



WSMO Deliverable
D28.4 v0.1
**NON-FUNCTIONAL PROPERTIES IN
WEB SERVICES**

WSMO Working Draft – October 20, 2006

Authors:

Ioan Toma
Douglas Foxvog

Editors:

Ioan Toma

Reviewers:

Axel Polleres

This version:

<http://www.wsmo.org/TR/d28/d28.4/v0.1/20061020/>

Latest version:

<http://www.wsmo.org/TR/d28/d28.4/v0.1/>

Previous version:

<http://www.wsmo.org/TR/d28/d28.4/v0.1/20060924/>



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	4
2	Survey of Non-functional properties approaches	6
2.1	NFP and Web services	6
2.1.1	UDDI	6
2.1.2	OWL-S	6
2.1.3	SWSF	6
2.1.4	WSMO/WSML	7
2.1.5	O'Sullivan approach	7
2.2	NFP and Software Systems	8
2.2.1	Process ^{NFL}	9
2.2.2	CQML	10
2.2.3	QML	11
2.2.4	QuO	12
3	Non-functional properties tasks	14
4	Non-functional properties in WSMO/WSML	16
4.1	Non-functional properties types	16
4.2	Modeling non-functional properties	17
4.3	Attaching non-functional properties to WSMO elements	17
4.3.1	Model	18
4.3.2	Syntax	19
4.3.3	Example	19
5	Annotations	20
6	Ontologies for specifying non-functional properties	21
7	Conclusions and Future Work	23
A	Appendix	24
	Bibliography	59



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* and discusses the modeling of these properties in WSMO/WSML. Furthermore, it proposes a concrete solution towards a better support for non-functional properties in WSMO/WSML. 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 elements. 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 capture 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 view 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 requestor 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 [Agedal, 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:

```
attribute attributeID1 extends attributeID2{
  primitives primitiveAttributes;
  contribution kindOfContribution;
}
```

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

```
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 [Agedal and Earl F. Ecklund, 2002] and [Agedal, 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:

```
quality_characteristic delay{
  domain: decreasing numeric milliseconds;
}
```

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

```
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:

```
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*:

```
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:

```
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 than 5 msec all of them should have a value lower than 10 msec.

Contracts can be refined using the *refined by* construct. If *A* and *B* are two contracts then 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 than 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 IDL¹. 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:

```
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;
    transitions are
      transition Low_Cost -> Available :
        ReplMgr.adjust_degree_of_replication(ClientExpectedReplicas);
      transition Available -> Low_Cost :
        ReplMgr.adjust_degree_of_replication(ClientExpectedReplicas);
    end transitions;
  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 than 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 WSMO/WSML

This section investigates how non-functional properties could be better supported in WSMO/WSML. First, Section 4.1 talks about the types of non-functional properties. The modeling of these properties in WSMO/WSML is discussed afterwards in Section 4.2. Finally, Section 4.3 proposes a concrete solution on how to attach non-functional properties descriptions to services, goals and other elements in WSMO/WSML.

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. Additionally 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

Once a model is available 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 consist 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 possessed 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 axiom1
    postcondition definedBy axiom2
    assumption definedBy axiom3
    effect definedBy axiom4
  nonFunctionalProperty idNFP
    definition definedBy axiom5
  
```

This approach has the following advantages:

- The set of non-functional properties is not an explicit, finite set. 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.

This approach has the following disadvantages:

- A major disadvantage of this approach is that 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

```

Class annotations
  hasContributor type dc:contributor
  hasCoverage type dc:coverage
  hasCreator type dc:creator
  hasDate type dc:date
  hasDescription type dc:description
  hasFormat type dc:format
  hasIdentifier type dc:identifier
  hasLanguage type dc:language
  hasOwner type owner
  hasPublisher type dc:publisher
  hasRelation type dc:relation
  hasRights type dc:rights
  hasSource type dc:source
  hasSubject type dc:subject
  hasTitle type dc:title
  hasType type dc:type
  hasTypeOfMatch type typeOfMatch
  hasVersion type version

Class nonFunctionalProperties
  hasAnnotations type annotations
  importsOntology type ontology
  usesMediator type oolMediator
  hasDefinition type axiom

```

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. Non-functional properties that are represented by this class are for example *author* and *subject*. The set of annotation non-functional properties is not restricted to the properties listed in the above listing.
- A class which covers strictly the service related non-functional properties. This class is called `nonFunctionalProperties` 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 `nonFunctionalProperties` class in a way similar to which a capability is currently modelled in WSMO/WSML. `nonFunctionalProperties` 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 `nonFunctionalProperties` 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

```

annotations = 'annotations' attributevalue* 'endAnnotations'

nfp = 'nfp' id? header* definition |
      'nonFunctionalProperty' id? header* definition

definition = 'definition' axiomdefinition

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 *nonFunctionalProperties*, *capability* and *interface* are all constructs at the same level.

