nfpo#hour

hasRelevance hasValue 0.6

capability wsCapability

interface wsInterface
5 Annotations

As mentioned in Section 4.1, we distinguish between two types of properties that are neither functional, neither behavioral. These are: (1) annotations and (2) non-functional properties. This section discusses the first category.

Annotations are used in WSMO to attach information to any WSMO element such as the title or the creator of the element. The recommended set of annotations is based on Dublin Core Metadata Initiative [Weibel et al., 1998]. Additionally, two other annotations were introduced: ‘version’ and ‘owner’. The ‘version’ annotation could be used to identify any WSML element in time, since these elements might change in time. The ‘owner’ annotation could be used to specify a person or organization to which the WSMO element belongs.

The complete set of annotations includes: contributor, coverage, creator, date, description, format, identifier, language, owner, publisher, relation, rights, source, subject, title, type, typeOfMatch, version. For a complete description on which annotations apply to which WSMO elements and what is the purpose of each annotation we refer the reader to [Roman et al., 2005] (Section 9).

Currently annotations are only part of the WSML syntax without having a logical meaning. The next versions of WSML [de Bruijn et al., 2005] will consider also solving this aspect.
6 Ontologies for specifying non-functional properties

This section introduces a set of 17 ontologies for specifying service related non-functional properties. These ontologies are based on the work available in [O'Sullivan et al., 2005]. They contains models of the most used non-functional properties of services and provide the terminology needed to specify non-functional aspects of services. A WSML formalization of these ontologies is available in Appendix [A]. Please note that they are not models that we impose but rather they can adopted and extended given various needs. Alternatively, other models for non-functional properties might be developed. The set of non-functional properties includes models for: locative, temporal, availability, obligation, price, payment, discounts, rights, trust, quality of service, security, intellectual property, rewards, provider, measures and currency aspects. In the rest of the section we shortly describe each ontology in terms of purpose and important concepts. For a detailed description of the non-functional properties models we refer the reader to [O'Sullivan et al., 2005].

1. Locative Ontology
   The Locative Ontology provides the locative concepts that are needed for locative descriptions of a service. Using the terminology provided in this ontology aspects such where a service can be requested from, where it can be provided can be modelled. Main concepts include: LocativeEntity, GeoLocation, Address, RouteSpecification, etc.

2. Temporal Ontology
   The Temporal Ontology provides the temporal concepts that are needed for time related descriptions of a service. Different temporal granularities are considered. Using the terminology provided in this ontology restrictions such as when the service can be requested, provided or queried for further information can be expressed. Main concepts include: TemporalEntity, TimeInterval, TimePoint, TimeZone, TemporalDate, etc.

3. Availability Ontology
   The Availability Ontology provides the terminology needed to specify when, where, and to whom a service is available. Concepts included are: Availability and RequestAvailability

4. Obligation Ontology
   The Obligation Ontology provides the terminology needed to describe various obligations which may be connected to service request and provision. This includes for example pricing and payment obligations. Pricing obligations are related to service providers and include information regarding refund procedures, negotiability, etc. Payment obligations are related to service requestors which have the obligation to pay the service and include information such as payment discounts, charge, etc. Main concepts include: PaymentObligation, PricingObligation, etc.

5. Price Ontology
   The Price Ontology provides the terminology needed to describe properties of a service related to price. Different types of prices are modelled. Main concepts include: Price, AbsolutePrice, ProportionalPrice, RangedPrice, MechanismAuction, etc.
6. Payment Ontology
The Payment Ontology provides the terminology needed to describe how a service requestor can fulfill their payment obligations. Payment and Price ontology contain two views of the same thing but from different perspectives. Main concepts include: PaymentInstrument, PaymentScheme, CashInstrument, ElectronicCashType, etc.

7. Discounts Ontology
The Discounts Ontology provides the terminology needed to describe various types of discounts. Discounts are either dependent of how a requestor pays (e.g. early payment, type of payment instrument, etc.) and who the requestor is (e.g. age group, student, membership, etc.). Main concepts include: Discount, PayeeDiscount, StudentDiscount, MembershipDiscount, etc.

8. Rights Ontology
The Rights Ontology provides the terminology needed to describe rights granted to service providers or service requestors. Main concepts include: Right, RightOfWarranty, RightOfAccess, etc.

9. Trust Ontology
The Trust Ontology provides the terminology needed to describe trust aspect of a service. This model is is directly influenced by other models such as endorsement. Main concepts include: Endorsement, InternallyManagedEndorsement and ExternallyManagedEndorsement

10. Quality of Service Ontology
The Quality of Service Ontology provides terminology relative to a standard, an industrial benchmark and/or a ranking schema. Main concepts include: Standard, Rating, Rated, Ranking, etc.

11. Security Ontology
The Security Ontology contains concepts such as IdentificationRequirement, Confidentiality, EncryptionTechnique, IdentificationType, etc.. Two aspects are modelled: identification and confidentiality.

12. Intellectual Property Ontology
The Intellectual Property Ontology provides the concepts that are needed to describe Intellectual Property aspects. Main concepts include: IPRight, Trademark, Patent, Design, etc.

13. Rewards Ontology
The Rewards Ontology includes concepts such as AccumulatedReward, AccumulatedPriceReward, RedeemableReward, etc.

14. Provider Ontology
The Provider Ontology provides the basic terminology that is required when talking about service providers. Main concepts include: Provider, ProviderMembership, Compliance, PartnerType, etc.

15. Measures Ontology
The Measures Ontology provides a general measures terminology. Main concepts include: UnitOfMeasure, MeasurableQuantity, Distance, etc.

16. Currency Ontology
The Currency Ontology is a simple ontology that contains the most used currencies.
7 Conclusions and Future Work

This deliverable aims to provide a better support for Web services non-functional properties descriptions in WSMO/WSML. We have surveyed some solutions for defining and modeling non-functional properties and we have discussed what are the tasks for which non-functional properties are relevant. Furthermore we distinguished between two categories of non-functional properties and we have proposed a set of changes to WSMO/WSML towards a richer and useful support for non-functional descriptions in WSMO/WSML. We have provided as well a set of ontologies for non-functional properties based on the model adopted from O’Sullivan et al., 2005.

As future work we plan to refine the set of ontologies provided in the Appendix A and to illustrate how they can be used in describing the non-functional properties of a service. More detailed examples of service non-functional properties will be provided.

Acknowledgements

The work is funded by the European Commission under the projects ASG, DIP, enIRAf, InfraWebs, Knowledge Web, Musing, Salero, SEKT, Seemp, SemanticGOV, Super, SWING and TripCom; by Science Foundation Ireland under the DERI-Lion Grant No.SFI/02/CE1/I13; by the FIT-IT (Forschung, Innovation, Technologie - Informationstechnologie) under the projects Grisino, RW², SemNetMan, SeNSE and TSC.

The editors would like to thank to all the members of the WSMO and WSML working groups for their advice and input to this document.
Appendix A


Listing A.1: Runner

```xml
wsmlVariant "http://www.w3.org/2001/XMLSchema"

namespace "WSRunner.wsml#"

definitions

"Runner's obligations definition"

definitionObligations

"in case the package is lost or damaged Runner's liability is"
hasPackageLiability(?package, 150): hasPackageLiability(?package, 150); hasPackageLiability(?package, ?value) and ( ?value = < 150 .

"in case the package is not lost or damaged Runner's liability is 0"

packageDeclaredValue(?package, ?value): packageDeclaredValue(?package, ?value) and ( ?value = < 100 .

"calculate the liability of package list"

calculate the liability of package list

packageListHasLiability(list, empty, 0).

packageListHasLiability(list, #list(?object, #list), #total LIABILITY) :

packageListHasLiability(list, #list(?object, #list), #total LIABILITY) :

packageListHasLiability(list, #list(?object, #list), #total LIABILITY) :

packageListHasLiability(list, #list(?object, #list), #total LIABILITY) :

packageListHasLiability(list, #list(?object, #list), #total LIABILITY) :

packageListHasLiability(list, #list(?object, #list), #total LIABILITY) :