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

```

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#"
}

webService _ "http://example.org/ws"
  annotations
    dc#creator hasValue {"Ioan Toma"}
  endAnnotations

  nonFunctionalProperties price
    annotations
      dc#decription hasValue "if the client is older than 60 or younger than 10 years old the
        invocation price is lower than 10 euro"
    endAnnotations

  importsOntology { _ "http://www.wsmo.org/ontologies/nfp/priceNFPontology",
    _ "http://www.example.org/ontologies/humansOntology",
    _ "http://www.example.org/ontologies/clientsOntology" }

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

  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 'typeOfMatch'. 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).

Annotations of WSMO elements are currently modelled as key-value pairs, following a template like the one below:

```
annotationID hasValue value
```

, where `annotationID` is the identifier of the annotation (e.g `dc#contributor`), `hasValue` is the WSML keyword and `value` is the value for annotation (e.g `{ "Ioan", "Doug" }`).

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 contain 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 be 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 as 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 fulfil 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 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 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.



A Appendix

Based on the conceptual model provided in [O’Sullivan et al., 2005] we have developed a set of ontologies. The complete descriptions of these ontologies in WSML is provided in the following listings. The non-functional properties are modeled in 17 ontologies: (1) Locative Ontology, (2) Temporal Ontology, (3) Availability Ontology, (4) Obligation Ontology, (5) Price Ontology, (6) Payment Ontology, (7) Discounts Ontology, (8) Rights Ontology, (9) Trust Ontology, (10) Quality of Service Ontology, (11) Security Ontology, (12) Intellectual Property Ontology, (14) Rewards Ontology, (15) Provider Ontology, (16) Measures Ontology and (17) Currency Ontology.

Listing A.1: Locative Ontology

```

wsmliVariant _"http://www.wsmo.org/wsmli/wsmli-syntax/wsmli-rule"

namespace {
  _"http://www.wsmo.org/ontologies/nfp/locativeNFPontology#",
  dc _"http://purl.org/dc/elements/1.1#",
  xsd _"http://www.w3.org/2001/XMLSchema#",
  wsmli _"http://www.wsmo.org/wsmli/wsmli-syntax#",
  temp _"http://www.wsmo.org/ontologies/nfp/temporalNFPontology#",
  qu _"http://www.wsmo.org/ontologies/nfp/qualityNFPontology#",
  sec _"http://www.wsmo.org/ontologies/nfp/securityNFPontology#",
  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#subject hasValue {"Address", "Point"}
  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"
  wsmli#version hasValue "Revision : 1.0"
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"
  dc#relation hasValue { validLatitude, validLongitude, validPoleGeoLocation }
endNonFunctionalProperties
hasLatitude ofType (1) meas#Angle
hasLongitude ofType (1 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(._decimal("–90"), ?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(._decimal("–180"), ?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** ?NU, ofUnits **hasValue** meas#DegreeOfArc] **memberOf** meas#Angle.

concept CompassDirection

nonFunctionalProperties

dc#description **hasValue** "Direction concept definition"
dc#relation **hasValue** validCompassDirection

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)).

concept VehicularRoute **subConceptOf** LocativeEntity

nonFunctionalProperties

dc#description **hasValue** "VehicularRoute concept definition"

endNonFunctionalProperties

hasName ofType (1 1) .string

hasSpecification ofType RouteSpecification

intendedForVehicles ofType (1 *) Vehicle

possiblePathFor 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
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 LocativeEntity
  nonFunctionalProperties
    dc#description hasValue "GeopoliticalPlace concept definition"
  endNonFunctionalProperties

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

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

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

concept HumanResidenceArea subConceptOf GeographicalRegion
  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 "NationalTerritory 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
  hasAddressee ofType (0 1) Addressee
  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
  hasStreetDirectoryReference ofType StreetDirectoryReference
  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 inSurrounded

relation aboveGeneric (ofType LocativeEntity, ofType LocativeEntity, impliesType .boolean) subRelationOf spatiallyRelated

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

relation aboveHigerThan (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 facing (ofType LocativeEntity, ofType LocativeEntity, impliesType .boolean) subRelationOf spatiallyRelated

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

concept Addressee

nonFunctionalProperties

dc:description hasValue "Addressee 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 (0 1) Region
hasCountryCode ofType (0 1) nonNegativeInteger
hasNationalDirectDialPrefix 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

concept InternationalPrefixForRegion
nonFunctionalProperties
  dc:description hasValue "InternationalPrefixForRegion concept definition"
endNonFunctionalProperties
hasInternationalDirectDialPrefix ofType (1 1) nonNegativeInteger
forCallersFromRegion ofType (1 *) Region

concept IPAddress subConceptOf LocativeEntity
nonFunctionalProperties
  dc:description hasValue "IPAddress concept definition"
endNonFunctionalProperties
//todo – to be defined

concept EthernetAddress subConceptOf LocativeEntity
nonFunctionalProperties
  dc:description hasValue "EthernetAddress concept definition"
endNonFunctionalProperties
//todo – to be defined

concept StreetDirectory
nonFunctionalProperties
  dc:description hasValue "StreetDirectory concept definition"
endNonFunctionalProperties
hasEdition ofType (0 1) nonNegativeInteger
hasProvider ofType (0 1) Provider
hasISBNCode ofType (0 1) .string
hasPublicationTitle ofType (1 1) .string
hasPublicationDate ofType (0 1) temp#TemporalDate

concept StreetDirectoryReference subConceptOf LocativeEntity
nonFunctionalProperties
  dc:description hasValue "StreetDirectoryReference concept definition"
endNonFunctionalProperties
hasXPosition ofType (1 1) .string
hasYPosition ofType (1 1) .string
hasRegion ofType (0 *) Region
hasMapNumber ofType (1 1) .string
hasReference ofType (1 *) StreetDirectory

concept nonNegativeInteger subConceptOf .integer
nonFunctionalProperties
  dc:description hasValue "Non negative integer"
  dc#relation hasValue validNonNegativeInteger
endNonFunctionalProperties

axiom validNonNegativeInteger
definedBy
  !– ?x memberOf .integer and lessThan(?x, 0).

```

Listing A.2: Temporal Ontology

```

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#",
  wsmll_"http://www.wsmo.org/wsmll/wsmll-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"
nonFunctionalProperties
  dc#title hasValue "Temporal Ontology"
  dc#creator hasValue {"Ioan Toma"}
  dc#subject hasValue {"Time", "Date"}
  dc#description hasValue "Temporal 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/temporalNFPontology"
  dc#language hasValue "en-US"
  wsmll#version hasValue "Revision : 1.0"
endNonFunctionalProperties

concept TemporalEntity
  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
  endTime 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
 !- _#[monthNumber hasValue ?n] memberOf January and
 nEqual(?n, 1).

axiom validFebruary
definedBy
 !- _#[monthNumber hasValue ?n] memberOf February and
 nEqual(?n, 2).

axiom validMarch
definedBy
 !- _#[monthNumber hasValue ?n] memberOf March and
 nEqual(?n, 3).

axiom validApril
definedBy
 !- _#[monthNumber hasValue ?n] memberOf April and
 nEqual(?n, 4).

axiom validMay
definedBy
 !- _#[monthNumber hasValue ?n] memberOf May and
 nEqual(?n, 5).

axiom validJune
definedBy
 !- _#[monthNumber hasValue ?n] memberOf June and
 nEqual(?n, 6).

axiom validJuly
definedBy
 !- _#[monthNumber hasValue ?n] memberOf July and
 nEqual(?n, 7).

axiom validAugust
definedBy
 !- _#[monthNumber hasValue ?n] memberOf August and
 nEqual(?n, 8).

axiom validSeptember
definedBy
 !- _#[monthNumber hasValue ?n] memberOf September and
 nEqual(?n, 9).

axiom validOctober
definedBy
 !- _#[monthNumber hasValue ?n] memberOf October and
 nEqual(?n, 10).

axiom validNovember
definedBy
 !- _#[monthNumber hasValue ?n] memberOf November and
 nEqual(?n, 11).

axiom validDecember
definedBy
 !- _#[monthNumber hasValue ?n] memberOf December and



nEqual(?n, 12).

concept CalendarDay subConceptOf TemporalDate

nonFunctionalProperties

dc#description hasValue "OrdinalDate concept definition"

dc#relation hasValue validDayOfMonth

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

hasYear ofType _year

hasDayOfYear ofType DayOfYear

hasDayOfMonthNumber ofType _gday

hasDayOfWeekNumber ofType _integer

axiom validDayOfMonth

definedBy

!- _#[dayOfMonth hasValue ?n] memberOf CalendarDay and
(greaterEqual(?n, 31) or greaterEqual(1, ?n)).

axiom validDayOfFebruary

definedBy

!- _#[dayOfMonth hasValue ?n, ofMonth hasValue ?m] memberOf CalendarDay
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

!- ?x[hasDayOfWeekNumber hasValue ?n] memberOf CalendarDay
and (lessEqual(7, ?n) or greaterEqual(1, ?n)).

concept Monday subConceptOf CalendarDay

nonFunctionalProperties

dc#relation hasValue validMonday

endNonFunctionalProperties

concept Tuesday subConceptOf CalendarDay

nonFunctionalProperties

dc#relation hasValue validTuesday

endNonFunctionalProperties

concept Wednesday subConceptOf CalendarDay

nonFunctionalProperties

dc#relation hasValue validWednesday

endNonFunctionalProperties

concept Thursday subConceptOf CalendarDay

nonFunctionalProperties

dc#relation hasValue validThursday

endNonFunctionalProperties

concept Friday subConceptOf CalendarDay

nonFunctionalProperties

dc#relation hasValue validFriday

endNonFunctionalProperties

concept Saturday subConceptOf CalendarDay

nonFunctionalProperties

dc#relation hasValue validSaturday

endNonFunctionalProperties

concept 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
 !- .#[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 ofType (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 MonthsGranularity

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** OneMinuteGranularity

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(HoursDuration, WeeksDuration, 7.0)
relationInstance uom#unitMultiplicationFactor(DaysDuration, MonthsDuration, 29.5)
relationInstance uom#unitMultiplicationFactor(DaysDuration, YearsDuration, 365.242)
relationInstance uom#unitMultiplicationFactor(YearsDuration, DecadesDuration, 10.0)
relationInstance uom#unitMultiplicationFactor(DecadesDuration, CenturysDuration, 10.0)
relationInstance uom#unitMultiplicationFactor(CenturysDuration, MilleniasDuration, 10.0)

axiom validMonthNumber
definedBy
 !- _#[monthNumber **hasValue** ?n] **memberOf** ?month and
 ?month[monthNumber **hasValue** ?m] and nEqual(?n, m).

axiom validDayOfWeek
definedBy
 !- _#[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 validURCOffset
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** validDayOfWeek
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
definedBy
  !- ?X[value hasValue ?V] memberOf Minutes and (lessEqual(59,?V) or greaterThan(0,?V)).

concept SecondOfMinute
nonFunctionalProperties
  dc#description hasValue "Seconds concept definition"
  dc#relation hasValue validSecond
endNonFunctionalProperties
  value ofType .integer

axiom validSecond
definedBy
  !- ?X[value hasValue ?V] memberOf Seconds and (lessEqual(60,?V) or greaterEqual(0,?V)).

```

Listing A.3: Availability Ontology

```

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

namespace {
  _ "http://www.wsmo.org/ontologies/nfp/availabilityNFPontology#",
  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#",
  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 Ontology"
  dc#publisher hasValue "DERI Innsbruck"
  dc#contributor hasValue {"Doug Foxvog", "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/availabilityNFPontology#"
  dc#language hasValue "en-US"
  wsml#version hasValue "Revision : 1.0"
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.4: Obligation Ontology

```

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

namespace {
  _"http://www.wsmo.org/ontologies/nfp/obligationsNFPOntology#",
  dc _"http://purl.org/dc/elements/1.1#",
  xsd _"http://www.w3.org/2001/XMLSchema#",
  wsml _"http://www.wsmo.org/wsml/wsml-syntax#",
  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 {"Obligation"}
  dc:description hasValue "Obligations 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/obligationsNFPOntology"
  dc:language hasValue "en-US"
  wsml#version hasValue "Revision : 1.0"
endNonFunctionalProperties

concept Obligation
  nonFunctionalProperties
    dc:description hasValue "Obligation concept definition"
  endNonFunctionalProperties
  forProvider ofType (1 1) _iri //Provider

relation ProvierObligation ( 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
  dc:description hasValue "PricingObligationRelated concept definition"
endNonFunctionalProperties