```

Listing A.2: Locative Ontology

```xml
wsmlVariant "http://www.w3.org/2001/XMLSchema"

namespace {

}```
"http://www.wsmo.org/ontologies/nfp/locativeNFPOntology#",

dc: _http://purl.org/dc/elements/1.1#",

xsd: _http://www.w3.org/2001/XMLSchema#",

wsml: _http://www.wsmo.org/wsml/wsml−syntax#",

temp: _http://www.wsmo.org/ontologies/nfp/temporalNFPOntology",

qu: _http://www.wsmo.org/ontologies/nfp/qualityNFPOntology",


meas: _http://www.wsmo.org/ontologies/nfp/measuresNFPOntology"

}  

ontology _http://www.wsmo.org/ontologies/nfp/locativeNFPOntology"

nonFunctionalProperties

dc:title hasValue "Locative Ontology"
dc:creator hasValue { "Ioan Toma" }
dc:description hasValue "Locative Ontology"
dc:publisher hasValue "DERI Innsbruck"
dc:contributor hasValue { "Ioan Toma", "Doug Foxvog" }
dc:date hasValue "2006−05−06"
dc:type hasValue "http://www.wsmo.org/2004/d2#ontologies"
dc:format hasValue "text/html"
dc:identifier hasValue "http://www.wsmo.org/ontologies/nfp/locativeNFPOntology"
dc:language hasValue "en−US"
wsml:version hasValue "Revision : 1.3"

endNonFunctionalProperties

importsOntology

{ "http://www.wsmo.org/ontologies/nfp/temporalNFPOntology",
  "http://www.wsmo.org/ontologies/nfp/qualityNFPOntology",
  "http://www.wsmo.org/ontologies/nfp/securityNFPOntology",
  "http://www.wsmo.org/ontologies/nfp/measuresNFPOntology" }

concept LocativeEntity

nonFunctionalProperties

dc:description hasValue "LocativeEntity concept definition"
endNonFunctionalProperties

hasName ofType string

supportsWrittenLanguage ofType Language

supportsSpokenLanguage ofType Language

canBeCommunicatedAccordingTo ofType qu#Standard

hasIdentificationRequirement ofType (1) sec#IdentificationRequirement

hasConfidentiality ofType (1) sec#Confidentiality

concept GeoLocation subConceptOf LocativeEntity

nonFunctionalProperties

dc:description hasValue "GeoLocation concept definition"
endNonFunctionalProperties

hasLatitude ofType (1) meas#Angle

hasLongitude ofType (1) meas#Angle

hasAltitude ofType (0 1) meas#Distance

axiom validLatitude
definedBy

|− ![point [latitude hasValue ?LA] memberOf GeoLocation and ?LA [numUnits hasValue ?NU, ofUnits hasValue meas#DegreeOfArc] memberOf meas#Angle and (less(90, ?NU) or greater(0, ?NU))].

axiom validLongitude
definedBy

|− ![point [longitude hasValue ?LO] memberOf GeoLocation and ?LO [numUnits hasValue ?NU, ofUnits hasValue meas#DegreeOfArc] memberOf meas#Angle and (less(180, ?NU) or greaterEqual(0, ?NU))].

axiom validPoleGeoLocation
definedBy

|− ![point [latitude hasValue ?LA, longitude hasValue ?LO] memberOf GeoLocation and ?LA [numUnits hasValue 90, ofUnits hasValue meas#DegreeOfArc] memberOf meas#Angle and ?LO [numUnits hasValue 79, ofUnits hasValue meas#DegreeOfArc] memberOf meas#Angle].

concept CompassDirection

nonFunctionalProperties

dc:description hasValue "Direction concept definition"
endNonFunctionalProperties

ofUnits ofType meas#UnitOfArc

axiom validCompassDirection
definedBy

|− ![dir [numUnits hasValue ?ANGLE, ofUnits hasValue meas#DegreeOfArc] memberOf CompassDirection and (lessEqual(360, ?ANGLE) or greater(0, ?ANGLE))].
nonFunctionalProperties

dc#description hasValue "VehicularRoute concept definition"
endNonFunctionalProperties

hasName ofType (1 1) string

hasSpecification ofType RouteSpecification

intendedForVehicles ofType (1 +) Vehicle

doPathFor ofType (1 +) Vehicle

concept VehicleType

nonFunctionalProperties

dc#description hasValue "VehicleType"
endNonFunctionalProperties

concept PathThroughWater subConceptOf VehicularRoute

nonFunctionalProperties

dc#description hasValue "PathThroughWater concept definition"
endNonFunctionalProperties

concept WaterSurfacePath subConceptOf PathThroughWater

nonFunctionalProperties

dc#description hasValue "WaterSurfacePath concept definition"
endNonFunctionalProperties

concept Canal subConceptOf WaterSurfacePath

nonFunctionalProperties

dc#description hasValue "Canal concept definition"
endNonFunctionalProperties

concept SeaLane subConceptOf WaterSurfacePath

nonFunctionalProperties

dc#description hasValue "SeaLane concept definition"
endNonFunctionalProperties

concept SolidSurfacePathThroughAir subConceptOf VehicularRoute

nonFunctionalProperties

dc#description hasValue "SolidSurfacePathThroughAir concept definition"
endNonFunctionalProperties

concept PathForWheeledVehicles subConceptOf SolidSurfacePathThroughAir

nonFunctionalProperties

dc#description hasValue "PathForWheeledVehicles concept definition"
endNonFunctionalProperties

concept Railway subConceptOf PathForWheeledVehicles

nonFunctionalProperties

dc#description hasValue "Railway concept definition"
endNonFunctionalProperties

concept Subway subConceptOf Railway

nonFunctionalProperties

dc#description hasValue "Subway concept definition"
endNonFunctionalProperties

concept Roadway subConceptOf PathForWheeledVehicles

nonFunctionalProperties

dc#description hasValue "Roadway concept definition"
endNonFunctionalProperties

concept RoadLane subConceptOf PathForWheeledVehicles

nonFunctionalProperties

dc#description hasValue "RoadLane concept definition"
endNonFunctionalProperties

concept Driveway subConceptOf PathForWheeledVehicles

nonFunctionalProperties

dc#description hasValue "Driveway concept definition"
endNonFunctionalProperties

concept BicyclePath subConceptOf PathForWheeledVehicles

nonFunctionalProperties

dc#description hasValue "BicyclePath concept definition"
endNonFunctionalProperties

concept SkiSlope subConceptOf SolidSurfacePathThroughAir

nonFunctionalProperties

dc#description hasValue "SkiSlope concept definition"
endNonFunctionalProperties

concept SkiJump subConceptOf SkiSlope
D28.4 – Non-functional properties in Web services

nonFunctionalProperties
dc#description hasValue "SkiJump concept definition"
endNonFunctionalProperties

concept Footpath subConceptOf SolidSurfacePathThroughAir
nonFunctionalProperties
dc#description hasValue "Footpath concept definition"
endNonFunctionalProperties

concept Stairway subConceptOf Footpath
nonFunctionalProperties
dc#description hasValue "Stairway concept definition"
endNonFunctionalProperties

concept MovingWalkway subConceptOf Footpath
nonFunctionalProperties
dc#description hasValue "MovingWalkway concept definition"
endNonFunctionalProperties

concept MovingStairway subConceptOf MovingWalkway
nonFunctionalProperties
dc#description hasValue "MovingStairway concept definition"
endNonFunctionalProperties

concept Trail subConceptOf Footpath
nonFunctionalProperties
dc#description hasValue "Trail concept definition"
endNonFunctionalProperties

concept Sidewalk subConceptOf Footpath
nonFunctionalProperties
dc#description hasValue "Sidewalk concept definition"
endNonFunctionalProperties

concept GangPlank subConceptOf Footpath
nonFunctionalProperties
dc#description hasValue "GangPlank concept definition"
endNonFunctionalProperties

concept PathThroughAir subConceptOf VehicularRoute
nonFunctionalProperties
dc#description hasValue "PathThroughAir concept definition"
endNonFunctionalProperties

concept AirLane subConceptOf PathThroughAir
nonFunctionalProperties
dc#description hasValue "AirLane concept definition"
endNonFunctionalProperties

concept GlideSlope subConceptOf PathThroughAir
nonFunctionalProperties
dc#description hasValue "GlideSlope concept definition"
endNonFunctionalProperties

concept IndicativeRouteType
nonFunctionalProperties
dc#description hasValue "IndicativeRouteType concept definition"
endNonFunctionalProperties

value ofType string

concept RouteSpecification
nonFunctionalProperties
dc#description hasValue "RouteSpecification concept definition"
endNonFunctionalProperties

hasSpecification ofType (1 *) RouteSpecificationType

concept RouteSpecificationType
nonFunctionalProperties
dc#description hasValue "RouteSpecificationType concept definition"
endNonFunctionalProperties

hasPoint ofType (1 *) GeoLocation

hasOrder ofType (1 1) nonNegativeInteger

concept RouteSpecification
nonFunctionalProperties
dc#description hasValue "RouteSpecification concept definition"
endNonFunctionalProperties

hasNthRoutePoint ofType (1 *) NthRoutePoint

concept NthRoutePoint
nonFunctionalProperties
dc#description hasValue "NthRoutePoint concept definition"
endNonFunctionalProperties

hasPoint ofType (1) GeoLocation
hasOrder ofType (1) nonNegativeInteger

concept Region subConceptOf LocativeEntity
nonFunctionalProperties
dc#description hasValue "Region concept definition"
endNonFunctionalProperties

hasName ofType (1) string
hasSpecification ofType RegionSpecification

concept GeopoliticalPlace subConceptOf GeopoliticalPlace
nonFunctionalProperties
dc#description hasValue "GeopoliticalPlace concept definition"
endNonFunctionalProperties

hasName ofType (1) string
hasSpecification ofType RegionSpecification

concept GeographicalRegion subConceptOf { GeographicalPlace, Region }
nonFunctionalProperties
dc#description hasValue "GeographicalRegion concept definition"
endNonFunctionalProperties

concept GeopoliticalRegion subConceptOf GeopoliticalRegion
nonFunctionalProperties
dc#description hasValue "GeopoliticalRegion concept definition"
endNonFunctionalProperties

concept NationalTerritory subConceptOf GeopoliticalRegion
nonFunctionalProperties
dc#description hasValue "GeopoliticalRegion concept definition"
endNonFunctionalProperties

concept HumanResidenceArea subConceptOf HumanResidenceArea
nonFunctionalProperties
dc#description hasValue "HumanResidenceArea concept definition"
endNonFunctionalProperties

concept UrbanArea subConceptOf HumanResidenceArea
nonFunctionalProperties
dc#description hasValue "UrbanArea concept definition"
endNonFunctionalProperties

concept Neighborhood subConceptOf HumanResidenceArea
nonFunctionalProperties
dc#description hasValue "Neighborhood concept definition"
endNonFunctionalProperties

concept SuburbanArea subConceptOf HumanResidenceArea
nonFunctionalProperties
dc#description hasValue "SuburbanArea concept definition"
endNonFunctionalProperties

concept NationalTerritory subConceptOf GeopoliticalRegion
nonFunctionalProperties
dc#description hasValue "GeopoliticalRegion concept definition"
endNonFunctionalProperties

concept SubnationalTerritory subConceptOf GeopoliticalRegion
nonFunctionalProperties
dc#description hasValue "SubnationalTerritory concept definition"
endNonFunctionalProperties

concept CityTerritory subConceptOf SubnationalTerritory
nonFunctionalProperties
dc#description hasValue "CityTerritory concept definition"
endNonFunctionalProperties

concept CapitolTerritory subConceptOf CityTerritory
nonFunctionalProperties
dc#description hasValue "CapitolTerritory concept definition"
endNonFunctionalProperties

concept PrimarySubnationalTerritory subConceptOf SubnationalTerritory
nonFunctionalProperties
dc#description hasValue "PrimarySubnationalTerritory concept definition; e.g. State, Province"
endNonFunctionalProperties

concept SecondarySubnationalTerritory subConceptOf SubnationalTerritory
nonFunctionalProperties
dc#description hasValue "SecondarySubnationalTerritory concept definition; e.g. Country, Parish"
endNonFunctionalProperties
concept PostalCodeArea subConceptOf SubnationalTerritory
  nonFunctionalProperties
  dc#description hasValue "PostalCodeArea concept definition"
endNonFunctionalProperties

concept SchoolDistrictTerritory subConceptOf SubnationalTerritory
  nonFunctionalProperties
  dc#description hasValue "SchoolDistrictTerritory concept definition"
endNonFunctionalProperties

concept ControlledLand subConceptOf GeopoliticalRegion
  nonFunctionalProperties
  dc#description hasValue "ControlledLand concept definition"
endNonFunctionalProperties

concept ColonialTerritory subConceptOf ControlledLand
  nonFunctionalProperties
  dc#description hasValue "ColonialTerritory concept definition"
endNonFunctionalProperties

concept DominionTerritory subConceptOf ControlledLand
  nonFunctionalProperties
  dc#description hasValue "DominionTerritory concept definition"
endNonFunctionalProperties

concept OccupiedTerritory subConceptOf ControlledLand
  nonFunctionalProperties
  dc#description hasValue "OccupiedTerritory concept definition"
endNonFunctionalProperties

concept BodyOfLand subConceptOf GeographicalRegion
  nonFunctionalProperties
  dc#description hasValue "BodyOfLand concept definition"
endNonFunctionalProperties

concept Continent subConceptOf BodyOfLand
  nonFunctionalProperties
  dc#description hasValue "Continent concept definition"
endNonFunctionalProperties

concept Subcontinent subConceptOf BodyOfLand
  nonFunctionalProperties
  dc#description hasValue "Subcontinent concept definition"
endNonFunctionalProperties

concept Island subConceptOf BodyOfLand
  nonFunctionalProperties
  dc#description hasValue "Island concept definition"
endNonFunctionalProperties

concept Archipelago subConceptOf BodyOfLand
  nonFunctionalProperties
  dc#description hasValue "Archipelago concept definition"
endNonFunctionalProperties

concept LandTopographicalFeature subConceptOf GeographicalRegion
  nonFunctionalProperties
  dc#description hasValue "LandTopographicalFeature concept definition"
endNonFunctionalProperties

concept Peninsula subConceptOf LandTopographicalFeature
  nonFunctionalProperties
  dc#description hasValue "Peninsula concept definition"
endNonFunctionalProperties

concept Isthmus subConceptOf LandTopographicalFeature
  nonFunctionalProperties
  dc#description hasValue "Isthmus concept definition"
endNonFunctionalProperties

concept Plateau subConceptOf LandTopographicalFeature
  nonFunctionalProperties
  dc#description hasValue "Plateau concept definition"
endNonFunctionalProperties

concept MountainRange subConceptOf LandTopographicalFeature
  nonFunctionalProperties
  dc#description hasValue "MountainRange concept definition"
endNonFunctionalProperties
concept Desert subConceptOf LandEcologicalRegion
nonFunctionalProperties
dc#description hasValue "Desert concept definition"
endNonFunctionalProperties

concept Glacier subConceptOf LandEcologicalRegion
nonFunctionalProperties
dc#description hasValue "Glacier concept definition"
endNonFunctionalProperties

concept Wetland subConceptOf LandEcologicalRegion
nonFunctionalProperties
dc#description hasValue "Wetland concept definition"
endNonFunctionalProperties

concept LandEcologicalRegion subConceptOf GeographicalRegion
nonFunctionalProperties
dc#description hasValue "LandEcologicalRegion concept definition"
endNonFunctionalProperties

concept Forest subConceptOf LandEcologicalFeature
nonFunctionalProperties
dc#description hasValue "Forest concept definition"
endNonFunctionalProperties

concept RainForest subConceptOf Forest
nonFunctionalProperties
dc#description hasValue "RainForest concept definition"
endNonFunctionalProperties

concept Savannah subConceptOf LandEcologicalFeature
nonFunctionalProperties
dc#description hasValue "Savannah concept definition"
endNonFunctionalProperties

concept Steppe subConceptOf LandEcologicalFeature
nonFunctionalProperties
dc#description hasValue "Steppe concept definition"
endNonFunctionalProperties

concept BodyOfWater subConceptOf GeographicalRegion
nonFunctionalProperties
dc#description hasValue "BodyOfWater concept definition"
endNonFunctionalProperties

concept Ocean subConceptOf BodyOfWater
nonFunctionalProperties
dc#description hasValue "Ocean concept definition"
endNonFunctionalProperties

concept Lake subConceptOf BodyOfWater
nonFunctionalProperties
dc#description hasValue "Lake concept definition"
endNonFunctionalProperties

concept WaterStream subConceptOf BodyOfWater
nonFunctionalProperties
dc#description hasValue "WaterStream concept definition"
endNonFunctionalProperties

concept River subConceptOf WaterStream
nonFunctionalProperties
dc#description hasValue "River concept definition"
endNonFunctionalProperties

concept Creek subConceptOf WaterStream
nonFunctionalProperties
dc#description hasValue "Creek concept definition"
endNonFunctionalProperties

concept PartialBodyOfWater subConceptOf BodyOfWater
nonFunctionalProperties
dc#description hasValue "PartialBodyOfWater concept definition"
endNonFunctionalProperties

concept BayGulf subConceptOf PartialBodyOfWater
nonFunctionalProperties
dc#description hasValue "BayGulf concept definition"
endNonFunctionalProperties

concept Sea subConceptOf PartialBodyOfWater
nonFunctionalProperties
  dc#description hasValue "Sea concept definition"
endNonFunctionalProperties

concept RegionSpecification
  nonFunctionalProperties
    dc#description hasValue "RegionSpecification concept definition"
  endNonFunctionalProperties
  hasSpecification ofType (3 *) NthBorderPoint
  numberOfBorderPoints ofType nonNegativeInteger

concept NthBorderPoint
  nonFunctionalProperties
    dc#description hasValue "NthBorderPoint concept definition"
  endNonFunctionalProperties
  hasPoint ofType (1) GeoLocation
  hasOrder ofType (1) integer

concept Address subConceptOf LocativeEntity
  nonFunctionalProperties
    dc#description hasValue "Address concept definition"
  endNonFunctionalProperties
  hasCountry ofType (0 1) Country
  hasCountrySubdivision ofType (0 1) PrimarySubnationalTerritory
  hasCountrySubSubdivision ofType (0 1) SecondarySubnationalTerritory
  hasTown ofType (0 1) CityTerritory /includes village, town, ...
  hasSubTown ofType (0 *) GeopoliticalRegion
  hasTerritory ofType (0 1) ControlledLand
  haspostcode ofType (0 1) string
  hasAddress ofType (0 1) Addresser
  inSupraNationalRegion ofType (0 1) GeographicalRegion

concept PostBoxAddress subConceptOf Address
  nonFunctionalProperties
    dc#description hasValue "Postbox Address"
  endNonFunctionalProperties
  hasPostBoxNumber ofType string

concept StreetAddress subConceptOf { Address, GeographicalRegion }
  nonFunctionalProperties
    dc#description hasValue "Street Address"
  endNonFunctionalProperties
  hasStreetType ofType string
  hasStreetName ofType string
  hasStreetNumber ofType string
  hasStreetDirectionReference ofType StreetDirectionReference
  hasProximity ofType Proximity
carrierInstructions ofType string

concept InternalAddress subConceptOf StreetAddress
  nonFunctionalProperties
    dc#description hasValue "Address internal to a StreetAddress"
  endNonFunctionalProperties
  hasBuildingID ofType string
  hasLevel ofType string
  hasUnitID ofType string
  hasRoomNumber ofType string
  mailStop ofType string
  internalRoutingInstructions ofType string

concept PostBoxType
  nonFunctionalProperties
    dc#description hasValue "PostBoxType concept definition"
  endNonFunctionalProperties
  value ofType string

relation spatiallyRelated ( ofType LocativeEntity, ofType LocativeEntity, impliesType boolean)
relation near ( ofType LocativeEntity, ofType LocativeEntity, impliesType boolean) subRelationOf
  spatiallyRelated
relation adjacentTo ( ofType LocativeEntity, ofType LocativeEntity, impliesType boolean) subRelationOf
  near
relation touching ( ofType LocativeEntity, ofType LocativeEntity, impliesType boolean) subRelationOf near
relation inGeneric ( ofType LocativeEntity, ofType LocativeEntity, impliesType boolean) subRelationOf
  spatiallyRelated
relation inPartially ( ofType LocativeEntity, ofType LocativeEntity, impliesType boolean) subRelationOf
  inGeneric
relation inAmong (ofType LocativeEntity, ofType LocativeEntity, impliesType boolean) subRelationOf inGeneric

relation inSurrounded (ofType LocativeEntity, ofType LocativeEntity, impliesType boolean) subRelationOf inGeneric

relation inEmbedded (ofType LocativeEntity, ofType LocativeEntity, impliesType boolean) subRelationOf inSurrounded

relation subRegions (ofType LocativeEntity, ofType LocativeEntity, impliesType boolean) subRelationOf inGeneric

relation borderSubRegions (ofType LocativeEntity, ofType LocativeEntity, impliesType boolean) subRelationOf subRegions

relation internalSubRegions (ofType LocativeEntity, ofType LocativeEntity, impliesType boolean) subRelationOf (subRegions, inSurrounded)

relation internalParts (ofType LocativeEntity, ofType LocativeEntity, impliesType boolean) subRelationOf inGeneric

relation aboveGeneric (ofType LocativeEntity, ofType LocativeEntity, impliesType boolean) subRelationOf aboveGenerally

relation aboveHigherThan (ofType LocativeEntity, ofType LocativeEntity, impliesType boolean) subRelationOf aboveGeneric

relation northOf (ofType LocativeEntity, ofType LocativeEntity, impliesType boolean) subRelationOf spatiallyRelated

relation southOf (ofType LocativeEntity, ofType LocativeEntity, impliesType boolean) subRelationOf spatiallyRelated

relation eastOf (ofType LocativeEntity, ofType LocativeEntity, impliesType boolean) subRelationOf spatiallyRelated

relation westOf (ofType LocativeEntity, ofType LocativeEntity, impliesType boolean) subRelationOf spatiallyRelated

relation levelWith (ofType LocativeEntity, ofType LocativeEntity, impliesType boolean) subRelationOf spatiallyRelated

concept Address
  nonFunctionalProperties
    dc#description hasValue "Address concept definition"
  endNonFunctionalProperties
    hasDepartmentName ofType (0 1) string
    hasName ofType (0 1) string
    hasFunctionalTitle ofType (0 1) string
    hasProfessionalTitle ofType (0 1) string
    hasOrganizationName ofType (0 1) string

concept PhoneNumber
  subConceptOf LocativeEntity
  nonFunctionalProperties
    dc#description hasValue "PhoneNumber concept definition"
  endNonFunctionalProperties
    hasType ofType (1 1) PhoneLineType
    hasInteractionType ofType (1 1) PhoneNumberInteractionType
tollFreeCallForCallersFromRegion ofType Region ofType (0 1) Region
    hasCountryCode ofType (0 1) nonNegativeInteger
    hasNationalDirectDial Prefix ofType (0 1) nonNegativeInteger
    hasCityOrAreaCode ofType (0 1) nonNegativeInteger
    hasLocalNumber ofType (1 1) nonNegativeInteger
    hasInternationalPrefix ofType InternationalPrefixForRegion

concept PhoneLineType
  nonFunctionalProperties
    dc#description hasValue "PhoneLineType concept definition"
  endNonFunctionalProperties

instance MobileNumberType memberOf PhoneLineType
instance CellNumberType memberOf PhoneLineType
instance FixedLineNumberType memberOf PhoneLineType
Listing A.3: Temporal Ontology

```xml
wsmlVariant _http://www.wsmo.org/wsml/wsml-syntax/wsml-rule_

namespace {
  http://www.wsmo.org/ontologies/nfp/temporalNFPOntology#,
  dc _http://purl.org/dc/elements/1.1#,
  xsd _http://www.w3.org/2001/XMLSchema#,
  wsml _http://www.wsmo.org/wsml/wsml-syntax#,
  loc _http://www.wsmo.org/ontologies/nfp/locativeNFPOntology#,
  uom _http://www.wsmo.org/ontologies/nfp/measuresNFPOntology#,
  }

ontology _http://www.wsmo.org/ontologies/nfp/temporalNFPOntology_ #dc
definition hasValue "Temporal Ontology" #dc
definition hasValue "DERI Innsbruck" #dc
definition hasValue "Ioan Tomâ", "Doug Foxvog" #dc
definition hasValue "2006–05–06" #dc
definition hasValue _http://www.wsmo.org/2004/d2#ontologies_ #dc
definition hasValue "text/html" #dc
definition hasValue _http://www.wsmo.org/ontologies/nfp/temporalNFPOntology_ #dc
definition hasValue "en-US" #dc
wsmlVersion hasValue "Revision : 1.3" #dc
definition hasValue "Non negative integer" #dc
definition hasValue "Time", "Date" #dc
definition hasValue "Non negative integer" #dc