```

Listing A.5: 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#",
  wsml _"http://www.wsmo.org/wsml/wsml-syntax#",
  loc _"http://www.wsmo.org/ontologies/nfp/locativeNFPOntology#",
  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 : 1.0"
endNonFunctionalProperties

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

concept MonetaryAmount
hasAmount ofType _decimal
hasCurrency ofType cur#Currency

concept AbsoultePrice subConceptOf {Price, MonetaryAmount}
nonFunctionalProperties
  dc:relation hasValue AbsolutePriceDefinition
endNonFunctionalProperties

axiom AbsolutePriceDefinition
definedBy
  !- ?price memberOf MonetaryAmount and
  ?price memberOf Price and
  naf (?price memberOf AbsoultePrice).

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
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
    ! – ?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.6: 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 : 1.0"
  endNonFunctionalProperties

  concept 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

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 ElectronicCashType

nonFunctionalProperties

dc:description hasValue "ElectronicCashInstrument concept definition"

endNonFunctionalProperties

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 TravellerChequeType subConceptOf ChequeType
nonFunctionalProperties
  dc:description hasValue "TravellerChequeType 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 loc#Region
  hasPaymentOptionCondition ofType _iri
  hasPaymentSchedule ofType PaymentSchedule
  providesTaxInvoice ofType (1 1) _boolean
  isPreferredPaymentOption ofType (1 1) _boolean
  hasInstrumentLocations ofType InstrumentLocations

concept InstrumentLocations
nonFunctionalProperties
  dc:description hasValue "InstrumentLocations concept definition"
endNonFunctionalProperties
  canBeFulfilledUsingPaymentInstrument ofType (1 1) PaymentInstrument
  atPaymentLocationOf 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

concept PaymentInstrumentType
  nonFunctionalProperties
    dc:description hasValue "PaymentInstrumentType concept definition"
  endNonFunctionalProperties
  value ofType .string
  hasCharacteristic ofType PaymentCharacteristicAndValue

concept PaymentCharacteristicAndValue
  nonFunctionalProperties
    dc:description hasValue "PaymentCharacteristicAndValue concept definition"
  endNonFunctionalProperties
  hasPaymentCharacteristic ofType (1 1) PaymentCharacteristic
  value ofType (1 1) PaymentCharacteristicValue

concept PaymentCharacteristic
  nonFunctionalProperties
    dc:description hasValue "PaymentCharacteristic concept definition"
  endNonFunctionalProperties
  hasType ofType (1 1) PaymentCharacteristicType
  value ofType (1 1) PaymentCharacteristicValue

concept PaymentCharactertisticType
  nonFunctionalProperties
    dc:description hasValue "PaymentCharactertisticType concept definition"
  endNonFunctionalProperties
  value ofType .string

concept PaymentCharacteristicValue
  nonFunctionalProperties
    dc:description hasValue "PaymentCharacteristicValue concept definition"
  endNonFunctionalProperties
  value ofType .string

```

Listing A.7: Discounts Ontology

```

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

namespace {
  "http://www.wsmo.org/ontologies/nfp/discountsNFPOntology#",
  dc "http://purl.org/dc/elements/1.1#",
  xsd "http://www.w3.org/2001/XMLSchema#",
  wsml "http://www.wsmo.org/wsml/wsml-syntax#",
  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#"
}

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 {"loan 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"
wsmo#version hasValue "Revision : 1.0"

```

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

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

isAvailableToHolders ofType Membership

concept ShareholderDiscount subConceptOf PayeeDiscount

nonFunctionalProperties

dc#description hasValue "ShareholderDiscount concept definition"

endNonFunctionalProperties

availableToShareholders ofType .iri //Provider

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

perctDiscount ofType (0 1) price#Percentage

forService ofType (1 1) .iri //Service

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
issuedBy ofType (1 1) _iri //could be a person, organization or any kind of provider

concept EarlyPaymentDiscount subConceptOf PaymentDiscount
nonFunctionalProperties
  dc#description hasValue "CouponPaymentDiscount concept definition"
endNonFunctionalProperties
hasEarlyPaymentOffset ofType temp#TemporalDuration
cutOffDate ofType (0 1) temp#TemporalDuration

```

Listing A.8: Rights Ontology

```

wsmIVariant _"http://www.wsmo.org/wsmI/wsmI-syntax/wsmI-rule"

namespace {
_"http://www.wsmo.org/ontologies/nfp/rightsNFPontology#",
  dc _"http://purl.org/dc/elements/1.1#",
  xsd _"http://www.w3.org/2001/XMLSchema#",
  wsmI _"http://www.wsmo.org/wsmI/wsmI-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#",
  ava _"http://www.wsmo.org/ontologies/nfp/availabilityNFPontology#",
  obl _"http://www.wsmo.org/ontologies/nfp/obligationsNFPontology#"
}

ontology _"http://www.wsmo.org/ontologies/nfp/rightsNFPontology"
nonFunctionalProperties
  dc#title hasValue "Rights Ontology"
  dc#creator hasValue {"Ioan Toma"}
  dc#subject hasValue {"Right"}
  dc#description hasValue "Rights 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/rightsNFPontology"
  dc#language hasValue "en-US"
  wsmI#version hasValue "Revision : 1.0"
endNonFunctionalProperties

concept Right
nonFunctionalProperties
  dc#description hasValue "Right concept definition"
endNonFunctionalProperties
hasName ofType (1 1) _string
hasTemporalValidity ofType (1 *) temp#TemporalEntity
isGrantedTo ofType (1 *) _iri //Service

concept RightOfWarranty subConceptOf Right
nonFunctionalProperties
  dc#description hasValue "RightOfWarranty concept definition"
endNonFunctionalProperties
hasWarrantiedItems ofType (1 *) WarrantiedItem
fulfilsWarranty ofType (1 1) _iri //Provider
hasWarrantyCondition ofType (1 *) _iri //Condition
hasWarrantyProcedureAvailability ofType (1 *) WarrantyProcedureAvailability

concept WarrantyProcedureAvailability
nonFunctionalProperties
  dc#description hasValue "WarrantyProcedureAvailability concept definition"
endNonFunctionalProperties
hasWarrantyProcedure ofType (1 1) Procedure
hasAvailability ofType (1 1) ava#Availability

concept WarrantiedItem
nonFunctionalProperties
  dc#description hasValue "WarrantiedItem concept definition"
endNonFunctionalProperties
isProvidedForItem ofType (1 1) Item
forWarrantyPeriod ofType (1 1) temp#TemporalDuration

```



```

concept RightOfAccess subConceptOf Right
nonFunctionalProperties
  dc:description hasValue "RightOfAccess concept definition"
endNonFunctionalProperties
endNonFunctionalProperties
  hasAccessTypeResource ofType (1 *) AccessTypeResource
  hasObligationTemporalDuration ofType ObligationTemporalDuration

concept ObligationTemporalDuration
nonFunctionalProperties
  dc:description hasValue "ObligationTemporalDuration concept definition"
endNonFunctionalProperties
  incursObligation ofType (1 1) obl#Obligation
  forExclusiveAccessDuration ofType (1 1) temp#TemporalDuration

concept AccessTypeResource
nonFunctionalProperties
  dc:description hasValue "AccessTypeResource concept definition"
endNonFunctionalProperties
  isAvailableForTemporalDuration ofType (1 1) temp#TemporalDuration
  underCondition ofType (1 1) .iri //Condition

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 obl#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
  refusalsAllowed 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 Element

hasPrivacyPolicyTypeCondition ofType (1 *) PrivacyPolicyCondition

withAvailability ofType 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) .iri //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 _iri //Condition
  hasResumptionCondition ofType _iri //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
  hasMaximumSuspensionPeriod 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
  isAvailableFor ofType temp#TemporalDuration
  hasAnnulmentProcedure ofType (0 1) _iri
  hasCoolingOffCondition ofType _iri //Condition

concept RightOfLiabilityLimitation subConceptOf Right
nonFunctionalProperties
  dc#description hasValue "RightOfLiabilityLimitation concept definition"
endNonFunctionalProperties
  hasLiabilityLimitCondition ofType _iri //Condition

concept RightOfRecourse subConceptOf Right
nonFunctionalProperties
  dc#description hasValue "RightOfRecourse concept definition"
endNonFunctionalProperties
  hasPeriodOfAvailabilityForRecourse ofType (1 1) temp#TemporalDuration
  hasRecourseProcedure ofType (0 1) Procedure
  isMediated ofType _boolean
  sMediatedThrough ofType Provider
  hasAdministeredJurisdiction ofType (1 1) Jurisdiction

concept Jurisdiction
nonFunctionalProperties
  dc#description hasValue "Jurisdiction concept definition"
endNonFunctionalProperties
  hasLocation ofType (1 1) loc#LocativeEntity
  hasLegislation ofType (1 1) Legislation

concept Legislation
nonFunctionalProperties
  dc#description hasValue "Legislation concept definition"
endNonFunctionalProperties
  hasAmendment ofType Legislation
  hasName ofType (1 1) _string
  hasYearOfIntroduction ofType (1 1) _year

  hasLegislation ofType (1 1) Legislation

//Relation supersededLegislation – tells if the Legislation provided as first argument
//supersedes the Legislation provided as second argument
relation supersededLegislation ( ofType Legislation, ofType Legislation, impliesType _boolean)

```

Listing A.9: Trust Ontology

```

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

namespace {
  _ "http://www.wsmo.org/ontologies/nfp/trustNFPontology#",
  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#"
}

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"
  wsmo#version hasValue "Revision : 1.0"
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

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