class TemporalEntity

class TemporalOntology

class TemporalEntity

class Time

class Date

class Instant

class Interval

class TemporalQuantity

class TemporalMath

class Temporal

class TemporalBehavior

class TemporalCapacities

class TemporalRequirements

class TemporalContract

class TemporalContractualRelation

class TemporalConfiguration

class TemporalConfigurationPlan

class TemporalConfigurationPlanElement

class TemporalConfigurationPlanElementType

endNonFunctionalProperties
```

D28.4 – NON-FUNCTIONAL PROPERTIES IN WEB SERVICES
nonFunctionalProperties
dc#description hasValue "Any time interval, duration, or temporally restricted type of time interval."
endNonFunctionalProperties
hasName ofType string

concept TimeInterval subConceptOf TemporalEntity
nonFunctionalProperties
dc#description hasValue "Time interval, may include temporal gaps"
endNonFunctionalProperties
startTime ofType (1 1) TimePoint
duration ofType (1 1) TemporalDuration
ofMillenium ofType (0 +) CalendarMillenium
ofCentury ofType (0 +) CalendarCentury
ofDecade ofType (0 +) CalendarDecade
ofYear ofType (0 +) CalendarYear
ofMonth ofType (0 +) CalendarMonth
ofDay ofType (0 +) CalendarDay
ofHour ofType (0 +) HourOfDay
ofMinute ofType (0 +) MinuteOfHour
ofSecond ofType (0 +) SecondOfMinute
hasTimeZone ofType (0 +) TimeZone

concept ContinuousTimeInterval subConceptOf TimeInterval
nonFunctionalProperties
dc#description hasValue "Time interval with no gap. The duration of a ContinuousTimeInterval is the difference in time between its startTime and its endTime."
endNonFunctionalProperties

concept TimePoint subConceptOf ContinuousTimeInterval
nonFunctionalProperties
dc#description hasValue "A TimePoint has the same startTime and endTime"
dc#relation hasValue validMillisecond
endNonFunctionalProperties
ofMillenium ofType (1 1) CalendarMillenium
ofCentury ofType (1 1) CalendarCentury
ofDecade ofType (1 1) CalendarDecade
ofYear ofType (1 1) CalendarYear
ofMonth ofType (1 1) CalendarMonth
ofDay ofType (1 1) CalendarDay
ofHour ofType (1 1) HourOfDay
ofMinute ofType (1 1) MinuteOfHour
ofSecond ofType (1 1) SecondOfMinute
millisecondOfSecond ofType (1 1) decimal

axiom validTimePoint
definedBy
!¬ ¬tp[startTime hasValue ?st] memberOf TimePoint and ¬tp[endTime hasValue ?st].

axiom validMillisecond
definedBy
!¬ ¬tp[millisecondOfSecond hasValue ?ms] memberOf TimePoint and (greaterEqual(?ms, 1000.0) or greaterEqual(0.0, ?ms)).

concept TimeZone subConceptOf loc#region
nonFunctionalProperties
dc#description hasValue "A TimeZone is a region of the Earth's surface"
dc#relation hasValue validTimeZone1
dc#relation hasValue validTimeZone2
endNonFunctionalProperties
hasRegion ofType (1 ∗) loc#Region
hasSummerRegion ofType (1 ∗) loc#Region
hasWinterRegion ofType (1 ∗) loc#Region
hasWinterUTCOffsetInHours ofType _decimal
hasSummerUTCOffsetInHours ofType _decimal

axiom validTimeZone1
definedBy
!¬ ¬tz[hasWinterUTCOffsetInHours hasValue ?off] memberOf TimeZone and (greaterEqual(?off, 15) or greaterEqual(_decimal("−14"), ?off)).

axiom validTimeZone2
definedBy
!¬ ¬tz[hasSummerUTCOffsetInHours hasValue ?off] memberOf TimeZone and (greaterEqual(?off, 15) or greaterEqual(_decimal("−14"), ?off)).

concept TemporalDate subConceptOf ContinuousTimeInterval
nonFunctionalProperties
dc#description hasValue "A ContinuousTimeInterval is an instance of TemporalDate if and only if it has a standardly defined location on particular calendar."
endNonFunctionalProperties
concept CalendarYear subConceptOf TemporalDate
    nonFunctionalProperties
dc#description hasValue "a calendar year"
endNonFunctionalProperties
ofMillenium ofType (1 1) CalendarMillenium
ofCentury ofType (1 1) CalendarCentury
ofDecade ofType (1 1) CalendarDecade

concept CalendarMonth subConceptOf TemporalDate
    nonFunctionalProperties
dc#description hasValue "a month on a calendar"
endNonFunctionalProperties
ofMillenium ofType (1 1) CalendarMillenium
ofCentury ofType (1 1) CalendarCentury
ofDecade ofType (1 1) CalendarDecade
ofYear ofType (1 1) CalendarYear
monthNumber ofType (1 1) integer

concept January subConceptOf CalendarMonth
    nonFunctionalProperties
dc#relation hasValue validJanuary
endNonFunctionalProperties

concept February subConceptOf CalendarMonth
    nonFunctionalProperties
dc#relation hasValue validFebruary
endNonFunctionalProperties

concept March subConceptOf CalendarMonth
    nonFunctionalProperties
dc#relation hasValue validMarch
endNonFunctionalProperties

concept April subConceptOf CalendarMonth
    nonFunctionalProperties
dc#relation hasValue validApril
endNonFunctionalProperties

concept May subConceptOf CalendarMonth
    nonFunctionalProperties
dc#relation hasValue validMay
endNonFunctionalProperties

concept June subConceptOf CalendarMonth
    nonFunctionalProperties
dc#relation hasValue validJune
endNonFunctionalProperties

concept July subConceptOf CalendarMonth
    nonFunctionalProperties
dc#relation hasValue validJuly
endNonFunctionalProperties

concept August subConceptOf CalendarMonth
    nonFunctionalProperties
dc#relation hasValue validAugust
endNonFunctionalProperties

concept September subConceptOf CalendarMonth
    nonFunctionalProperties
dc#relation hasValue validSeptember
endNonFunctionalProperties

concept October subConceptOf CalendarMonth
    nonFunctionalProperties
dc#relation hasValue validOctober
endNonFunctionalProperties

concept November subConceptOf CalendarMonth
    nonFunctionalProperties
dc#relation hasValue validNovember
endNonFunctionalProperties

concept December subConceptOf CalendarMonth
    nonFunctionalProperties
dc#relation hasValue validDecember
endNonFunctionalProperties

axiom validJanuary definedBy
– NON-FUNCTIONAL PROPERTIES IN WEB SERVICES

\[\text{axiom validFebruary definedBy} \]
\[\text{axiom validMarch definedBy} \]
\[\text{axiom validApril definedBy} \]
\[\text{axiom validMay definedBy} \]
\[\text{axiom validJune definedBy} \]
\[\text{axiom validJuly definedBy} \]
\[\text{axiom validAugust definedBy} \]
\[\text{axiom validSeptember definedBy} \]
\[\text{axiom validOctober definedBy} \]
\[\text{axiom validNovember definedBy} \]
\[\text{axiom validDecember definedBy} \]

\[\text{concept CalendarDay subConceptOf TemporalDate nonFunctionalProperties} \]
\[\text{dc#description hasValue "OrdinalDate concept definition" dc#relation hasValue validDayOfMonth} \]
\[\text{endNonFunctionalProperties ofMillenium ofType (1 1) CalendarMillenium} \]
\[\text{ofCentury ofType (1 1) CalendarCentury} \]
\[\text{ofDecade ofType (1 1) CalendarDecade} \]
\[\text{ofYear ofType (1 1) CalendarYear} \]
\[\text{ofMonth ofType (1 1) CalendarMonth} \]
\[\text{hasYear ofType _year} \]
\[\text{hasDayOfMonthOfType DayOfYear} \]
\[\text{hasDayOfMonthNumber ofType _gday} \]
\[\text{hasDayOfWeekNumber ofType _integer} \]
\[\text{axiom validDayOfMonth definedBy} \]
\[\text{axiom validDayOfMonth definedBy} \]
\[\text{axiom validDayOfMonth definedBy} \]
\[\text{axiom validDayOfMonth definedBy} \]
\[\text{axiom validDayOfMonth definedBy} \]
\[\text{axiom validDayOfMonth definedBy} \]
and ?m memberOf February and greaterThan(?n, 29).

axiom validDayOfAprilJuneSeptemberOrNovember
definedBy
→ _[dayOfMonth hasValue ?n] memberOf CalendarDay and
?m memberOf ThirtyDayMonth and greaterThan(?n, 30).

axiom validDayOfAprilJuneSeptemberOrNovember
definedBy
→ _[dayOfMonth hasValue ?n] memberOf CalendarDay and
(greaterEqual(?n, 30) or greaterEqual(1, ?n)).

axiom validDayOfWeekNumber
definedBy
→ _[hasDayOfWeekNumber hasValue ?n] memberOf CalendarDay and
(lessEqual(7, ?n) or greaterEqual(1, ?n)).

concept Monday subConceptOf CalendarDay
nonFunctionalProperties
dc#relation hasValue validMonday
endNonFunctionalProperties

calendar Tuesday subConceptOf CalendarDay
nonFunctionalProperties
dc#relation hasValue validT uesday
endNonFunctionalProperties

calendar Wednesday subConceptOf CalendarDay
nonFunctionalProperties
dc#relation hasValue validWednesday
endNonFunctionalProperties

calendar Thursday subConceptOf CalendarDay
nonFunctionalProperties
dc#relation hasValue validThursday
endNonFunctionalProperties

calendar Friday subConceptOf CalendarDay
nonFunctionalProperties
dc#relation hasValue validFriday
endNonFunctionalProperties

calendar Saturday subConceptOf CalendarDay
nonFunctionalProperties
dc#relation hasValue validSaturday
endNonFunctionalProperties

calendar Sunday subConceptOf CalendarDay
nonFunctionalProperties
dc#relation hasValue validSunday
endNonFunctionalProperties

axiom validMonday
definedBy
→ _[hasDayOfWeekNumber hasValue ?n] memberOf Monday and
(greaterEqual(?n, 1) or greaterEqual(1, ?n)).

axiom validTuesday
definedBy
→ _[hasDayOfWeekNumber hasValue ?n] memberOf Tuesday and
(greaterEqual(?n, 2) or greaterEqual(2, ?n)).

axiom validWednesday
definedBy
→ _[hasDayOfWeekNumber hasValue ?n] memberOf Wednesday and
(greaterEqual(?n, 3) or greaterEqual(3, ?n)).

axiom validThursday
definedBy
→ _[hasDayOfWeekNumber hasValue ?n] memberOf Thursday and
(greaterEqual(?n, 4) or greaterEqual(4, ?n)).

axiom validFriday
definedBy
→ _[hasDayOfWeekNumber hasValue ?n] memberOf Friday and
(greaterEqual(?n, 5) or greaterEqual(5, ?n)).

axiom validSaturday
definedBy
→ _[hasDayOfWeekNumber hasValue ?n] memberOf Saturday and
(greaterEqual(?n, 6) or greaterEqual(6, ?n)).
axiom validSunday
  definedBy
  ![→ ](-[J]hasDayOfWeekNumber hasValue ?n] memberOf Sunday and
  (greaterEqual(?n, 7) or greaterEqual(7, ?n)).

concept TemporalDuration subConceptOf {TemporalEntity, uom#ScalarValue}
  nonFunctionalProperties
dc#description hasValue "Temporal Duration"
endNonFunctionalProperties
hasTemporalGranularity oType (1 1) TemporalGranularity

concept TemporalGranularity subConceptOf uom#Granularity
  nonFunctionalProperties
dc#description hasValue "Temporal Granularity — unit of measure for temporal duration"
endNonFunctionalProperties
hasUnits ofType UnitOfDuration

instance MonthsGranularity memberOf TemporalGranularity
  hasUnits hasValue DaysDuration
  amount hasValue 1.5

instance OneSecondGranularity memberOf TemporalGranularity
  hasUnits hasValue SecondsDuration
  amount hasValue 1.0

instance SeveralSecondsGranularity memberOf TemporalGranularity
  hasUnits hasValue SecondsDuration
  amount hasValue 10.0

instance OneMinuteGranularity memberOf TemporalGranularity
  hasUnits hasValue MinutesDuration
  amount hasValue 1.0

instance YearsDuration memberOf UnitOfDuration
  hasTemporalGranularity hasValue OneSecondGranularity

instance MonthsDuration memberOf UnitOfDuration
  hasTemporalGranularity hasValue OneSecondGranularity

instance WeeksDuration memberOf UnitOfDuration
  hasTemporalGranularity hasValue OneSecondGranularity

instance DaysDuration memberOf UnitOfDuration
  hasTemporalGranularity hasValue OneSecondGranularity

instance HoursDuration memberOf UnitOfDuration
  hasTemporalGranularity hasValue OneSecondGranularity

instance MinutesDuration memberOf UnitOfDuration
  hasTemporalGranularity hasValue OneSecondGranularity

instance SecondsDuration memberOf UnitOfDuration

instance MillisecondsDuration memberOf UnitOfDuration

instance DecadesDuration memberOf UnitOfDuration
  hasTemporalGranularity hasValue OneSecondGranularity

instance CenturysDuration memberOf UnitOfDuration
  hasTemporalGranularity hasValue SeveralSecondsGranularity

instance MilleniasDuration memberOf UnitOfDuration
  hasTemporalGranularity hasValue OneSecondGranularity

relationInstance uom#unitMultiplicationFactor(MillisecondsDuration, SecondsDuration, 1000.0)
relationInstance uom#unitMultiplicationFactor(SecondsDuration, MinutesDuration, 60.0)
relationInstance uom#unitMultiplicationFactor(MinutesDuration, HoursDuration, 60.0)
relationInstance uom#unitMultiplicationFactor(HoursDuration, DaysDuration, 24.0)
relationInstance uom#unitMultiplicationFactor(DaysDuration, WeeksDuration, 7.0)
relationInstance uom#unitMultiplicationFactor(WeeksDuration, MonthsDuration, 4.33)
relationInstance uom#unitMultiplicationFactor(MonthsDuration, DecadesDuration, 10.0)
relationInstance uom#unitMultiplicationFactor(MonthsDuration, CenturysDuration, 32.0)
relationInstance uom#unitMultiplicationFactor(CenturysDuration, MilleniasDuration, 10.0)

axiom validMonthNumber
  definedBy
  ![→ ](-[J]hasMonthNumber hasValue ?n] memberOf ?month and
  ?month(hasMonthNumber hasValue ?m] and rEqual(?n, ?m).
axiom validDayOfWeek
  definedBy
  ! − ?j[hasDayOfWeekNumber hasValue ?n] memberOf ?day and
  ?day[dayOfWeekNumber hasValue ?d] and nEqual(?n, d).

concept TimeZone
  nonFunctionalProperties
  dc#description hasValue "TimeZone concept definition"
  endNonFunctionalProperties
  hasRegion ofType (1 *) loc#Region
  hasUTCOffset ofType _decimal

axiom validUTCOffset
  definedBy
  !− ?x[hasUTCOffset hasValue ?V] memberOf TimeZone and
  (lessEqual(15, ?V) or greaterEqual(decimal("−15"), ?V)).

concept DayOfYear
  nonFunctionalProperties
  dc#description hasValue "DayOfYear concept definition"
  dc#relation hasValue validDayOfYear
  endNonFunctionalProperties
  value ofType _integer

axiom validDayOfYear
  definedBy
  !− ?x[value hasValue ?V] memberOf DayOfYear and
  (lessEqual(366, ?V) or greaterThan(1, ?V)).

concept WeekDate
  subConceptOf TemporalDate
  nonFunctionalProperties
  dc#description hasValue "WeekDate concept definition"
  endNonFunctionalProperties
  hasWeekNumber ofType WeekNumber
  hasDayOfWeekNumber ofType DayOfWeekNumber

concept WeekNumber
  nonFunctionalProperties
  dc#description hasValue "WeekNumber concept definition"
  dc#relation hasValue validWeekNumber
  endNonFunctionalProperties
  value ofType _integer

axiom validWeekNumber
  definedBy
  !− ?x[value hasValue ?V] memberOf WeekNumber and
  (lessEqual(54, ?V) or greaterEqual(1, ?V)).

concept TemporalGranularity
  nonFunctionalProperties
  dc#description hasValue "TemporalGranularity concept definition"
  endNonFunctionalProperties
  hasName ofType _string

concept UserDefinedTemporalGranularity
  nonFunctionalProperties
  dc#description hasValue "UserDefinedTemporalGranularity concept definition"
  endNonFunctionalProperties

concept StandardTemporalGranularity
  nonFunctionalProperties
  dc#description hasValue "StandardTemporalGranularity concept definition"
  endNonFunctionalProperties

concept HourOfDay
  nonFunctionalProperties
  dc#description hasValue "Hours concept definition"
  dc#relation hasValue validHour
  endNonFunctionalProperties
  value ofType _integer

axiom validHour
  definedBy
  !− ?x[value hasValue ?V] memberOf Hours and
  (lessEqual(23, ?V) or greaterThan(0, ?V)).

concept MinuteOfHour
  nonFunctionalProperties
  dc#description hasValue "Minutes concept definition"
  dc#relation hasValue validMinute
  endNonFunctionalProperties
  value ofType _integer

axiom validMinute
**Listing A.4: Availability Ontology**

```xml
wsmlVariant "http://www.wsmo.org/wsml/wsml−syntax/wsml−rule"

namespace {
    dc "http://purl.org/dc/elements/1.1#",
    xsd "http://www.w3.org/2001/XMLSchema#",
    wsmi "http://www.wsmo.org/wsml/wsml−syntax#",
    loc "http://www.wsmo.org/ontologies/nfp/locativeNFPOntology#",
    temp "http://www.wsmo.org/ontologies/nfp/temporalNFPOntology#",
    qua "http://www.wsmo.org/ontologies/nfp/qualityNFPOntology#"
}

ontology "http://www.wsmo.org/ontologies/nfp/availabilityNFPOntology"

nonFunctionalProperties

dc#title hasValue "Availability Ontology"

dc#creator hasValue "{Ioan Toma}"

dc#subject hasValue "Availability"

dc#description hasValue "Availability in terms of when, where, and to whom something is available"

dc#publisher hasValue "DERI Innsbruck"

dc#date hasValue "2006−05−08"

dc#type hasValue "http://www.wsmo.org/2004/d2#ontologies"

dc#format hasValue "text/html"

dc#language hasValue "en−US"

wsml#version hasValue "Revision : 1.3"

endNonFunctionalProperties

concept Availability

nonFunctionalProperties

dc#description hasValue "Availability in terms of when, where, and to whom something is available"

endNonFunctionalProperties

isAvailableAt ofType (1 ∗) loc#LocativeEntity

isAvailableDuring ofType (1 −) temp#TemporalEntity

isAvailableTo ofType (1 −) iri //available to an agent

concept RequestAvailability subConceptOf Availability

nonFunctionalProperties

dc#description hasValue "links Availability to request"

endNonFunctionalProperties

forRequest ofType (1 1) iri

hasNegotiableTime ofType (0 1) boolean

isContinuouslyAvailable ofType (0 1) boolean
```

**Listing A.5: Obligation Ontology**

```xml
wsmlVariant "http://www.wsmo.org/wsml/wsml−syntax/wsml−rule"

namespace {
    dc "http://purl.org/dc/elements/1.1#",
    xsd "http://www.w3.org/2001/XMLSchema#",
    ava "http://www.wsmo.org/ontologies/nfp/availabilityNFPOntology#",
    price "http://www.wsmo.org/ontologies/nfp/priceNFPOntology#",
    pay "http://www.wsmo.org/ontologies/nfp/paymentNFPOntology#",
    loc "http://www.wsmo.org/ontologies/nfp/locativeNFPOntology#",
    temp "http://www.wsmo.org/ontologies/nfp/temporalNFPOntology#",
    right "http://www.wsmo.org/ontologies/nfp/rightsNFPOntology#",
    disc "http://www.wsmo.org/ontologies/nfp/discountsNFPOntology#"
}

ontology "http://www.wsmo.org/ontologies/nfp/obligationsNFPOntology"

nonFunctionalProperties

dc#title hasValue "Obligations Ontology"

dc#creator hasValue "{Ioan Toma}"

dc#subject hasValue "Obligations"

dc#description hasValue ""Obligations Ontology"

dc#publisher hasValue "DERI Innsbruck"

dc#date hasValue "2006−05−08"

dc#type hasValue "http://www.wsmo.org/2004/d2#ontologies"

dc#format hasValue "text/html"

dc#language hasValue "en−US"

wsml#version hasValue "Revision : 1.3"

endNonFunctionalProperties

concept Obligation

nonFunctionalProperties

dc#description hasValue "definition of an obligation"

endNonFunctionalProperties

isObligationOf ofType (1 +) ava#Availability

isDueDuring ofType (1 −) temp#TemporalEntity

isDueTo ofType (1 −) iri //available to an agent

concept RequestObligation subConceptOf Obligation

nonFunctionalProperties

dc#description hasValue "links Obligation to request"

endNonFunctionalProperties

forRequest ofType (1 1) iri

hasNegotiableTime ofType (0 1) boolean

isContinuouslyAvailable ofType (0 1) boolean
```
ontology
"http://www.wsmo.org/ontologies/nfp/obligationsNFPOntology"

nonFunctionalProperties
dc#title hasValue "Obligations Ontology"
dc#creator hasValue "Ioan Toma"
dc#subject hasValue {"Obligation"}
dc#description hasValue "Obligations Ontology"
dc#publisher hasValue "DERI Innsbruck"
dc#date hasValue "2006-05-08"
dc#type hasValue "http://www.wsmo.org/2004/d2#ontologies"
dc#format hasValue "text/html"
dc#identifier hasValue "http://www.wsmo.org/ontologies/nfp/obligationsNFPOntology"
dc#language hasValue "en-US"

endNonFunctionalProperties

concept Obligation
nonFunctionalProperties
dc#description hasValue "Obligation concept definition"
endNonFunctionalProperties

forProvider ofType (1 1) iri //Provider

relation ProviderObligation ( ofType _iri, ofType Obligation)

concept RelationshipObligation subConceptOf Obligation
nonFunctionalProperties
dc#description hasValue "RelationshipObligation concept definition"
endNonFunctionalProperties

hasCondition ofType _iri //Condition
hasDuration ofType (1 1) temp#TemporalDuration
isRequiredToReceive ofType (0 1) price#PriceObligation
hasAssociatedRight ofType (0 1) right#Right

concept Membership subConceptOf RelationshipObligation
nonFunctionalProperties
dc#description hasValue "Membership concept definition"
endNonFunctionalProperties

hasMembershipLevelCharge ofType (1 1) price#PricingObligation
hasMembershipLevel ofType (1 1) _string

concept PaymentObligation subConceptOf Obligation
nonFunctionalProperties
dc#description hasValue "PaymentObligation concept definition"
endNonFunctionalProperties

hasCharge ofType (0 1) PriceObligation
requiresPaymentDepositOrBond ofType (1 1) price#PriceObligation
requiresEstablishmentFee ofType (0 1) price#PriceObligation
hasPaymentDiscount ofType (0 1) disc#PaymentDiscount
hasPaymentObligationCondition ofType (0 1) _iri //Condition
hasInterestFreePeriod ofType (0 1) temp#TemporalDuration
hasInterestCharge ofType (0 1) TemporalGranularityPricingObligation
hasAdministrativeCharge ofType (0 1) TemporalGranularityPricingObligation
canBeExecutedWithInsuranceService ofType _iri //IRI of the Insurance Service
canBeExecutedWithEscrowService ofType _iri //IRI of the Escrow Service

concept TemporalGranularityPricingObligation
nonFunctionalProperties
dc#description hasValue "emporalGranularityPricingObligation concept definition"
endNonFunctionalProperties

hasTemporalGranularity ofType (1 1) temp#TemporalGranularity
hasPricingObligation ofType (1 1) PricingObligation

concept PricingObligation subConceptOf Obligation
nonFunctionalProperties
dc#description hasValue "PricingObligation concept definition"
endNonFunctionalProperties

resultsInPrice ofType ServicePrice
hasPricingObligationAvailability ofType Availability
hasPricingCondition ofType _iri
hasRefundCondition ofType _iri
hasRefundProcedure ofType Procedure
hasNegotiablePrice ofType (1 1) _boolean
requiresCustomisedPricingOrQuote ofType (1 1) _boolean
relatesTo ofType PricingObligationRelated
hasAvailablePayeeDiscount ofType (1 *) PayeeDiscount

concept PricingObligationRelated

nonFunctionalProperties
Listing A.6: Price Ontology

wsmlVariant _http://www.wsmo.org/wsml/wsml−syntax/wsml−rule_

namespace { _http://www.wsmo.org/ontologies/nfp/priceNFPOntology#, 
    dc _http://purl.org/dc/elements/1.1#,
    xsd _http://www.w3.org/2001/XMLSchema#,
    wsmi _http://www.wsmo.org/wsml/wsml−syntax#,
    loc _http://www.wsmo.org/ontologies/nfp/localizeNFPOntology#,
    ava _http://www.wsmo.org/ontologies/nfp/availabilityNFPOntology#,
    cur _http://www.wsmo.org/ontologies/nfp/currencyNFPOntology#" 
}

ontology _http://www.wsmo.org/ontologies/nfp/priceNFPOntology" 

nonFunctionalProperties
dc#title hasValue "Price Ontology" 
dc#creator hasValue {"Ioan Toma"} 
dc#subject hasValue {"Price", "MechanismAuction"} 
dc#description hasValue "Price Ontology" 
dc#publisher hasValue "DERI Innsbruck" 
dc#contributor hasValue {"Ioan Toma"} 
dc#date hasValue "2006−05−06" 
dc#type hasValue _http://www.wsmo.org/2004/d2#ontologies" 
dc#format hasValue "text/html" 
dc#identifier hasValue _http://www.wsmo.org/ontologies/nfp/priceNFPOntology" 
dc#language hasValue "en−US" 
wsml#version hasValue "Revision : 1.3" 
endNonFunctionalProperties

concept Price 
    nonFunctionalProperties 
    dc#description hasValue "Price concept definition" 
endNonFunctionalProperties

concept MonetaryAmount
    hasAmount ofType decimal
    hasCurrency ofType cur#Currency

concept AbsolutePrice subConceptOf {Price, MonetaryAmount}
    nonFunctionalProperties 
    dc#relation hasValue AbsolutePriceDefinition 
endNonFunctionalProperties

axiom AbsolutePriceDefinition definedBy
1 − ?price memberOf MonetaryAmount and
?price memberOf Price and
naf (?price memberOf AbsolutePrice).

concept ProportionalPrice subConceptOf Price
    nonFunctionalProperties 
    dc#description hasValue "ProportionalPrice concept definition" 
endNonFunctionalProperties

hasPercentage ofType Percentage
hasItem ofType Item

concept RangedPrice subConceptOf Price
    nonFunctionalProperties 
    dc#description hasValue "RangedPrice concept definition" 
endNonFunctionalProperties

concept RangedAbsolutePrice subConceptOf RangedPrice
    nonFunctionalProperties 
    dc#description hasValue "RangedAbsolutePrice concept definition" 
endNonFunctionalProperties

hasRangedAbsolutePriceFrom ofType AbsolutePrice
hasRangedAbsolutePriceTo ofType AbsolutePrice

concept RangedProportionalPrice subConceptOf RangedPrice
    nonFunctionalProperties 
    dc#description hasValue "RangedProportionalPrice concept definition" 
endNonFunctionalProperties

hasRangedProportionalPriceFrom ofType ProportionalPrice
hasRangedProportionalPriceTo ofType ProportionalPrice

concept DynamicPrice subConceptOf Price
D28.4 – NON-FUNCTIONAL PROPERTIES IN WEB SERVICES

nonFunctionalProperties
dc#description hasValue "DynamicPrice concept definition"
endNonFunctionalProperties

hasMechanismProvider ofType (0 1) iri

hasIndicativePrice ofType (0 1) IndicativePrice

hasReservedPrice ofType (0 1) ReservedPrice

hasMechanismAuction ofType (0 1) MechanismAuction

hasMechanismCondition ofType iri

hasAvailability ofType (1 ∗) ava#Availability

concept IndicativePrice
nonFunctionalProperties
dc#description hasValue "IndicativePrice concept definition"
endNonFunctionalProperties

concept ReservedPrice
nonFunctionalProperties
dc#description hasValue "ReservedPrice concept definition"
endNonFunctionalProperties

concept MechanismAuction
nonFunctionalProperties
dc#description hasValue "MechanismAuction concept definition"
endNonFunctionalProperties

concept DutchAuction subConceptOf MechanismAuction
nonFunctionalProperties
dc#description hasValue "DutchAuction concept definition"
endNonFunctionalProperties

concept DoubleAuction subConceptOf MechanismAuction
nonFunctionalProperties
dc#description hasValue "DoubleAuction concept definition"
endNonFunctionalProperties

concept EnglishAuction subConceptOf MechanismAuction
nonFunctionalProperties
dc#description hasValue "EnglishAuction concept definition"
endNonFunctionalProperties

concept SealedBidAuction subConceptOf MechanismAuction
nonFunctionalProperties
dc#description hasValue "SealedBidAuction concept definition"
endNonFunctionalProperties

concept VickreyAuction subConceptOf MechanismAuction
nonFunctionalProperties
dc#description hasValue "VickreyAuction concept definition"
endNonFunctionalProperties

concept TaxRateForRegion
nonFunctionalProperties
dc#description hasValue "TaxRateForRegion concept definition"
endNonFunctionalProperties

hasTaxRate ofType Percentage

hasRegion ofType loc#Region

concept TaxItem
nonFunctionalProperties
dc#description hasValue "TaxItem concept definition"
endNonFunctionalProperties

hasTaxType ofType TaxType

hasTaxRateForRegion ofType TaxRateForRegion

concept Percentage
nonFunctionalProperties
dc#description hasValue "Percentage concept definition"
dc#relation hasValue validPercent
endNonFunctionalProperties

value ofType decimal

axiom validPercentage
definedBy
!− 7 x [value hasValue ?V] memberOf Percentage and (lessEqual(100,?V) or greaterEqual(0,?V)).

concept TaxType
nonFunctionalProperties
dc#description hasValue "TaxType concept definition"
dc#relation hasValue validTaxType
endNonFunctionalProperties
value ofType _string

concept GST subConceptOf TaxType
  nonFunctionalProperties
dc#description hasValue "GST concept definition"
endNonFunctionalProperties

concept VAT subConceptOf TaxType
  nonFunctionalProperties
dc#description hasValue "VAT concept definition"
endNonFunctionalProperties

concept StateTax subConceptOf TaxType
  nonFunctionalProperties
dc#description hasValue "StateTax concept definition"
endNonFunctionalProperties

Listing A.7: Payment Ontology

<wsmlVariant _http://www.wsmo.org/wsml/wsml-syntax/wsml-rule">
  namespace {
    _http://www.wsmo.org/ontologies/nfp/paymentNFPOntology#,
    dc _http://purl.org/dc/elements/1.1#,
    xsd _http://www.w3.org/2001/XMLSchema#,
    wsml _http://www.wsmo.org/wsml/wsml-syntax#,
    loc _http://www.wsmo.org/ontologies/nfp/locativeNFPOntology#,
    temp _http://www.wsmo.org/ontologies/nfp/temporalNFPOntology#,
    ava _http://www.wsmo.org/ontologies/nfp/availabilityNFPOntology#,
    price _http://www.wsmo.org/ontologies/nfp/priceNFPOntology#,
    cur _http://www.wsmo.org/ontologies/nfp/currencyNFPOntology#
  }

  ontology _http://www.wsmo.org/ontologies/nfp/paymentNFPOntology"
    nonFunctionalProperties
dc#title hasValue "Payment Ontology"
    dc#creator hasValue {"Ioan Toma"}
    dc#subject hasValue {"Payment"}
    dc#description hasValue "Payment Ontology -- complementar to Price Ontology"
    dc#publisher hasValue "DERI Innsbruck"
    dc#contributor hasValue {"Ioan Toma"}
    dc#date hasValue "2006−05−08"
    dc#type hasValue _http://www.wsmo.org/2004/d2#ontologies"
    dc#format hasValue "text/html"
    dc#identifier hasValue _http://www.wsmo.org/ontologies/nfp/paymentNFPOntology"
    dc#language hasValue "en−US"
    wsml#version hasValue "Revision : 1.3"
  endNonFunctionalProperties

cconcept PaymentInstrument
  nonFunctionalProperties
dc#description hasValue "PaymentInstrument concept definition"
endNonFunctionalProperties

  supportsCurrency ofType (1 *) cur#Currency
  isIssuedIn ofType (0 1) loc#Region
  isLimitedToUseIn ofType (0 1) loc#Region
  isIssuerOf ofType (0 1) iri://Provider
  hasSurcharge ofType PriceProvider
  supportPaymentScheme ofType PaymentScheme
  hasPaymentInstrumentType ofType (1 1) PaymentInstrumentType

concept PaymentScheme
  nonFunctionalProperties
dc#description hasValue "PaymentScheme concept definition"
endNonFunctionalProperties

  hasName ofType (1 1) _string
  controlledBy ofType (1 1) iri://Provider

concept PriceProvider
  nonFunctionalProperties
dc#description hasValue "PriceProvider concept definition"
endNonFunctionalProperties

concept AbsolutePriceProvider
  nonFunctionalProperties
dc#description hasValue "AbsolutePriceProvider concept definition"
endNonFunctionalProperties

  hasSurcharge ofType (1 1) price#AbsolutePrice
  hasProvider ofType (1 1) iri://Provider
</wsmlVariant>
concept ProportionalPriceProvider  
nonFunctionalProperties  
dc#description hasValue "ProportionalPriceProvider concept definition"  
endNonFunctionalProperties  
hasSurcharge ofType (1 1) price#PromotionalPrice  
hasProvider ofType (1 1) iri://Provider

concept CashInstrument subConceptOf PaymentInstrument  
nonFunctionalProperties  
dc#description hasValue "CashInstrument concept definition"  
endNonFunctionalProperties

concept ElectronicCashInstrument subConceptOf CashInstrument  
nonFunctionalProperties  
dc#description hasValue "ElectronicCashInstrument concept definition"  
endNonFunctionalProperties  
hasElectronicCashType ofType (1 1) ElectronicCashType

concept DirectDebit subConceptOf ElectronicCashType  
nonFunctionalProperties  
dc#description hasValue "DirectDebit concept definition"  
endNonFunctionalProperties

concept DirectTransfer subConceptOf ElectronicCashType  
nonFunctionalProperties  
dc#description hasValue "DirectTransfer concept definition"  
endNonFunctionalProperties

concept DigitalCash subConceptOf ElectronicCashType  
nonFunctionalProperties  
dc#description hasValue "DigitalCash concept definition"  
endNonFunctionalProperties

concept WireTransfer subConceptOf ElectronicCashType  
nonFunctionalProperties  
dc#description hasValue "WireTransfer concept definition"  
endNonFunctionalProperties

concept VoucherBasedInstrument subConceptOf PaymentInstrument  
nonFunctionalProperties  
dc#description hasValue "VoucherBasedInstrument concept definition"  
endNonFunctionalProperties  
hasTemporalValidity ofType (0 1) temp#TemporalEntity  
validForRedemptionWith ofType iri://Service or Provider

concept ChequeInstrument subConceptOf PaymentInstrument  
nonFunctionalProperties  
dc#description hasValue "ChequeInstrument concept definition"  
endNonFunctionalProperties  
hasChequeType ofType (1 1) ChequeType

concept ChequeType  
nonFunctionalProperties  
dc#description hasValue "ChequeType concept definition"  
endNonFunctionalProperties

concept PersonalChequeType subConceptOf ChequeType  
nonFunctionalProperties  
dc#description hasValue "PersonalChequeType concept definition"  
endNonFunctionalProperties

concept BankChequeType subConceptOf ChequeType  
nonFunctionalProperties  
dc#description hasValue "BankChequeType concept definition"  
endNonFunctionalProperties

concept TravelerChequeType subConceptOf ChequeType  
nonFunctionalProperties  
dc#description hasValue "TravelerChequeType concept definition"  
endNonFunctionalProperties

concept BusinessChequeType subConceptOf ChequeType  
nonFunctionalProperties  
dc#description hasValue "BusinessChequeType concept definition"  
endNonFunctionalProperties
concept CardBasedInstrument subConceptOf PaymentInstrument
  nonFunctionalProperties
dc#description hasValue "CardBasedInstrument concept definition"
endNonFunctionalProperties

concept DebitCard subConceptOf CardBasedInstrument
  nonFunctionalProperties
dc#description hasValue "DebitCard concept definition"
endNonFunctionalProperties

concept CreditCard subConceptOf CardBasedInstrument
  nonFunctionalProperties
dc#description hasValue "CreditCard concept definition"
endNonFunctionalProperties

concept StoreCard subConceptOf CardBasedInstrument
  nonFunctionalProperties
dc#description hasValue "StoreCard concept definition"
endNonFunctionalProperties

concept ChargeCard subConceptOf CardBasedInstrument
  nonFunctionalProperties
dc#description hasValue "ChargeCard concept definition"
endNonFunctionalProperties

concept StoredValueCard subConceptOf CardBasedInstrument
  nonFunctionalProperties
dc#description hasValue "StoredValueCard concept definition"
endNonFunctionalProperties

concept PaymentOption
  nonFunctionalProperties
dc#description hasValue "PaymentOption concept definition"
endNonFunctionalProperties

hasInvoiceTermsOfPayment ofType (0 1) temp#TemporalDuration
hasAssociatedCharge ofType (0 1) PricingObligation
limitedToRequestorsFromRegion ofType (0 1) loc#Region
hasPaymentOptionCondition ofType (1 1) boolean
isPreferredPaymentOption ofType (1 1) boolean
hasInstrumentLocations ofType InstrumentLocations

concept InstrumentLocations
  nonFunctionalProperties
dc#description hasValue "InstrumentLocations concept definition"
endNonFunctionalProperties

canBeFilledUsingPaymentInstrument ofType (1 1) PaymentInstrument
atPaymentLocation ofType (1 1) PaymentLocation

concept PaymentLocation
  nonFunctionalProperties
dc#description hasValue "PaymentLocation concept definition"
endNonFunctionalProperties

hasLocation ofType (1 1) loc#LocativeEntity
acceptsCombinationsOfInstruments ofType PaymentInstrumentType
requiresInPersonAttendance ofType (1 1) boolean

concept Percent
  nonFunctionalProperties
dc#description hasValue "Percent concept definition"
endNonFunctionalProperties

value ofType price#Percentage

concept PercentTemporalProvision subConceptOf Percent
  nonFunctionalProperties
dc#description hasValue "PercentTemporalProvision concept definition"
endNonFunctionalProperties

accordingToTemporalPattern ofType (1 1) temp#TemporalEntity

concept PercentServiceProvision subConceptOf Percent
  nonFunctionalProperties
dc#description hasValue "PercentServiceProvision concept definition"
endNonFunctionalProperties

relationshipToServiceProvision ofType (1 1) TemporalRelationship

concept TemporalRelationship
  nonFunctionalProperties
dc#description hasValue "TemporalRelationship concept definition"
endNonFunctionalProperties

/to do -- before, during, after
Listing A.8: Discounts Ontology