```

Listing A.10: Quality of Service Ontology

```

wsmoVariant "_http://www.wsmo.org/wsmo/wsmo-syntax/wsmo-rule"

namespace {
  "_http://www.wsmo.org/ontologies/nfp/qualityNFPontology#",
  dc "_http://purl.org/dc/elements/1.1#",
  xsd "_http://www.w3.org/2001/XMLSchema#",
  wsmo "_http://www.wsmo.org/wsmo/wsmo-syntax#",
  loc "_http://www.wsmo.org/ontologies/nfp/locativeNFPontology#",
  temp "_http://www.wsmo.org/ontologies/nfp/temporalNFPontology#",
  price "_http://www.wsmo.org/ontologies/nfp/priceNFPontology#"
}

ontology "_http://www.wsmo.org/ontologies/nfp/qualityNFPontology"
  nonFunctionalProperties
    dc#title hasValue "Quality of Service Ontology"
    dc#creator hasValue {"Ioan Toma"}
    dc#subject hasValue {"Non-functional properties"}
    dc#description hasValue "Quality of Service 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/qualityNFPontology"
    dc#language hasValue "en-US"
    wsmo#version hasValue "Revision : 1.0"
  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"
  endNonFunctionalProperties
  isApplicableToRegion ofType (0 1) loc#Region
  hasPercentageChange ofType (0 1) price#Percentage

concept AssessmentStandard subConceptOf Standard
  nonFunctionalProperties
    dc:description hasValue "AssessmentStandard concept definition"
  endNonFunctionalProperties
  hasParticipant ofType Provider
  hasRatingValue ofType (0 1) RatingValue

concept RankingScheme subConceptOf AssessmentStandard
  nonFunctionalProperties
    dc:description hasValue "RankingScheme concept definition"
  endNonFunctionalProperties
  hasRanking ofType (1 *) Ranking

concept Ranking
  nonFunctionalProperties
    dc:description hasValue "RankingScheme concept definition"
  endNonFunctionalProperties
  isValidForPeriod ofType (0 1) temp#TemporalDuration
  hasRankingPosition ofType (1 1) loc#nonNegativeInteger

concept Benchmark subConceptOf AssessmentStandard
  nonFunctionalProperties
    dc:description hasValue "Benchmark concept definition"
  endNonFunctionalProperties
  hasIndicator ofType (1 *) Indicator

concept Indicator
  nonFunctionalProperties
    dc:description hasValue "Indicator concept definition"
  endNonFunctionalProperties
  isApplicableIn ofType (0 1) loc#Region
  isApplicableDuring ofType (0 1) temp#TemporalEntity

```

Listing A.11: 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 : 1.0"
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
  hasAssignedIdentificationPoints ofType (1 1) loc#nonNegativeInteger

concept IdentificationType
nonFunctionalProperties
  dc#description hasValue "IdentificationType concept definition"
endNonFunctionalProperties
  value ofType _string

concept Confidentiality
nonFunctionalProperties
  dc#description hasValue "IdentificationRequirement concept definition"
endNonFunctionalProperties
  hasConfidentialityCondition ofType _iri
  hasConfidentialityStatement ofType Statement

concept ConfidentialityMechanism subConceptOf Confidentiality
nonFunctionalProperties
  dc#description hasValue "ConfidentialityMechanism concept definition"
endNonFunctionalProperties
  supportsEncryptionTechnique ofType (1 1) SupportsEncryptionTechnique

concept ConfidentialityAgreement subConceptOf Confidentiality
nonFunctionalProperties
  dc#description hasValue "ConfidentialityAgreement concept definition"
endNonFunctionalProperties
  hasValidityPeriod ofType (0 1) temp#TimeInterval
  isControlledWithin ofType (0 1) Jurisdiction

concept EncryptionTechnique
nonFunctionalProperties
  dc#description hasValue "ConfidentialityAgreement concept definition"
endNonFunctionalProperties
  hasName ofType (1 1) _string
  hasKeyLength ofType (1 1) loc#nonNegativeInteger
  hasDetails ofType (1 1) qua#Standard

```

Listing A.12: 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 {"loan 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"
wsmo#version hasValue "Revision : 1.0"

```

endNonFunctionalProperties

concept RegisteredIPRight

nonFunctionalProperties

```

dc#description hasValue "RegisteredIPRight concept definition"
endNonFunctionalProperties
hasIPRight ofType IPRight
hasApplicationNumber ofType (1 1) _decimal
hasRightStatus ofType (1 1) IPRightStatus
hasRegionOfProtection 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

```

concept IPRight

nonFunctionalProperties

```

dc#description hasValue "IPRight concept definition"
endNonFunctionalProperties

```

concept Trademark

nonFunctionalProperties

```

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

```

concept FigurativeMarkClassificationVienna

nonFunctionalProperties

```

dc#description hasValue "FigurativeMarkClassificationVienna concept definition"
endNonFunctionalProperties
value ofType (1 1) _string

```

concept TrademarkClassificationNice

nonFunctionalProperties

```

dc#description hasValue "TrademarkClassificationNice concept definition"
endNonFunctionalProperties
value ofType (1 1) _string

```

concept Patent

nonFunctionalProperties

```

dc#description hasValue "Patent concept definition"
endNonFunctionalProperties
hasPatentClassification ofType (1 1) PatentClassificationIPCStrasbourg
hasTitle ofType (1 1) _string
hasInvetor ofType (1 1) _string

```

concept PatentClassificationIPCStrasbourg

nonFunctionalProperties

```

dc#description hasValue "PatentClassificationIPCStrasbourg concept definition"
endNonFunctionalProperties
value ofType (1 1) _string

```

concept Design

nonFunctionalProperties

```

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

```

concept IDCLocarno

nonFunctionalProperties

```

dc#description hasValue "IDCLocarno concept definition"
endNonFunctionalProperties
value ofType (1 1) _string

```

concept RegisteredIPRight

nonFunctionalProperties

```

dc#description hasValue "RegisteredIPRight concept definition"