```xml
wsmlVariant _http://www.wsmo.org/wsml/wsml−syntax/wsml−rule"
namespace {
  dc: _http://purl.org/dc/elements/1.1#", 
  xsd _http://www.w3.org/2001/XMLSchema#", 
  ava _http://www.wsmo.org/ontologies/nfp/availabilityNFPOntology#", 
  price _http://www.wsmo.org/ontologies/nfp/priceNFPOntology#", 
  pay _http://www.wsmo.org/ontologies/nfp/paymentNFPOntology#", 
  loc _http://www.wsmo.org/ontologies/nfp/localiNFP Ontology#", 
  temp _http://www.wsmo.org/ontologies/nfp/temporal NFPOntology#"
} 

ontology _http://www.wsmo.org/ontologies/nfp/discountsNFPOntology" 
nonFunctionalProperties 
  dc#title hasValue "Discounts Ontology"
  dc#creator hasValue { Ioan Toma} 
  dc#subject hasValue ("Discount")
  dc#description hasValue ("Discounts Ontology")
  dc#publisher hasValue ("DERI Innsbruck")
  dc#contributor hasValue { Ioan Toma} 
  dc#date hasValue "2006−05−08"
  dc#type hasValue _http://www.wsmo.org/2004/d2#ontologies"
  dc#format hasValue "text/html"
  dc#identifier hasValue _http://www.wsmo.org/ontologies/nfp/discountsNFPOntology"
  dc#language hasValue "en−US"
wsml#version hasValue "Revision : 1.3"
endNonFunctionalProperties 

concept Discount 
  nonFunctionalProperties 
  dc#description hasValue "Discount concept definition"
endNonFunctionalProperties 

hasCondition ofType Condition 
hasAmount ofType DiscountAmount 
hasResultingDiscountedPrice ofType ResultingDiscountPrice 
hasAvailability ofType (1 +) ava#Availability

concept PayeeDiscount subConceptOf Discount 
nonFunctionalProperties 
  dc#description hasValue "PayeeDiscount concept definition"
endNonFunctionalProperties 
```
Listing A.9: Rights Ontology

wsmlVariant \"http://www.wsmo.org/wsml/wsml−syntax/wsml−rule\"

namespace { 
  \"http://www.wsmo.org/ontologies/nfp/rightsNFPOntology#\",
  
  concept StudentDiscount subConceptOf PayeeDiscount
  nonFunctionalProperties
dc#description hasValue "StudentDiscount concept definition"
endNonFunctionalProperties
applicableToSchoolStudents ofType (1 1) boolean
applicableToFulltimeUniversityStudents ofType (1 1) boolean

concept MembershipDiscount subConceptOf PayeeDiscount
nonFunctionalProperties
dc#description hasValue "MembershipDiscount concept definition"
endNonFunctionalProperties
isApplicableToSchoolStudents

concept ShareholderDiscount subConceptOf PayeeDiscount
nonFunctionalProperties
dc#description hasValue "ShareholderDiscount concept definition"
endNonFunctionalProperties
availableToShareholders
availableToShareholdersWithMinimumNumberOfUnits ofType (1 1) loc#nonNegativeInteger

concept AgeGroupDiscount subConceptOf PayeeDiscount
nonFunctionalProperties
dc#description hasValue "AgeGroupDiscount concept definition"
endNonFunctionalProperties
hasName ofType (0 1) string
ageFromValue ofType (1 1) loc#nonNegativeInteger
ageToValue ofType (1 1) loc#nonNegativeInteger

concept DiscountAmount
nonFunctionalProperties
dc#description hasValue "DiscountAmount concept definition"
endNonFunctionalProperties
absoluteDiscount ofType (0 1) price#MonetaryAmount
percentDiscount ofType (0 1) price#Percentage

concept ResultingDiscountedPrice subConceptOf price#Price
nonFunctionalProperties
dc#description hasValue "ResultingDiscountedPrice concept definition"
endNonFunctionalProperties

concept PaymentDiscount
nonFunctionalProperties
dc#description hasValue "PaymentDiscount concept definition"
endNonFunctionalProperties
hasMinimumPriceRequiredToReceiveDiscount ofType (0 1) price#AbsoutePrice

concept PaymentInstrumentTypeDiscount subConceptOf PaymentDiscount
nonFunctionalProperties
dc#description hasValue "PaymentInstrumentTypeDiscount concept definition"
endNonFunctionalProperties
offersPaymentInstrumentTypeDiscountFor ofType (1 1) pay#PaymentInstrumentType

concept PaymentLocationTypeDiscount subConceptOf PaymentDiscount
nonFunctionalProperties
dc#description hasValue "PaymentLocationTypeDiscount concept definition"
endNonFunctionalProperties
offersPaymentLocationTypeDiscountFor ofType (0 1) loc#LocativeEntityType

concept CouponPaymentDiscount subConceptOf PaymentDiscount
nonFunctionalProperties
dc#description hasValue "CouponPaymentDiscount concept definition"
endNonFunctionalProperties
hasValidityPeriod ofType (0 1) temp#TemporalEntity
isIssuedBy ofType (1 1) iri //"could be a person, organization or any kind of provider"

concept EarlyPaymentDiscount subConceptOf PaymentDiscount
nonFunctionalProperties
dc#description hasValue "EarlyPaymentDiscount concept definition"
endNonFunctionalProperties
hasEarlyPaymentOffset ofType temp#TemporalDuration
cutOffDate ofType (0 1) temp#TemporalDuration
dc: http://purl.org/dc/elements/1.1#,
xsd: http://www.w3.org/2001/XMLSchema#
wsml: http://www.wsmo.org/wsml/wsml-syntax#,
loc: http://www.wsmo.org/ontologies/ftp/locativeNFPOntology#,
temp: http://www.wsmo.org/ontologies/ftp/temporalNFP Ontology#,
qua: http://www.wsmo.org/ontologies/ftp/qualityNFPOntology#,
ava: http://www.wsmo.org/ontologies/ftp/availabilityNFPOntology#,
obl: http://www.wsmo.org/ontologies/ftp/obligationsNFPOntology#}

ontology: http://www.wsmo.org/ontologies/ftp/rightsNFPOntology

nonFunctionalProperties

dc:title hasValue "Rights Ontology"
dc:creator hasValue {"Ioan Tomă"}
dc:subject hasValue {"Right"}
dc:description hasValue "Rights Ontology"
dc:publisher hasValue "DERI Innsbruck"
dc:contributor hasValue {"Ioan Tomă"}
dc:date hasValue "2006−05−08"
dc:type hasValue "http://www.wsmo.org/2004/d2#ontologies"
dc:format hasValue "text/html"
dc:identifier hasValue "http://www.wsmo.org/ontologies/ftp/rightsNFPOntology"
dc:language hasValue "en−US"
wsml:version hasValue "Revision : 1.3"

endNonFunctionalProperties

concept Right

nonFunctionalProperties

dc:description hasValue "Right concept definition"
endNonFunctionalProperties

concept RightOfWarranty subConceptOf Right

nonFunctionalProperties

dc:description hasValue "RightOfWarranty concept definition"
endNonFunctionalProperties

concept WarrantyProcedureAvailability

nonFunctionalProperties

dc:description hasValue "WarrantyProcedureAvailability concept definition"
endNonFunctionalProperties

concept WarrantiedItem

nonFunctionalProperties

dc:description hasValue "WarrantiedItem concept definition"
endNonFunctionalProperties

concept RightOfAccess subConceptOf Right

nonFunctionalProperties

dc:description hasValue "RightOfAccess concept definition"
endNonFunctionalProperties

concept ObligationTemporalDuration

nonFunctionalProperties

dc:description hasValue "ObligationTemporalDuration concept definition"
endNonFunctionalProperties

concept AccessTypeResource

nonFunctionalProperties

dc:description hasValue "AccessTypeResource concept definition"
endNonFunctionalProperties

concept Resource

nonFunctionalProperties

dc:description hasValue "Resource concept definition"
endNonFunctionalProperties
hasLocation ofType (0 1) loc#LocativeEntity
hasName ofType (1 1) #string
hasType ofType (1 1) ResourceType

concept ResourceType
nonFunctionalProperties
dc#description hasValue "ResourceType concept definition"
endNonFunctionalProperties

value ofType #string

concept AccessType
nonFunctionalProperties
dc#description hasValue "AccessType concept definition"
endNonFunctionalProperties

concept ExclusiveAccessType subConceptOf AccessType
nonFunctionalProperties
dc#description hasValue "ExclusiveAccessType concept definition"
endNonFunctionalProperties

concept SharedAccessType subConceptOf AccessType
nonFunctionalProperties
dc#description hasValue "SharedAccessType concept definition"
endNonFunctionalProperties

concept RestrictedAccessType subConceptOf AccessType
nonFunctionalProperties
dc#description hasValue "RestrictedAccessType concept definition"
endNonFunctionalProperties

concept ProhibitedAccessType subConceptOf AccessType
nonFunctionalProperties
dc#description hasValue "ProhibitedAccessType concept definition"
endNonFunctionalProperties

concept RightOfExtension subConceptOf Right
nonFunctionalProperties
dc#description hasValue "RightOfExtension concept definition"
endNonFunctionalProperties

incursObligation ofType ob#Obligation
isInitiatedUsingProcedure ofType Procedure
hasTemporalDurationCondition ofType (1 ∗) TemporalDurationCondition

concept TemporalDurationCondition
nonFunctionalProperties
dc#description hasValue "TemporalDurationCondition concept definition"
endNonFunctionalProperties

isAvailableForTemporalDuration ofType (1 1) temp#TemporalDuration
underCondition ofType (1 1) iri//Condition

concept RightOfRefusal subConceptOf Right
nonFunctionalProperties
dc#description hasValue "RightOfRefusal concept definition"
endNonFunctionalProperties

refusalIsAllowed ofType (1 1) boolean
isAdministeredUnderRefusalProcedure ofType (0 1) Procedure
hasAppeal ofType Appeal

relation RightOfRefusalUnderCondition ( ofType RightOfRefusal, ofType _iri, impliesType _boolean) // RightOfRefusal, Condition

concept Appeal
nonFunctionalProperties
dc#description hasValue "Appeal concept definition"
endNonFunctionalProperties

hasProcedure ofType (1 1) Procedure
withAvailability ofType (1 1) ava#Availability

concept RightOfPrivacy subConceptOf Right
nonFunctionalProperties
dc#description hasValue "RightOfPrivacy concept definition"
endNonFunctionalProperties

isAvailableFor ofType (1 1) ava#Availability

concept Element
nonFunctionalProperties
dc#description hasValue "RightOfPrivacy applies to this Element"
endNonFunctionalProperties
concept DisclosedItem
nonFunctionalProperties
dc#description hasValue "DisclosedItem concept definition"
endNonFunctionalProperties
hasDuration ofType (1 1) temp#TemporalDuration
isSharedWithProvider ofType Provider

concept PrivacyPolicyCondition
nonFunctionalProperties
dc#description hasValue "PrivacyPolicyCondition concept definition"
endNonFunctionalProperties
hasPrivacyPolicy ofType (1 1) PrivacyPolicy
hasPrivacyCondition ofType (1 1) //Condition

concept PrivacyPolicy
nonFunctionalProperties
dc#description hasValue "PrivacyPolicy concept definition"
endNonFunctionalProperties

concept AccessPrivacyPolicy subConceptOf PrivacyPolicy
nonFunctionalProperties
dc#description hasValue "AccessPrivacyPolicy concept definition"
endNonFunctionalProperties

concept CollectionPrivacyPolicy subConceptOf PrivacyPolicy
nonFunctionalProperties
dc#description hasValue "CollectionPrivacyPolicy concept definition"
endNonFunctionalProperties

concept StoragePrivacyPolicy subConceptOf PrivacyPolicy
nonFunctionalProperties
dc#description hasValue "StoragePrivacyPolicy concept definition"
endNonFunctionalProperties

concept AlterationPrivacyPolicy subConceptOf PrivacyPolicy
nonFunctionalProperties
dc#description hasValue "AlterationPrivacyPolicy concept definition"
endNonFunctionalProperties

concept RightOfDisclosure subConceptOf Right
nonFunctionalProperties
dc#description hasValue "RightOfDisclosure concept definition"
endNonFunctionalProperties
capturesInformationRelating ofType (1 *) DisclosedItem

concept TerminationType
nonFunctionalProperties
dc#description hasValue "TerminationType concept definition"
endNonFunctionalProperties

concept ProviderInitiatedTerminationType subConceptOf TerminationType
nonFunctionalProperties
dc#description hasValue "ProviderInitiatedTerminationType concept definition"
endNonFunctionalProperties

concept RequestorInitiatedTerminationType subConceptOf TerminationType
nonFunctionalProperties
dc#description hasValue "RequestorInitiatedTerminationType concept definition"
endNonFunctionalProperties

concept RightOfSuspension subConceptOf Right
nonFunctionalProperties
dc#description hasValue "RightOfSuspension concept definition"
endNonFunctionalProperties
hasSuspensionCondition ofType //Condition
hasResumptionCondition ofType //Condition
hasSuspensionObligation ofType (0 1) obl#Obligation
hasResumptionObligation ofType (0 1) obl#Obligation
hasMaximumNumberOfSuspensions ofType (0 1) loc#nonNegativeInteger
hasMinimumSuspensionPeriod ofType (0 1) temp#TemporalDuration
hasMaximumDurationSuspensionPeriod ofType (0 1) temp#TemporalDuration
hasMaximumAggregatedSuspensionPeriod ofType (0 1) temp#TemporalDuration
hasSuspensionProcedure ofType (0 1) Procedure
hasResumptionProcedure ofType (0 1) Procedure
hasSuspensionAvailability ofType (0 1) ava#Availability

concept RightOfCoolingOffPeriod subConceptOf Right
nonFunctionalProperties
dc#description hasValue "RightOfCoolingOffPeriod concept definition"
endNonFunctionalProperties
Listing A.10: Trust Ontology

wsmlVariant "http://www.wsmo.org/wsml/wsml−syntax/wsml−rule"

namespace {
  dc: "http://purl.org/dc/elements/1.1#",
  xsd: "http://www.w3.org/2001/XMLSchema#",
  wsml: "http://www.wsmo.org/wsml−syntax#",
  temp: "http://www.wsmo.org/ontologies/nfp/temporalNFPOntology#",
  qua: "http://www.wsmo.org/ontologies/nfp/qualityNFPOntology#"
}

ontology "http://www.wsmo.org/ontologies/nfp/trustNFPOntology"

  nonFunctionalProperties
dc#title hasValue "Trust Ontology"
dc#creator hasValue {"Ioan Toma"}
dc#subject hasValue {"Trust", "Endorsement"}
dc#description hasValue "Trust Ontology"
dc#publisher hasValue "DERI Innsbruck"
dc#contributor hasValue "Ioan Toma"
dc#date hasValue "2006−05−08"
dc#type hasValue "http://www.wsmo.org/2004/d2#ontologies"
dc#format hasValue "text/html"
dc#identifier hasValue "http://www.wsmo.org/ontologies/nfp/trustNFPOntology"
dc#language hasValue "en−US"
wsml#version hasValue "Revision : 1.3"
endNonFunctionalProperties

concept Endorsement
  nonFunctionalProperties
dc#description hasValue "Endorsement concept definition"
endNonFunctionalProperties

concept InternallyManagedEndorsement
  nonFunctionalProperties
dc#description hasValue "InternallyManagedEndorsement concept definition"
endNonFunctionalProperties
hasPartyThatProvidesEndorsement ofType (1 1) ServiceParty
hasEndorsementReceived ofType (1 1) temp#TimePoint
hasRating ofType (1 1) qua#Rating
hasComment ofType (1 1) string

calendar:ExternallyManagedEndorsement

nonFunctionalProperties
dc#description hasValue "ExternallyManagedEndorsement concept definition"
endNonFunctionalProperties
isManagedExternallyAt ofType (1 1) iri

Listing A.11: Quality of Service Ontology

```xml
wsmlVariant _"http://www.wsmo.org/wsml/wsml—syntax/wsml—rule"

namespace {
    _"http://www.wsmo.org/ontologies/nfp/qualityNFPOntology#",
    dc _"http://purl.org/dc/elements/1.1#",
    xsd _"http://www.w3.org/2001/XMLSchema#",
    wsml _"http://www.wsmo.org/wsml/wsml—syntax#",
    loc _"http://www.wsmo.org/ontologies/nfp/locativeNFPOntology#",
    temp _"http://www.wsmo.org/ontologies/nfp/temporalNFPOntology#",
    price _"http://www.wsmo.org/ontologies/nfp/pricingNFPOntology#"
}

ontology _"http://www.wsmo.org/ontologies/nfp/qualityNFPOntology#"

nonFunctionalProperties
dc#title hasValue "Quality of Service Ontology"
dc#creator hasValue {"Ioan Tomă"}
dc#subject hasValue {"Non-functional properties"}
dc#description hasValue "Quality of Service Ontology"
dc#publisher hasValue "DERI Innsbruck"
dc#contributor hasValue {"Ioan Tomă"}
dc#date hasValue "2006−05−06"
dc#type hasValue _"http://www.wsmo.org/2004/d2#ontologies"
dc#format hasValue "text/html"
dc#identifier hasValue _"http://www.wsmo.org/ontologies/nfp/qualityNFPOntology#"
dc#language hasValue "en-US"
wsml#version hasValue "Revision 1.3"
endNonFunctionalProperties

concept Standard

nonFunctionalProperties
dc#description hasValue "Standard concept definition"
endNonFunctionalProperties

hasPublisher ofType Publisher
supercedes ofType Standard
hasCoverage ofType loc#Region
isAvailableAt ofType loc#LocativeEntity
hasSupportedConformanceLevel ofType StandardLevel
hasStatus ofType Status
hasPublicationDate ofType (1 1) temp#TemporalDate
hasTitle ofType (1 1) string
hasAuthor ofType (1 *) Author
hasVersionNumber ofType (1 1) decimal

concept Rating

nonFunctionalProperties
dc#description hasValue "Rating concept definition"
endNonFunctionalProperties

hasName ofType (1 1) string
hasDescription ofType (1 1) string
hasRatingOf ofType Rated
ratingValue ofType RatingValue

concept Rated

nonFunctionalProperties
dc#description hasValue "Rated concept definition"
endNonFunctionalProperties

concept Ranking

nonFunctionalProperties
dc#description hasValue "Ranking concept definition"
endNonFunctionalProperties

isValidForPeriod ofType (0 1) temp#TemporalDuration
hasRankingPosition ofType (1 1) integer

concept RatingValue

nonFunctionalProperties
dc#description hasValue "RatingValue concept definition"
```
Listing A.12: Security Ontology