```



endNonFunctionalProperties

Listing A.13: Rewards Ontology

```

wsm:Variant _"http://www.wsmo.org/wsm/wsm-syntax/wsm-rule"

namespace {
  _"http://www.wsmo.org/ontologies/nfp/rewardsNFPontology#",
  dc _"http://purl.org/dc/elements/1.1#",
  xsd _"http://www.w3.org/2001/XMLSchema#",
  wsm _"http://www.wsmo.org/wsm/wsm-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#",
  ava _"http://www.wsmo.org/ontologies/nfp/availabilityNFPontology#",
  obl _"http://www.wsmo.org/ontologies/nfp/obligationsNFPontology#"
}

ontology _"http://www.wsmo.org/ontologies/nfp/rewardsNFPontology"
nonFunctionalProperties
  dc:title hasValue "Reward Ontology"
  dc:creator hasValue {"Ioan Toma"}
  dc:subject hasValue {"Reward"}
  dc:description hasValue "Reward 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/rewardNFPontology"
  dc:language hasValue "en-US"
  wsm:version hasValue "Revision : 1.0"
endNonFunctionalProperties

concept AccumulatedReward
nonFunctionalProperties
  dc:description hasValue "AccumulatedReward concept definition"
endNonFunctionalProperties
  isRewardsSchemeFor ofType (1 1) _iri //Service
  providesAccumulationDuring ofType (1 1) temp#TemporalEntity
  underRewardsAccumulationCondition ofType (1 1) _iri

concept AccumulatedPriceReward
nonFunctionalProperties
  dc:description hasValue "AccumulatedPriceReward concept definition"
endNonFunctionalProperties
  hasAccumulateReward ofType (1 1) AccumulatedReward
  hasRewardPoints ofType RewardPoint

concept RedeemableReward
nonFunctionalProperties
  dc:description hasValue "RedeemableReward concept definition"
endNonFunctionalProperties
  serviceOf ofType (1 1) _iri //Service
  isRewardsSchemeFor ofType (1 1) _iri //Service
  thatAllowsRedemptionDuring ofType (1 1) temp#TemporalEntity

concept RedeemablePaymentReward
nonFunctionalProperties
  dc:description hasValue "RedeemableReward concept definition"
endNonFunctionalProperties
  hasRedeemableReward ofType (1 1) RedeemableReward
  requiresPoints ofType RewardPoint
  hasRedemptionCondition ofType (1 *) _iri //Condition
  satisfiesPaymentFor ofType (1 1) obl#ServicePayment

concept RewardPoint
nonFunctionalProperties
  dc:description hasValue "RewardPoint concept definition"
endNonFunctionalProperties
  value ofType loc#nonNegativeInteger

```

Listing A.14: Provider Ontology

```

wsm:Variant _"http://www.wsmo.org/wsm/wsm-syntax/wsm-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#",
tmp _"http://www.wsmo.org/ontologies/nfp/temporalNFPOntology#",
obl _"http://www.wsmo.org/ontologies/nfp/obligationsNFPOntology#",
qua _"http://www.wsmo.org/ontologies/nfp/qualityNFPOntology#"
}

ontology _"http://www.wsmo.org/ontologies/nfp/providerNFPOntology"
nonFunctionalProperties
  dc#title hasValue "Provider Ontology"
  dc#creator hasValue {"Ioan Toma"}
  dc#subject hasValue {"Provider"}
  dc#description hasValue "Provider 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/providerNFPOntology"
  dc#language hasValue "en-US"
  wsml#version hasValue "Revision : 1.0"
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
  achievedConformanceOfStandard ofType (1 1) qua#Standard
  forService ofType (1 1) .iri //Service

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

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

```

Listing A.15: Measures Ontology

```

wsmiVariant _"http://www.wsmo.org/wsmi/wsmi-syntax/wsmi-rule"

namespace {
  _"http://www.wsmo.org/ontologies/nfp/measuresNFPOntology#",
  dc _"http://purl.org/dc/elements/1.1#",
  xsd _"http://www.w3.org/2001/XMLSchema#",
  wsmi _"http://www.wsmo.org/wsmi/wsmi-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"
  wsmi#version hasValue "Revision : 1.0"
endNonFunctionalProperties

concept MeasurableQuantity
  numUnits ofType _decimal
  ofUnits ofType (1) UnitOfMeasure

concept Distance subConceptOf MeasurableQuantity
  numUnits ofType (1) _decimal
  ofUnits ofType (1) UnitOfDistance

concept Angle subConceptOf MeasurableQuantity
  numUnits ofType (1) _decimal
  ofUnits ofType (1) UnitOfArc

concept UnitOfMeasure
concept UnitOfDistance subConceptOf UnitOfMeasure
concept UnitOfMass subConceptOf UnitOfMeasure
concept UnitOfTemperature subConceptOf UnitOfMeasure
concept UnitOfArc subConceptOf UnitOfMeasure

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

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

```



```

concept Granularity
  nonFunctionalProperties
    dc#description hasValue "Granularity concept definition"
  endNonFunctionalProperties
  value ofType _string

concept GranularityAndUnitOfMeasure
  nonFunctionalProperties
    dc#description hasValue "GranularityAndUnitOfMeasure concept definition"
  endNonFunctionalProperties
  hasUnitOfMeasure ofType UnitOfMeasure
  hasGranularity ofType Granularity

concept ItemGranularity
  nonFunctionalProperties
    dc#description hasValue "ItemGranularity concept definition"
  endNonFunctionalProperties
  hasCardinality ofType (1 1) _integer
  hasGranularity ofType (1 *) GranularityAndUnitOfMeasure
  hasItemNumber ofType (1 *) _integer //non negative integer
  relatesToItem ofType (0 1) Item

```

Listing A.16: Currency Ontology

```

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#",
  wsm1 _"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"
    wsm1#version hasValue "Revision : 1.0"
  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