```
wsmlVariant "http://www.wsmo.org/wsml/wsml-syntax/wsml-rule"

namespace {
    _http://www.wsmo.org/ontologies/nfp/securityNFPOntology#
    dc _http://purl.org/dc/elements/1.1#"
    xsd _http://www.w3.org/2001/XMLSchema#"
    wsml _http://www.wsmo.org/wsml/wsml-syntax#"
    temp _http://www.wsmo.org/ontologies/nfp/temporalNFPOntology#
    qua _http://www.wsmo.org/ontologies/nfp/qualityNFPOntology#
    loc _http://www.wsmo.org/ontologies/nfp/locativeNFPOntology#
}

ontology _http://www.wsmo.org/ontologies/nfp/securityNFPOntology"
nonFunctionalProperties
dc#title hasValue “Security Ontology”
dc#creator hasValue {Ioan Toma}
dc#subject hasValue {“Security”, “Confidentiality”}
dc#description hasValue “Security Ontology”
dc#publisher hasValue “DERI Innsbruck”
dc#contributor hasValue {Ioan Toma}
dc#date hasValue “2006–05–06”
dc#type hasValue _http://www.wsmo.org/2004/d2#ontologies”
dc#format hasValue “text/html”
dc#identifier hasValue _http://www.wsmo.org/ontologies/nfp/securityNFPOntology"
dc#language hasValue “en—US”
wsml#version hasValue “Revision : 1.3”
endNonFunctionalProperties

concept IdentificationRequirement
nonFunctionalProperties
dc#description hasValue “IdentificationRequirement concept definition”
endNonFunctionalProperties
requiresCollectiveIdentificationPoints ofType (0 1) loc#nonNegativeInteger
acceptsIdentification ofType (1 *) AcceptableIdentification

concept AcceptableIdentification
nonFunctionalProperties
dc#description hasValue “AcceptableIdentification concept definition”
endNonFunctionalProperties
hasIdentificationType ofType (1 1) IdentificationType
isMandatory ofType (1 1) boolean
```
Listing A.13: Intellectual Property Ontology

wsmlVariant _http://www.wsmo.org/wsml/wsml−syntax/wsml−rule"

namespace { _http://www.wsmo.org/ontologies/nfp/ipNFPOntology#,
  dc _http://purl.org/dc/elements/1.1#,
  xsd _http://www.w3.org/2001/XMLSchema#,
  wsml _http://www.wsmo.org/wsml/wsml−syntax#,
  loc _http://www.wsmo.org/ontologies/nfp/locativeNFPOntology#,
  temp _http://www.wsmo.org/ontologies/nfp/temporalNFPOntology#,
  obl _http://www.wsmo.org/ontologies/nfp/obligationsNFPOntology#,
  qua _http://www.wsmo.org/ontologies/nfp/qualityNFPOntology#,
  pro _http://www.wsmo.org/ontologies/nfp/providerNFPOntology#} 

ontology _http://www.wsmo.org/ontologies/nfp/ipNFPOntology/

nonFunctionalProperties
dc#title hasValue "Intellectual Property Ontology"
dc#creator hasValue {"Ioan Toma"}
dc#subject hasValue ["Trademark", "Patent"]
dc#description hasValue "Intellectual Property Ontology"
dc#publisher hasValue "DERI Innsbruck"
dc#contributor hasValue ["Ioan Toma"]
dc#date hasValue "2006−05−08"
dc#type hasValue _http://www.wsmo.org/2004/d2#ontologies"
dc#format hasValue "text/html"
dc#identifier hasValue _http://www.wsmo.org/ontologies/nfp/ipNFPOntology" dc#language hasValue "en−US"
wsml#version hasValue "Revision : 1.3"
endNonFunctionalProperties

concept RegisteredIPRight
nonFunctionalProperties
dc#description hasValue "RegisteredIPright concept definition"
endNonFunctionalProperties

hasIPRight ofType IPRight
hasApplicationNumber ofType (1 1) decimal
hasRightStatus ofType (1 1) IPRightStatus
hasRegionOJProtection ofType (0 1) loc#Region
hasCountryOfOrigin ofType (0 1) loc#Region
hasAgentAddress ofType (0 1) loc#Address
hasAgentName ofType (0 1) string
isDetailsFor ofType (0 1) loc#LocativeEntity
hasRegistrationDate ofType (0 1) temp#TemporalDate
hasLodgingDate ofType (1 1) temp#TemporalDate
hasOwner ofType (1 1) pro#Provider

category IPRight
category nonFunctionalProperties
dc#description hasValue "IPRight concept definition"
endNonFunctionalProperties
category Trademark
category nonFunctionalProperties
dc#description hasValue "Trademark concept definition"
endNonFunctionalProperties
hasWordmark ofType (1 1) iri
hasFigurativeMarkClassification ofType (1 1) FigurativeMarkClassificationVienna
hasTrademarkClassification ofType (1 1) TrademarkClassificationNice
// either Goods or Services

concept FigurativeMarkClassificationVienna
category nonFunctionalProperties
dc#description hasValue "FigurativeMarkClassificationVienna concept definition"
endNonFunctionalProperties

value

concept TrademarkClassificationNice
category nonFunctionalProperties
dc#description hasValue "TrademarkClassificationNice concept definition"
endNonFunctionalProperties

value

concept Patent
category nonFunctionalProperties
dc#description hasValue "Patent concept definition"
endNonFunctionalProperties
hasPatentClassification ofType (1 1) PatentClassificationIPCStrasbourg
hasTitle ofType (1 1) string
hasInventor ofType (1 1) string

concept PatentClassificationIPCStrasbourg
category nonFunctionalProperties
dc#description hasValue "PatentClassificationIPCStrasbourg concept definition"
endNonFunctionalProperties

value

concept Design
category nonFunctionalProperties
dc#description hasValue "Design concept definition"
endNonFunctionalProperties
hasDesignClassification ofType (1 1) IDCLocarno

concept IDCLocarno
category nonFunctionalProperties
dc#description hasValue "IDCLocarno concept definition"
endNonFunctionalProperties

value

concept RegisteredIPRight
category nonFunctionalProperties
dc#description hasValue "RegisteredIPRight concept definition"
endNonFunctionalProperties

Listing A.14: Rewards Ontology

wsmlVariant _http://www.wsmo.org/wsml/wsml−syntax/wsml−rule"

namespace {
    _http://www.wsmo.org/ontologies/nfp/rewardsNFPOntology#,
    dc _http://purl.org/dc/elements/1.1#,
    xsd _http://www.w3.org/2001/XMLSchema#,
    wsml _http://www.wsmo.org/wsml/wsml−syntax#,
    loc _http://www.wsmo.org/ontologies/nfp/locativeNFPOntology#,
    qua _http://www.wsmo.org/ontologies/nfp/qualityNFPOntology#,
    ava _http://www.wsmo.org/ontologies/nfp/availabilityNFPOntology#,
    obl _http://www.wsmo.org/ontologies/nfp/obligationsNFPOntology#}
D28.4 – NON-FUNCTIONAL PROPERTIES IN WEB SERVICES

Listing A.15: Provider Ontology

wsmlVariant _http://www.wsmo.org/wsml/wsml−syntax/wsml−rule"

namespace {
 _http://www.wsmo.org/ontologies/nfp/providerNFPOntology#,
 dc _http://purl.org/dc/elements/1.1#,
 xsd _http://www.w3.org/2001/XMLSchema#,
 wsml _http://www.wsmo.org/wsml/wsml−syntax#,
 loc _http://www.wsmo.org/ontologies/nfp/locativeNFPOntology#,
 obl _http://www.wsmo.org/ontologies/nfp/obligationsNFPOntology#,
 qua _http://www.wsmo.org/ontologies/nfp/qualityNFPOntology#,
}

ontology _http://www.wsmo.org/ontologies/nfp/providerNFPOntology"

endNonFunctionalProperties
concept Provider
  nonFunctionalProperties
  dc#description hasValue "Provider concept definition"
endNonFunctionalProperties
hasName ofType (1 1) string
hasPriceMatching ofType (1 1) PriceMatching
hasCompliance ofType AchievedCompliance
offersPriceMatching ofType (1 1) PriceMatching
hasProviderFeedback ofType (0 1) Endorsement
hasMissionStatement ofType (0 1) Statement
isLegallyBoundBy ofType (1 ∗) Legislation
hasYearOfInception ofType (0 1) year
hasProviderMembership ofType ProviderMembership
hasAssociationWith ofType AssociationTypeProvider

concept AssociationTypeProvider
  nonFunctionalProperties
  dc#description hasValue "AssociationTypeProvider concept definition"
endNonFunctionalProperties
hasAssociationType ofType (1 1) AssociationType
withProvider ofType (1 1) Provider

concept ProviderMembership
  nonFunctionalProperties
  dc#description hasValue "ProviderMembership concept definition"
endNonFunctionalProperties
providerOf ofType (1 1) Provider
hasMembership ofType (1 1) obl#Membership
wasAchievedOn ofType (1 1) tmp#TemporalDate
hasMembershipExpiryOf ofType (1 1) tmp#TemporalDate

concept AchievedCompliance
  nonFunctionalProperties
  dc#description hasValue "AchievedCompliance concept definition"
endNonFunctionalProperties
hasCompliance ofType (1 1) Compliance
hasConformanceRating ofType (1 1) StandardLevelName
wasAchievedOn ofType (1 1) tmp#TemporalDate
wasVerifiedBy ofType (1 1) Provider

concept Compliance
  nonFunctionalProperties
  dc#description hasValue "Compliance concept definition"
endNonFunctionalProperties

concept AssociationType
  nonFunctionalProperties
  dc#description hasValue "AssociationType concept definition"
endNonFunctionalProperties

concept PartnerType subConceptOf AssociationType
  nonFunctionalProperties
  dc#description hasValue "PartnerType concept definition"
endNonFunctionalProperties

concept SubsidiaryType subConceptOf AssociationType
  nonFunctionalProperties
  dc#description hasValue "SubsidiaryType concept definition"
endNonFunctionalProperties

concept OwnerType subConceptOf AssociationType
  nonFunctionalProperties
  dc#description hasValue "OwnerType concept definition"
endNonFunctionalProperties

concept SupplierToType subConceptOf AssociationType
  nonFunctionalProperties
  dc#description hasValue "SupplierToType concept definition"
endNonFunctionalProperties

concept AgencyType subConceptOf AssociationType
  nonFunctionalProperties
  dc#description hasValue "AgencyType concept definition"
endNonFunctionalProperties

concept DivisionType subConceptOf AssociationType
  nonFunctionalProperties
  dc#description hasValue "DivisionType concept definition"
endNonFunctionalProperties
Listing A.16: Measures Ontology

```xml
wsmlVariant _http://www.wsmo.org/wsml/wsml−syntax/wsml−rule"

namespace {
   _http://www.wsmo.org/ontologies/nfp/measuresNFPOntology#,
   dc _http://purl.org/dc/elements/1.1#,
   xsd _http://www.w3.org/2001/XMLSchema#,
   wsml _http://www.wsmo.org/wsml/wsml−syntax#
}

ontology _http://www.wsmo.org/ontologies/nfp/measuresNFPOntology"

nonFunctionalProperties
dc#title hasValue "Measures Ontology"
dc#creator hasValue {"Ioan Toma"}
dc#subject hasValue {"MeasurableQuantity", "Angle"}
dc#description hasValue "Measures Ontology"
dc#publisher hasValue "DERI Innsbruck"
dc#contributor hasValue {"Doug Foxvog", "Ioan Toma"}
dc#date hasValue 2006−08−30"
dc#type hasValue _http://www.wsmo.org/2004/d2#ontologies"
dc#format hasValue "text/html"
dc#identifier hasValue _http://www.wsmo.org/ontologies/nfp/measuresNFPOntology"
dc#language hasValue "en−US"
wsml#version hasValue "Revision : 1.3"
endNonFunctionalProperties

custom

concept MeasurableQuantity
numUnits ofType _decimal
ofUnits ofType UnitOfMeasure

custom

concept Distance subConceptOf MeasurableQuantity
numUnits ofType _decimal
ofUnits ofType UnitOfDistance

custom

concept Angle subConceptOf MeasurableQuantity
numUnits ofType _decimal
ofUnits ofType UnitOfArc

custom

concept UnitOfMeasure

custom

concept UnitOfDistance subConceptOf UnitOfMeasure

custom

concept UnitOfMass subConceptOf UnitOfMeasure

custom

concept UnitOfTemperature subConceptOf UnitOfMeasure

custom

instance MeterUnit memberOf UnitOfDistance
instance DegreeOfArc memberOf UnitOfArc
instance CelsiusDegree memberOf UnitOfTemperature

relation unitMultiplicationFactor ( ofType UnitOfMeasure, ofType UnitOfMeasure, ofType _decimal)

custom

concept Granularity
nonFunctionalProperties
dc#description hasValue "Granularity concept definition"
endNonFunctionalProperties

value ofType _string

custom

concept GranularityAndUnitOfMeasure
nonFunctionalProperties
dc#description hasValue "GranularityAndUnitOfMeasure concept definition"
endNonFunctionalProperties

hashCode ofType UnitOfMeasure
hasGranularity ofType GranularityAndUnitOfMeasure

concept ItemGranularity
nonFunctionalProperties
dc#description hasValue "ItemGranularity concept definition"
endNonFunctionalProperties

hashCode ofType ItemGranularity

relation relatesToItem ofType Item
```
Listing A.17: Currency Ontology

```xml
wsmlVariant _http://www.wsmo.org/wsml/wsml−syntax/wsml−rule"

namespace {
  _http://www.wsmo.org/ontologies/nfp/currencyNFPOntology#,
  dc _http://purl.org/dc/elements/1.1#,
  xsd _http://www.w3.org/2001/XMLSchema#,
  wsml _http://www.wsmo.org/wsml/wsml−syntax#}

ontology _http://www.wsmo.org/ontologies/nfp/currencyNFPOntology"
  nonFunctionalProperties
    dc#title hasValue "Currency Ontology"
    dc#creator hasValue {"Ioan Toma"}
    dc#subject hasValue "Currency"
    dc#description hasValue "Price Ontology"
    dc#publisher hasValue "DERI Innsbruck"
    dc#contributor hasValue {"Doug Foxvog", "Ioan Toma"}
    dc#date hasValue "2006−06−19"
    dc#type hasValue _http://www.wsmo.org/2004/d2#ontologies"
    dc#format hasValue "text/html"
    dc#identifier hasValue _http://www.wsmo.org/ontologies/nfp/currencyNFPOntology"
    dc#language hasValue "en−US"
    wsml#version hasValue "Revision : 1.3"

endNonFunctionalProperties

concept Currency
  nonFunctionalProperties
    dc#description hasValue "Currency concept definition"
    dc#relation hasValue validCurrency

endNonFunctionalProperties

concept Euro subConceptOf Currency
  nonFunctionalProperties
    dc#description hasValue "Euro concept definition"

endNonFunctionalProperties

concept USD subConceptOf Currency
  nonFunctionalProperties
    dc#description hasValue "USD concept definition"

endNonFunctionalProperties

concept GBP subConceptOf Currency
  nonFunctionalProperties
    dc#description hasValue "GBP concept definition"

endNonFunctionalProperties

concept AUD subConceptOf Currency
  nonFunctionalProperties
    dc#description hasValue "AUD concept definition"

endNonFunctionalProperties
```

B Appendix B

In the following appendix we show how the same non-functional property specification is attached to a Web service description according to the approaches proposed in Section 4.3, Section 4.5 and Section 4.6.

Listing B.1: Example of NFPs descriptions based on the approach proposed in Section 4.3

```xml
namespace { ...
    po "http://www.wsmo.org/ontologies/nfp/priceNFPOntology#",
    hu "http://www.example.org/ontologies/humansOntology#",
    cl "http://www.example.org/ontologies/clientsOntology#",
    cur "http://www.wsmo.org/ontologies/nfp/currencyNFPOntology#"
}

webService "http://example.org/ws"
    annotations
        dc#title hasValue "WSML example service with non-functional properties"
    endAnnotations

    nonFunctionalProperty
        price hasValue ?price
        annotations
            dc#description hasValue "If the client is older than 60 or younger than 10 years old the invocation price is lower than 10 euro"
        endAnnotations
    endNonFunctionalProperty

    importsOntology { ...
    }

    definedBy
        ?client[age hasValue ?age] memberOf hu#human and ?age[amount hasValue ?years, units hasValue hu#YearsDuration] memberOf hu#age and (greaterEqual(?years, 60) or lessEqual(?years, 10)) implies ?price[hasAmount hasValue ?amount, hasCurrency hasValue cur#Euro] memberOf po#AbsolutePrice and lessEqual(?amount, 10).
    endDefinedBy

    capability wsCapability
        interface wsInterface
```

Listing B.2: Example revised according to the approach proposed in Section 4.5

```xml
// WS description referring to the Policy according namespace { ...
    pol "http://www.siti.disco.unimib.it/research/ontologies/Policies#"
}

webService "http://example.org/ws"
    annotations
        dc#title hasValue "WSML example service with non-functional properties"
    endAnnotations

    nonFunctionalProperty
        annotations
            dc#description hasValue "If the client is older than 60 or younger than 10 years old the invocation price is lower than 10 euro"
        endAnnotations
    endNonFunctionalProperty

    importsOntology { ...
    }

    policy hasValue pol#policyGold

    capability wsCapability
        interface wsInterface

// Policy ontology recalled by the WS description
wsmlVariant "http://www.wsmo.org/wsml/wsml-syntax/wsml-flight"
namespace { ...
    ontoNFP "http://www.siti.disco.unimib.it/research/ontologies/OntoNFPwsml#",
    po "http://www.wsmo.org/ontologies/nfp/priceNFPOntology#",
}
Listing B.3: Example revised according to the approach proposed in Section 4.6

```xml
namespace { _http://example.org/ws#,
    po "http://www.wsmo.org/ontologies/nfp/priceNFPOntology#",
    hu _http://www.example.org/ontologies/humansOntology#,
    cl _http://www.example.org/ontologies/clientsOntology#,
    cur _http://www.wsmo.org/ontologies/nfp/currencyNFPOntology#,
    dc _http://purl.org/dc/elements/1.1" }

ontology Policies

importsOntology {
    _http://www.siti.disco.unimib.it/research/ontologies/OntoNFPwsml#,
    _http://www.wsmo.org/ontologies/nfp/priceNFPOntology#,
    _http://www.example.org/ontologies/humansOntology#,
    _http://www.example.org/ontologies/clientsOntology#,
    _http://www.wsmo.org/ontologies/nfp/currencyNFPOntology#,
}

instance policyGold memberOf ontoNFP#Policy
annotations
dc#description hasValue "if the client is older than 60 or younger than 10 years old the invocation price is lower than 10 euro"
endAnnotations
ontoNFP#referenceFunctionalDescription hasValue _http://example.org/ws"
ontoNFP#hasConditionDescription hasValue "This policy is applicable for client older than 60 or younger than 10 years old"
ontoNFP#hasCondition hasValue policyGoldCondition
ontoNFP#hasNfp hasValue fastP_price

axiom goldPolicyCondition definedBy

ontoNFP#satisfiedFor(goldPolicyCondition,?client) :
    − ?client[age hasValue ?age] memberOf [{ontoNFP#Client,hu#human} and ?age[amount hasValue ?years, units hasValue hu#YearsDuration] memberOf hu#age and (greaterEqual(?years, 60) or lessEqual(?years,10)).

instance goldP_price memberOf po#AbsoultePrice
ontoNFP#hasExpression hasValue goldP_price

instance goldP_price constExpression memberOf po#PriceExpression
ontoNFP#hasOperator hasValue ontoNFP#lessEqual
ontoNFP#hasParameter hasValue 10.0
ontoNFP#hasUnit hasValue cur#Euro
```

Listing B.3: Example revised according to the approach proposed in Section 4.6.
constraintOperator hasValue lessEqual
parameter hasValue 10.0
unit hasValue cur#Euro
capability wsCapability
interface wsInterface
This appendix reports the non-functional property meta-model described in Section 4.4.1 and a domain ontology that uses the meta-model to define non-functional properties.

Listing C.1: OntoNFPwsml

```xml
wsmlVariant "http://www.wsmo.org/wsml/syntax/wsml-flight"
namespace {  
  "http://www.siti.disco.unimib.it/research/ontologies/OntoNFPwsml#",  
  dc "http://purl.org/dc/elements/1.1#"
}

ontology OntoNFPwsml
  importsOntology  
  "http://purl.org/dc/elements/1.1#"

concept PolicyNfp
  hasExpression ofType (1 1) PolicyExpression

concept QualitativeNfp subConceptOf PolicyNfp
  hasExpression ofType (1 1) QualitativeExpression

concept ListNfp subConceptOf QualitativeNfp
  hasExpression ofType (1 1) ListExpression

concept UserDefinedNfp subConceptOf QualitativeNfp
  hasExpression ofType (1 1) UserDefinedExpression

concept QuantitativeNfp subConceptOf PolicyNfp
  hasExpression ofType (1 1) QuantitativeExpression

concept SingleValueNfp subConceptOf QuantitativeNfp
  hasExpression ofType (1 1) SingleValueExpression

concept RangeNfp subConceptOf QuantitativeNfp
  hasExpression ofType (1 1) RangeExpression

concept PolicyExpression
  hasOperator ofType (1 1) ConstraintOperator

concept QualitativeExpression subConceptOf PolicyExpression
  hasOperator ofType (1 1) QualitativeOperator
  hasParameters ofType (1 *) anyConcept

concept ListExpression subConceptOf QualitativeExpression
  hasOperator ofType (1 1) ListOperator
  hasParameters ofType (1 *) anyConcept

concept UserDefinedExpression subConceptOf QualitativeExpression
  hasOperator ofType (1 1) UserDefinedOperator
  hasParameters ofType (1 *) anyConcept

concept QuantitativeExpression subConceptOf PolicyExpression
  hasOperator ofType (1 1) QuantitativeOperator
  hasUnit ofType (1 1) Unit

concept SingleValueExpression subConceptOf QuantitativeExpression
  hasOperator ofType (1 1) BinaryOperator
  hasParameter ofType (1 1) float

concept RangeExpression subConceptOf QuantitativeExpression
  hasOperator ofType (1 1) TernaryOperator
  hasMinParameter ofType (1 1) float
  hasMaxParameter ofType (1 1) float

concept ConstraintOperator

concept QuantitativeOperator subConceptOf ConstraintOperator

concept QualitativeOperator subConceptOf ConstraintOperator

concept ListOperator subConceptOf QualitativeOperator

concept UserDefinedOperator subConceptOf QualitativeOperator
```

concept BinaryOperator subConceptOf QuantitativeOperator

concept TernaryOperator subConceptOf QuantitativeOperator

concept Policy
  nonFunctionalProperties
    dc#label hasValue "NFP Policy"
    dc#description hasValue "A policy collecting a number of NFP specifications"
  endNonFunctionalProperties
  hasReferenceFunctionalDescription ofType (1 1) _iri
  hasConditionDescription ofType (0 1) _string
  hasCondition ofType PolicyCondition
  hasNfp impliesType (1 \(*\) PolicyNfp

concept RequestedPolicy subConceptOf Policy
  hasNfp impliesType Request

concept Request subConceptOf PolicyNfp
  hasRelevance ofType (1 1) double

concept ListRequest subConceptOf ( ListNfp, Request )

concept UserDefinedRequest subConceptOf ( UserDefinedNfp, Request )

concept SingleValueRequest subConceptOf ( SingleValueNfp, Request )

concept RangeRequest subConceptOf ( RangeNfp, Request )

concept Unit

concept PolicyCondition

concept Client

  relation satisfiedFor ( ofType PolicyCondition, ofType Client)

  relation satisfied ( ofType PolicyCondition)

instance greaterEqual memberOf BinaryOperator
  nonFunctionalProperties
data#description hasValue "the required value is a lower bound and the higher offered value is better"
  endNonFunctionalProperties

instance lessEqual memberOf BinaryOperator
  nonFunctionalProperties
data#description hasValue "the required value is an upper bound and the lower offered value is better"
  endNonFunctionalProperties

instance atLeast memberOf BinaryOperator
  nonFunctionalProperties
data#description hasValue "the required value is a lower bound and the lower offered value is better"
  endNonFunctionalProperties

instance atMost memberOf BinaryOperator
  nonFunctionalProperties
data#description hasValue "the required value is an upper bound and the higher offered value is better"
  endNonFunctionalProperties

instance interval memberOf TernaryOperator
  nonFunctionalProperties
data#description hasValue "the required value is specified in a range"
  endNonFunctionalProperties

instance equal memberOf BinaryOperator

instance exact memberOf ListOperator

instance exist memberOf ListOperator

Listing C.2: DomainOnto
wsmlVariant _http://www.wsmo.org/wsml/wsml−syntax/wsml−flight"
namespace {
    _http://www.siti.disco.unimib.it/research/ontologies/DomainOnto#,
    ontoNFP _http://www.siti.disco.unimib.it/research/ontologies/OntoNFPwsml#,
    go _http://www.wsmo.org/ontologies/nfp/locativeNFPOntology#,
    co _http://www.wsmo.org/ontologies/nfp/currencyNFPOntology#
}

ontology Domain_Onto
importsOntology {
    _http://www.siti.disco.unimib.it/research/ontologies/OntoNFPwsml#,
    _http://www.wsmo.org/ontologies/nfp/locativeNFPOntology#,
    _http://www.wsmo.org/ontologies/nfp/currencyNFPOntology#}

case CoveredLocation subConceptOf ontoNFP#ListNfp
ontoNFP#hasExpression impliesType (1 1) LocationExpression

case CoveredLocationRequest subConceptOf ontoNFP#ListRequest
ontoNFP#hasExpression impliesType (1 1) LocationExpression

case LocationExpression subConceptOf ontoNFP#ListExpression
ontoNFP#hasParameters impliesType (1 +) go#GeographicalRegion

case Cost subConceptOf ontoNFP#SingleValueNfp
ontoNFP#hasExpression impliesType (1 1) CostExpression

case CostRequest subConceptOf ontoNFP#SingleValueRequest
ontoNFP#hasExpression impliesType (1 1) CostExpression

case CostExpression subConceptOf ontoNFP#SingleValueExpression
ontoNFP#hasUnit impliesType (1 1) co#Currency

case MapResolution subConceptOf ontoNFP#RangeNfp
ontoNFP#hasExpression impliesType (1 1) MapResolutionExpression

case MapResolutionRequest subConceptOf ontoNFP#RangeRequest
ontoNFP#hasExpression impliesType (1 1) MapResolutionExpression

case MapResolutionExpression subConceptOf ontoNFP#RangeExpression
ontoNFP#hasUnit impliesType (1 1) ScaleOneTo

case ScaleOneTo subConceptOf ontoNFP#Unit

case User
    hasTypology impliesType UserTypology

case UserTypology

instance Public memberOf UserTypology

instance Private memberOf UserTypology
D Appendix D

This appendix proposes examples of Goal and Web service descriptions related to shipment services provided by logistic operators. Moreover, this appendix shows Policies and Requested Policies associated to Goals and Web services created using the meta-model described in Section 4.4.1.

Policies and Requested Policies contain the specification of NFPs that characterize a shipment service: (i) the Payment method that can be carriage paid or carriage forward; (ii) the Payment Deadline that represents a temporal limit to make the payment; (iii) the Insurance on the shipment (e.g., refund for loss, refund for damage); (iv) the Base Price that represents the cost for the service invocation; (v) the Hours To Delivery that specifies the minimum and maximum hours that the service required to perform the shipment.

In order to provide a realistic shipment discovery scenario, we propose the description of one Goal (ordinaryTransport) with its RequestedPolicy (LORequest1) and three Web Services (WSNorthItalyOrdinaryTransport, WSSouthItalyOrdinaryTransport, WSOmniTransport) with their associated Policies (fastPolicy, cheapPolicy, policyForRepetitiveTravel, goldPolicy, silverPolicy).

In the following a description of these elements and their WSML specifications is provided:

- **ordinaryTransport**: a goal for a shipment service that requires the set of NFPs specified in LORequest1.

- **LORequest1**: a RequestedPolicy specifying the following required NFPs: (i) the payment method is required to be carriage paid; (ii) the payment deadline is optionally preferred to be at least 15 days; (iii) the insurance is preferred to be of type refundForDamage and refundForLoss; (iv) the base price is required to be less or equal than 50 Euros; (v) the hours to delivery are preferred to be less or equal than 72.

- **WSNorthItalyOrdinaryTransport**: a shipment service that offers two different combined offers of NFPs specified in fastPolicy and cheapPolicy.

- **WSSouthItalyOrdinaryTransport**: a shipment service that offers two different combined offers of NFPs specified in goldPolicy and silverPolicy.

- **WSOmniTransport**: a shipment service that offers the combined offers of NFPs specified in policyForRepetitiveTravel.

- **fastPolicy**: a Policy applicable for users that offers the following NFPs: (i) the payment method is carriage paid; (ii) the payment deadline is equal to 15 days; (iii) the insurance is refundForLoss; (iv) the base price is equal to 40 Euros; (v) the hours to delivery are minimum 24 and maximum 48.

- **cheapPolicy**: a Policy applicable for users that offers the following NFPs: (i) the payment method is carriage paid; (ii) the payment deadline is equal to 30 days; (iii) the insurance is refundForLoss; (iv) the base price is equal to 25 Euros; (v) the hours to delivery are minimum 48 and maximum 72.

- **policyForRepetitiveTravel**: a Policy applicable for users that requires multiple shipment services that offers the following NFPs: (i) the payment method is carriage forward; (ii) the payment deadline is equal to 15 days; (iii) the insurance is refundForLoss and refundForDamage; (iv) the base price is equal to 20 Euros; (v) the hours to delivery are minimum 24 and maximum 48.
• **goldPolicy**: a Policy applicable for gold users that offers the following NFPs:
  (i) the payment method is carriage paid; (ii) the payment deadline is equal to 15 days; (iii) the insurance is refundForLoss and refundForDamage; (iv) the base price is equal to 45 Euros; (v) the hours to delivery are minimum 24 and maximum 48.

• **silverPolicy**: a Policy applicable for silver users that offers the following NFPs:
  (i) the payment method is carriage forward; (ii) the payment deadline is equal to 45 days; (iii) the insurance is refundForDamage; (iv) the base price is equal to 45 Euros; (v) the hours to delivery are minimum 72 and maximum 96.

Listing D.1: ordinary transport goal description

```xml
<wsmlVariant:_http:="http://www.wsmo.org/wsml/wsml-syntax/wsml-flight"
 namespace:_http:="http://www.siti.disco.unimib.it/research/LOGoal#",
dc:_http:="http://purl.org/dc/elements/1.1#",
req1:_http:="http://www.siti.disco.unimib.it/research/ontologies/LORequest1#"
>
goal ordinaryTransport
importsOntology (_http:="http://www.siti.disco.unimib.it/research/ontologies/RequestInstanceLOOntology#
requestInstanceLOOntology")
nonFunctionalProperty
annotations
dc#creator hasValue ("Marco Comerio")
endAnnotations
policy hasValue (req1#LOrequest1)
capability wsCapability
interface wsInterface
```

Listing D.2: Requested Policy for ordinary transport goal

```xml
<wsmlVariant:_http:="http://www.wsmo.org/wsml/wsml-syntax/wsml-flight"
 namespace:
 _http:="http://www.siti.disco.unimib.it/research/ontologies/LORequest#",
ontoNFP:_http:="http://www.siti.disco.unimib.it/research/ontologies/OntoNFPwsml#",
dc:_http:="http://purl.org/dc/elements/1.1#",
lo:_http:="http://www.siti.disco.unimib.it/research/ontologies/LOOntology#",
 nfpo:_http:="http://www.siti.disco.unimib.it/research/ontologies/NFPOntology#"
>
ontology requestInstanceLOOntology
importsOntology (_http:="http://www.siti.disco.unimib.it/research/ontologies/OntoNFPwsml#OntoNFPwsml",
 _http:="http://www.siti.disco.unimib.it/research/ontologies/LOOntology#LOOntology",
 _http:="http://www.siti.disco.unimib.it/research/ontologies/NFPOntology#NFPOntology"
)
instance LORequest1
memberOf ontoNFP#Request
annotations
dc#description hasValue "Request Instance for Logistic Operator"
endAnnotations
ontoNFP#referenceFunctionalDescription hasValue _http:="http://www.siti.disco.unimib.it/research/LOGoal#ordinaryTransport"
ontoNFP#hasNfp hasValue requestedPaymentMethod
ontoNFP#hasNfp hasValue requestedPaymentDeadline
ontoNFP#hasNfp hasValue requestedInsurance
ontoNFP#hasNfp hasValue requestedBasePrice
```
ontoNFP#hasNfp hasValue requestedHoursToDelivery

instance requestedPaymentMethod memberOf nfpo#ListPaymentMethodRequest
  hasExpression hasValue requestedPaymentMethodExpression
  hasRelevance hasValue 0.8

instance requestedPaymentMethodExpression memberOf nfpo#ListPaymentMethodExpression
  hasOperator hasValue ontoNFP#exact
  hasParameters hasValue nfpo#carriagePaid

instance requestedPaymentDeadline memberOf nfpo#SingleValuePaymentDeadlineRequest
  hasExpression hasValue requestedPaymentDeadlineExpression
  hasRelevance hasValue 0.4

instance requestedPaymentDeadlineExpression memberOf nfpo#SingleValuePaymentDeadlineExpression
  hasOperator hasValue ontoNFP#atLeast
  hasParameter hasValue 15
  hasUnit hasValue nfpo#days

instance requestedInsurance memberOf nfpo#ListInsuranceRequest
  hasExpression hasValue requestedInsuranceExpression
  hasRelevance hasValue 0.6

instance requestedInsuranceExpression memberOf nfpo#ListInsuranceExpression
  hasOperator hasValue ontoNFP#exact
  hasParameters hasValue {nfpo#refundForDamage, nfpo#refundForLoss}

instance requestedBasePrice memberOf nfpo#SingleValueBasePriceRequest
  hasExpression hasValue requestedBasePriceExpression
  hasRelevance hasValue 0.8

instance requestedBasePriceExpression memberOf nfpo#SingleValueBasePriceExpression
  hasOperator hasValue ontoNFP#lessEqual
  hasParameter hasValue 50
  hasUnit hasValue nfpo#euro

instance requestedHoursToDelivery memberOf nfpo#SingleValueHoursToDeliveryRequest
  hasExpression hasValue requestedHoursToDeliveryExpression
  hasRelevance hasValue 0.6

instance requestedHoursToDeliveryExpression memberOf nfpo#SingleValueHoursToDeliveryExpression
  hasParameter hasValue 72
  hasUnit hasValue nfpo#hour

Listing D.3: WSNorthItalyOrdinaryTransport description

wsmlVariant _"http://www.wsmo.org/wsml/wsml−syntax/wsml−flight"
namespace { _"http://www.siti.disco.unimib.it/research/LOWS#",
  dc :"http://purl.org/dc/elements/1.1",
  pol1 :"http://www.siti.disco.unimib.it/research/ontologies/LOPolicy1#",
  pol2 :"http://www.siti.disco.unimib.it/research/ontologies/LOPolicy2#"}

webService WSNorthItalyOrdinaryTransport
  annotations
dc#creator hasValue {"Marco Comerio"}
endAnnotations
  nonFunctionalProperty
  annotations
dc#description hasValue {"The WSNorthItalyOrdinaryTransport offers two different policies: fastPolicy and cheapPolicy"}
endAnnotations
  importsOntology { _"http://www.siti.disco.unimib.it/research/ontologies/LOPolicy1#PolicyInstanceLOOntology1",
    _"http://www.siti.disco.unimib.it/research/ontologies/LOPolicy2#PolicyInstanceLOOntology2"}
  policy hasValue {pol1#fastPolicy, pol2#cheapPolicy}
Listing D.4: WSSouthItalyOrdinaryTransport description

```xml
wsmlVariants = "http://www.wsmo.org/wsml/wsml-syntax/wsml-flight"
namespace {
  _http://www.siti.disco.unimib.it/research/LOWS#,
  dc _http://purl.org/dc/elements/1.1#,
  pol4 _http://www.siti.disco.unimib.it/research/ontologies/LOPolicy4#,
  pol5 _http://www.siti.disco.unimib.it/research/ontologies/LOPolicy9# }

webService WSSouthItalyOrdinaryTransport
  annotations
    dc#creator hasValue { "Marco Comerio" }
  endAnnotations
  nonFunctionalProperty
    annotations
      dc#description hasValue { "The WSSouthItalyOrdinaryTransport offers two different policies: goldPolicy and silverPolicy" }
    endAnnotations
    importsOntology {
      _http://www.siti.disco.unimib.it/research/ontologies/LOPolicy4#
        PolicyInstanceLOOntology4",
      _http://www.siti.disco.unimib.it/research/ontologies/LOPolicy5#
        PolicyInstanceLOOntology5"
    }
  policy hasValue { pol1#goldPolicy, pol2#silverPolicy }
  capability wsCapability
  interface wsInterface
```