```



Bibliography

- [ISO, 1986] (1986). Quality Vocabulary. Report: ISO 8402, ISO.
- [UNS, 2000] (2000). United Nations Standards Products and Services Codes. Technical report, United Nations.
- [Aagedal, 2001] Aagedal, J. (2001). *Quality of Service Support in Development of Distributed Systems*. PhD thesis, Faculty of Mathematics and Natural Science, University of Oslo.
- [Aagedal and Earl F. Ecklund, 2002] Aagedal, J. and Earl F. Ecklund, J. (2002). Modelling QoS: Towards a UML Profile. In *UML '02: Proceedings of the 5th International Conference on The Unified Modeling Language*, pages 275–289, London, UK. Springer-Verlag.
- [Bellwood, 2002] Bellwood, T. (2002). UDDI version 2.04 API specification. Available from <http://uddi.org/pubs/ProgrammersAPI-V2.04-Published-20020719.htm>.
- [Bellwood et al., 2002] Bellwood, T., Clément, L., Ehnebuske, D., Hatley, A., Hondo, M., Husband, Y., Januszewski, K., Lee, S., McKee, B., Munter, J., and von Riegen, C. (2002). UDDI Version 3.0.
- [Christensen et al., 2001] Christensen, E., Curbera, F., Meredith, G., and Weerawarana, S. (2001). Web Services Description Language (WSDL) 1.1. <http://www.w3.org/TR/wsdl>.
- [Chung, 1991] Chung, L. (1991). Non-Functional Requirements for Information Systems Design. In *Proceedings of the 3rd International Conference on Advanced Information Systems Engineering - CAiSE'91, April 7-11, 1991 Trondheim, Norway*, LNCS, pages 5–30. Springer-Verlag.
- [de Bruijn et al., 2005] de Bruijn, J., Lausen, H., Krummenacher, R., Polleres, A., Predoiu, L., Kifer, M., and Fensel, D. (2005). The Web Service Modeling Language WSML. Technical report, WSML. WSML Final Draft D16.1v0.21. <http://www.wsmo.org/TR/d16/d16.1/v0.21/>.
- [Eenoo et al., 2005] Eenoo, C. V., Hylooz, O., and Khan, K. M. (2005). Addressing Non-Functional Properties in Software Architecture using ADL. In *Proceedings of the 6th Australian Workshop on Software and Systems Architectures - AWSA'05, March 29, 2005, Brisbane, Australia*, pages 6–13.
- [Frølund and Koistinen, 1998] Frølund, S. and Koistinen, J. (1998). Qml: A Language for Quality of Service Specification. Technical report, Hewlett Packard. Available from <http://www.hpl.hp.com/techreports/98/HPL-98-10.html>.
- [Halpin, 2001] Halpin, T. (2001). *Information Modeling and Relational Databases: From Conceptual Analysis to Logical Design*. Morgan Kaufmann, San-Francisco.
- [Hauswirth et al., 2005] Hauswirth, M., Porto, F., and Vu, L.-H. (2005). P2p & qos-enabled service discovery specification. Working Draft D4.17, DIP.
- [Jin and Nahrstedt, 2002] Jin, J. and Nahrstedt, K. (2002). Classification and Comparison of QoS Specification Languages for Distributed Multimedia Applications. Technical report, Department of Computer Science, University of Illinois at Urbana-Champaign.



- [Keller et al., 2004] Keller, U., Lara, R., Polleres, A., Toma, I., Kiffer, M., and Fensel, D. (2004). Web Service Modeling Ontology – Discovery. Working draft, Digital Enterprise Research Institute (DERI). Available from <http://www.wsmo.org/2004/d5.1/v0.1>.
- [Loyall et al., 1998] Loyall, J., Bakken, D., Schantz, R., Zinky, J., Karr, D., and Vane-gas, R. (1998). Qos aspect languages and their runtime interactions in languages, compilers, and run-time systems for scalable computers. In *In O. David, editor, Lecture Notes in Computer Science 1511*. Springer-Verlag.
- [McGuinness and van Harmelen, 2004] McGuinness, D. L. and van Harmelen, F. (2004). Owl web ontology language overview (february 2004). Technical report, W3C. Available from <http://www.w3.org/TR/owl-features/>.
- [O’Sullivan et al., 2005] O’Sullivan, J., Edmond, D., and ter Hofstede, A. H. (2005). Formal description of non-functional service properties. Technical report, Queensland University of Technology, Brisbane. Available from <http://www.service-description.com/>.
- [Roman et al., 2005] Roman, D., Lausen, H., and (Ed.), U. K. (2005). Web service modeling ontology(WSMO). Working Draft D2v1.2, WSMO. Available from <http://www.wsmo.org/TR/d2/v1.2/>.
- [Rosa et al., 2002] Rosa, N. S., Cunha, P. R., Freire, L., and Justo, G. R. (2002). Process NFL: A language for describing non-functional properties. In *Proceedings of the 35th Annual Hawaii International Conference (HICSS), March 29, 2005, Hawaii, USA*, pages 3676–3685.
- [Röttger and Zschaler, 2003] Röttger, S. and Zschaler, S. (2003). CQML+: Enhance-ments to CQML. In *QoS in CBSE workshop 2003*, Toulouse, France.
- [SWSL, 2005a] SWSL (2005a). Semantic Web Service Framework. Available from <http://www.daml.org/services/swsf/1.1/overview/>.
- [SWSL, 2005b] SWSL (2005b). Semantic Web Service Framework. Available from <http://www.daml.org/services/swsf/1.1/swsl/>.
- [SWSL, 2005c] SWSL (2005c). Semantic Web Service Ontology. Available from <http://www.daml.org/services/swsf/1.1/swso/>.
- [The OWL Services Coalition, 2004] The OWL Services Coalition (2004). OWL-S 1.1 release. Available from <http://www.daml.org/services/owl-s/1.1/>.
- [W3C, 2003] W3C (2003). SOAP Version 1.2 Part 0: Primer.
- [Weibel et al., 1998] Weibel, S., Kunze, J., Lagoze, C., and Wolf, M. (1998). RFC 2413 - Dublin Core Metadata for Resource Discovery. Technical report.
- [Zinky et al., 1997] Zinky, J. A., Bakken, D. E., and Schantz, R. E. (1997). Archi-tectural support for quality of service for CORBA objects. *Theory and Practice of Object Systems*, 3(1).