Listing D.5: WSOmniTransport description

```xml
wsmlVariants = "http://www.wsmo.org/wsml/wsml-syntax/wsml-flight"
namespace {
  _http://www.siti.disco.unimib.it/research/LOWS#,
  dc _http://purl.org/dc/elements/1.1#,
  pol3 _http://www.siti.disco.unimib.it/research/ontologies/LOPolicy3# }

webService WSOmniTransport
  annotations
    dc#creator hasValue { "Marco Comerio" }
  endAnnotations
  nonFunctionalProperty
    annotations
      dc#description hasValue { "The WSOmniTransport offers the NFP specified in multiTravelPolicy" }
    endAnnotations
    importsOntology {
      _http://www.siti.disco.unimib.it/research/ontologies/LOPolicy3#
        PolicyInstanceLOOntology3"
    }
  policy hasValue { pol3#multiTravelPolicy }
  capability wsCapability
  interface wsInterface
```

Listing D.6: Fast Policy description

```xml
wsmlVariants = "http://www.wsmo.org/wsml/wsml-syntax/wsml-flight"
namespace {
  _http://www.siti.disco.unimib.it/research/ontologies/LOPolicy#,
  ontoNFP _http://www.siti.disco.unimib.it/research/ontologies/OntoNFPwsml#,
  dc _http://purl.org/dc/elements/1.1#,
  lo _http://www.siti.disco.unimib.it/research/ontologies/LOOntology#,
  nfp _http://www.siti.disco.unimib.it/research/ontologies/NFPOntology# }

ontology policyInstanceLOOntology1
```
Listing D.7: Cheap Policy description

```xml
importsOntology ["http://www.siti.disco.unimib.it/research/ontologies/OnfOPwsml#OntoNFPwsml",
"http://www.siti.disco.unimib.it/research/ontologies/LOOntology#LOOntology",
"http://www.siti.disco.unimib.it/research/ontologies/NFPOntology#NFPOntology"
]

instance fastPolicy memberOf ontoNFP#Policy
  annotations
dc#description hasValue "Policy Instance for Logistic Operator"
endAnnotations
ontoNFP#referenceFunctionalDescription hasValue "http://www.siti.disco.unimib.it/research/LOWS#
WSNorthItalyOrdinaryTransport"
ontoNFP#hasConditionDescription hasValue "This policy is applicable for subscribed client"
ontoNFP#hasCondition hasValue policyCondition
ontoNFP#hasNfp hasValue offeredPaymentMethod
ontoNFP#hasNfp hasValue offeredPaymentDeadline
ontoNFP#hasNfp hasValue offeredInsurance
ontoNFP#hasNfp hasValue offeredBasePrice
ontoNFP#hasNfp hasValue offeredHoursToDelivery
axiom policyCondition
  definedBy
    applicable(fastPolicy) :- #userTypology hasValue subscribed] memberOf lo#LogisticOperatorClient.

Instance offeredPaymentMethod memberOf ntpo#ListPaymentMethod
  hasExpression hasValue offeredPaymentMethodExpression

Instance offeredPaymentMethodExpression memberOf ntpo#ListPaymentMethodExpression
  hasOperator hasValue ontoNFP#exact
  hasParameters hasValue ntpo#carriagePaid

Instance offeredPaymentDeadline memberOf ntpo#SingleValuePaymentDeadline
  hasExpression hasValue offeredPaymentDeadlineExpression

Instance offeredPaymentDeadlineExpression memberOf ntpo#
  SingleValuePaymentDeadlineExpression
  hasOperator hasValue ontoNFP#equal
  hasParameter hasValue 15
  hasUnit hasValue ntpo#days

Instance offeredInsurance memberOf ntpo#ListInsurance
  hasExpression hasValue offeredInsuranceExpression

Instance offeredInsuranceExpression memberOf ntpo#ListInsuranceExpression
  hasOperator hasValue ontoNFP#exact
  hasParameters hasValue ntpo#refundForLoss

Instance offeredBasePrice memberOf ntpo#SingleValueBasePrice
  hasExpression hasValue offeredBasePriceExpression

Instance offeredBasePriceExpression memberOf ntpo#SingleValueBasePriceExpression
  hasOperator hasValue ontoNFP#equal
  hasParameter hasValue 40
  hasUnit hasValue ntpo#euro

Instance offeredHoursToDelivery memberOf ntpo#RangeHoursToDelivery
  hasExpression hasValue offeredHoursToDeliveryExpression

Instance offeredHoursToDeliveryExpression memberOf ntpo#RangeHoursToDeliveryExpression
  hasOperator hasValue ontoNFP#interval
  hasMinParameter hasValue 24
  hasMaxParameter hasValue 48
  hasUnit hasValue ntpo#hour
```

wsmlVariant _"http://www.wsmo.org/wsml/wsml−syntax/wsml−flight"
namespace {
  "http://www.siti.disco.unimib.it/research/ontologies/LOPolicy#";
}

ontology policyInstanceLOOntology2

instance cheapPolicy memberOf ontoNFP#Policy
  annotations
dc#description hasValue "Policy Instance for Logistic Operator"
endAnnotations

ontoNFP#referenceFunctionalDescription hasValue _"http://www.siti.disco.unimib.it/research/LOWS# WSNorthItalyOrdinaryTransport"

ontoNFP#hasConditionDescription hasValue "This policy is applicable for subscribed client"
ontoNFP#hasCondition hasValue policyCondition
ontoNFP#hasNfp hasValue offeredPaymentMethod
ontoNFP#hasNfp hasValue offeredPaymentDeadline
ontoNFP#hasNfp hasValue offeredInsurance
ontoNFP#hasNfp hasValue offeredBasePrice
ontoNFP#hasNfp hasValue offeredHoursToDelivery

axiom policyCondition definedBy
  applicable(cheapPolicy) :- _[userTypology hasValue subscribed] memberOf to# LogisticOperatorClient.

instance offeredPaymentMethod memberOf nfpo#ListPaymentMethod
  hasExpression hasValue offeredPaymentMethodExpression

instance offeredPaymentMethodExpression memberOf nfpo#ListPaymentMethodExpression
  hasOperator hasValue ontoNFP#exact
  hasParameters hasValue nfpo#carriagePaid

instance offeredPaymentDeadline memberOf nfpo#SingleValuePaymentDeadline
  hasExpression hasValue offeredPaymentDeadlineExpression

instance offeredPaymentDeadlineExpression memberOf nfpo#SingleValuePaymentDeadlineExpression
  hasOperator hasValue ontoNFP#equal
  hasParameter hasValue 30
  hasUnit hasValue nfpo#days

instance offeredInsurance memberOf nfpo#ListInsurance
  hasExpression hasValue offeredInsuranceExpression

instance offeredInsuranceExpression memberOf nfpo#ListInsuranceExpression
  hasOperator hasValue ontoNFP#exact
  hasParameters hasValue nfpo#refundForLoss

instance offeredBasePrice memberOf nfpo#SingleValueBasePrice
  hasExpression hasValue offeredBasePriceExpression

instance offeredBasePriceExpression memberOf nfpo#SingleValueBasePriceExpression
  hasOperator hasValue ontoNFP#equal
  hasParameter hasValue 25
  hasUnit hasValue nfpo#euro

instance offeredHoursToDelivery memberOf nfpo#RangeHoursToDelivery
  hasExpression hasValue offeredHoursToDeliveryExpression

instance offeredHoursToDeliveryExpression memberOf nfpo#RangeHoursToDeliveryExpression
  hasOperator hasValue ontoNFP#interval
  hasMinParameter hasValue 48
  hasMaxParameter hasValue 72
  hasUnit hasValue nfpo#hour
Listing D.8: Repetitive Travel Policy description

```xml
wsmlVariant "http://www.wsmo.org/wsml/wsml--syntax/wsml--flight"
namespace {
    "http://www.siti.disco.unimib.it/research/ontologies/LOPolicy#",
    ontoNFP "http://www.siti.disco.unimib.it/research/ontologies/OntoNFPwsml#",
    dc "http://purl.org/dc/elements/1.1#",
    lo "http://www.siti.disco.unimib.it/research/ontologies/LOOntology#",
    nfpo "http://www.siti.disco.unimib.it/research/ontologies/NFPOntology#"
}

ontology policyInstanceLOOntology3
    importsOntology {
        "http://www.siti.disco.unimib.it/research/ontologies/OntoNFPwsml#OntoNFPwsml",
        "http://www.siti.disco.unimib.it/research/ontologies/LOOntology#LOOntology",
        "http://www.siti.disco.unimib.it/research/ontologies/NFPOntology#NFPOntology"
    }

instance policyForRepetitiveTravel memberOf ontoNFP#Policy
    annotations
dc#description hasValue "Policy Instance for Logistic Operator"
endAnnotations

ontoNFP#referenceFunctionalDescription hasValue _http://www.siti.disco.unimib.it/research/LOWS#WSOminiTransport"

ontoNFP#hasConditionDescription hasValue "This policy is applicable for client that required multiple travels"

ontoNFP#hasCondition hasValue policyCondition

ontoNFP#hasNfp hasValue offeredPaymentMethod
ontoNFP#hasNfp hasValue offeredPaymentDeadline
ontoNFP#hasNfp hasValue offeredInsurance
ontoNFP#hasNfp hasValue offeredBasePrice
ontoNFP#hasNfp hasValue offeredHoursToDelivery

axiom policyCondition definedBy
    applicable(policyForRepetitiveTravel) :- _[(numberOfTravelsSubscribed hasValue ?n and ?n > 1)] memberOf lo#LogisticOperatorClient.

Instance offeredPaymentMethod memberOf nfpo#ListPaymentMethod
hasExpression hasValue offeredPaymentMethodExpression

Instance offeredPaymentMethodExpression memberOf nfpo#ListPaymentMethodExpression
hasOperator hasValue ontoNFP#exact
hasParameters hasValue nfpo#carriageForward

Instance offeredPaymentDeadline memberOf nfpo#SingleValuePaymentDeadline
hasExpression hasValue offeredPaymentDeadlineExpression

Instance offeredPaymentDeadlineExpression memberOf nfpo#SingleValuePaymentDeadlineExpression
hasOperator hasValue ontoNFP#equal
hasParameter hasValue 15
hasUnit hasValue nfpo#days

Instance offeredInsurance memberOf nfpo#ListInsurance
hasExpression hasValue offeredInsuranceExpression

Instance offeredInsuranceExpression memberOf nfpo#ListInsuranceExpression
hasOperator hasValue ontoNFP#exact
hasParameters hasValue (nfpo#refundForLoss, nfpo#refundForDamage)

Instance offeredBasePrice memberOf nfpo#SingleValueBasePrice
hasExpression hasValue offeredBasePriceExpression

Instance offeredBasePriceExpression memberOf nfpo#SingleValueBasePriceExpression
hasOperator hasValue ontoNFP#equal
hasParameter hasValue 20
hasUnit hasValue nfpo#euro

Instance offeredHoursToDelivery memberOf nfpo#RangeHoursToDelivery
hasExpression hasValue offeredHoursToDeliveryExpression
```

D28.4 – NON-FUNCTIONAL PROPERTIES IN WEB SERVICES
Listing D.9: Gold Policy description

```xml
wsmlVariant _http://www.wsmo.org/wsml/wsml -- syntax/wsml -- flight
namespace {
  _http://www.siti.disco.unimib.it/research/ontologies/LOPolicy#,
  ontoNFP _http://www.siti.disco.unimib.it/research/ontologies/OntoNFPwsml#,
  dc _http://purl.org/dc/elements/1.1#,
  lo _http://www.siti.disco.unimib.it/research/ontologies/LOOntology#,
  nfpo _http://www.siti.disco.unimib.it/research/ontologies/NFPOntology#"
}

ontology policyInstanceLOOntology4
importsOntology { _http://www.siti.disco.unimib.it/research/ontologies/OntoNFPwsml#OntoNFPwsml,
  _http://www.siti.disco.unimib.it/research/ontologies/LOOntology#LOOntology,
  _http://www.siti.disco.unimib.it/research/ontologies/NFPOntology#NFPOntology" }

instance goldPolicy memberOf ontoNFP#Policy
  annotations
dc#description hasValue "Policy Instance for Logistic Operator"
endAnnotations

ontoNFP#ReferenceFunctionalDescription hasValue _http://www.siti.disco.unimib.it/research/LOWS# WSSouthItalyOrdinaryTransport"

ontoNFP#hasConditionDescription hasValue "This policy is applicable for gold client"
ontoNFP#hasCondition hasValue policyCondition
ontoNFP#hasNfp hasValue offeredPaymentMethod
ontoNFP#hasNfp hasValue offeredPaymentDeadline
ontoNFP#hasNfp hasValue offeredInsurance
ontoNFP#hasNfp hasValue offeredBasePrice
ontoNFP#hasNfp hasValue offeredHoursToDelivery

axiom policyCondition definedBy
  appliable(goldPolicy) :- ![UserTypology hasValue gold]] memberOf lo#LogisticOperatorClient.

instance offeredPaymentMethod memberOf nfpo#ListPaymentMethod
  hasExpression hasValue offeredPaymentMethodExpression

instance offeredPaymentMethodExpression memberOf nfpo#ListPaymentMethodExpression
  hasOperator hasValue ontoNFP#exact
  hasParameters hasValue nfpo#carriagePaid

instance offeredPaymentDeadline memberOf nfpo#SingleValuePaymentDeadline
  hasExpression hasValue offeredPaymentDeadlineExpression

instance offeredPaymentDeadlineExpression memberOf nfpo#SingleValuePaymentDeadlineExpression
  hasOperator hasValue ontoNFP#equal
  hasParameter hasValue 15
  hasUnit hasValue nfpo#days

instance offeredInsurance memberOf nfpo#ListInsurance
  hasExpression hasValue offeredInsuranceExpression

instance offeredInsuranceExpression memberOf nfpo#ListInsuranceExpression
  hasOperator hasValue ontoNFP#exact
  hasParameters hasValue { nfpo#refundForLoss, nfpo#refundForDamage }

instance offeredBasePrice memberOf nfpo#SingleValueBasePrice
  hasExpression hasValue offeredBasePriceExpression

instance offeredBasePriceExpression memberOf nfpo#SingleValueBasePriceExpression
  hasOperator hasValue ontoNFP#equal
```
Listing D.10: Silver Policy description

wsmlVariant _"http://www.wsmo.org/wsml/wsml−syntax/wsml−flight"
}

}

instance silverPolicy memberOf ontoNFP#Policy

annotations:
dc#description hasValue "Policy Instance for Logistic Operator"
endAnnotations

ontoNFP#referenceFunctionalDescription hasValue _"http://www.siti.disco.unimib.it/research/LOWS#WSSouthItalyOrdinaryTransport"

ontoNFP#hasConditionDescription hasValue "This policy is applicable for silver client"
ontoNFP#hasCondition hasValue policyCondition
ontoNFP#hasNfp hasValue offeredPaymentMethod
ontoNFP#hasNfp hasValue offeredPaymentDeadline
ontoNFP#hasNfp hasValue offeredInsurance
ontoNFP#hasNfp hasValue offeredBasePrice
ontoNFP#hasNfp hasValue offeredHoursToDelivery

axiom policyCondition
definedBy
  appliable(silverPolicy) :→ [userTypology hasValue silver] memberOf lo#LogisticOperatorClient.

ontoNFP#hasNfp hasValue offeredPaymentMethod
ontoNFP#hasNfp hasValue offeredPaymentDeadline
ontoNFP#hasNfp hasValue offeredInsurance
ontoNFP#hasNfp hasValue offeredBasePrice
ontoNFP#hasNfp hasValue offeredHoursToDelivery

instance offeredPaymentMethod memberOf nfpo#ListPaymentMethod

hasExpression hasValue offeredPaymentMethodExpression

instance offeredPaymentMethodExpression memberOf nfpo#ListPaymentMethodExpression

hasOperator hasValue ontoNFP#exact

hasParameters hasValue nfpo#carriageForward

instance offeredPaymentDeadline memberOf nfpo#SingleValuePaymentDeadline

hasExpression hasValue offeredPaymentDeadlineExpression

instance offeredPaymentDeadlineExpression memberOf nfpo#SingleValuePaymentDeadlineExpression

hasOperator hasValue ontoNFP#equal

hasParameter hasValue 45

hasUnit hasValue nfpo#days
instance offeredInsurance memberOf nfpo#ListInsurance
    hasExpression hasValue offeredInsuranceExpression

instance offeredInsuranceExpression memberOf nfpo#ListInsuranceExpression
    hasOperator hasValue ontoNFP#exact
    hasParameters hasValue nfpo#refundForDamage

instance offeredBasePrice memberOf nfpo#SingleValueBasePrice
    hasExpression hasValue offeredBasePriceExpression

instance offeredBasePriceExpression memberOf nfpo#SingleValueBasePriceExpression
    hasOperator hasValue ontoNFP#equal
    hasParameter hasValue 45
    hasUnit hasValue nfpo#euro

instance offeredHoursToDelivery memberOf nfpo#RangeHoursToDelivery
    hasExpression hasValue offeredHoursToDeliveryExpression

instance offeredHoursToDeliveryExpression memberOf nfpo#RangeHoursToDeliveryExpression
    hasOperator hasValue ontoNFP#interval
    hasMinParameter hasValue 72
    hasMaxParameter hasValue 96
    hasUnit hasValue nfpo#hour
Bibliography


