Negotiability Constraints in WS-Agreement

Version 0.1
16 January 2004

Authors

Alain Andrieux (Globus Alliance / USC/ISI)

Asit Dan (IBM)

Kate Keahy, (Globus Alliance / ANL)

Heiko Ludwig (IBM)

John Rofrano (IBM)

Abstract

This document specifies negotiability constraints associated with agreement terms in WS-Agreement. Current WS-Agreement Specification allows the choice of one or more terms from a set of terms during negotiation. The current specification additionally enables customization of individual terms. It identifies elements of an agreement term as negotiable and defines constraints to be followed during negotiation. The specification also includes the initial discovery of agreement templates supported by a provider.

Table of Contents

1. Introduction

In the prior draft to WS-Agreement, terms of an agreement document that were negotiable were labeled as such but it was left open to which extent and within what range they were negotiable from the agreement document's sender point of view.

This proposed extension of the term type definition enables agreement document creators to indicate which elements and attributes of a proposed term can be modified and which values are considered eligible. For example, a maximum CPU usage term may only assume values between 60% and 90% because other values are technically or economically not feasible from a Grid service provider's point of view.

Agreement templates, predefined agreement with negotiable fields and associated constraints, are a very effective means of speeding up the negotiation process. Typically, those templates are provided by a service provider. The current draft of WS-Agreement provides no standardized way for an Agreement Factory to disclose a set of negotiable templates to a prospective client. Hence, a proposed extension to the Agreement Factory interface enables a prospective Grid service client to retrieve a set of negotiable templates to start the negotiation process more effectively.

1.1 Goals and Requirements

The goal of this proposal is to define an extension to the WS-Agreement draft that enables the definition of negotiation constraints on a sub-term level and to extend the interface of an Agreement Factory to provide an operation to retrieve supported Agreement Templates.

1.2 Notational Conventions

The keywords "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC 2119].

1.3 Namespaces

The following namespaces are used in this document:

Prefix	**Namespace**
xs	http://www.w3.org/2001/XMLSchema
wsa	http://GRAAP/WS-Agreement
wsp	http://schemas.xmlsoap.org/ws/2002/12/policy

2. Terminology and Concepts

(We introduce the following terms which are used throughout this document:

Term template – A *policy assertion* represents an individual preference, requirement, capability or other property.

Negotiability description – A set of negotiable *items*.

Item – A negotiable part of a term, potentially having a constraint associated to it.

3. Extension of the Term Type

The TermType definition is extended by an optional element NegotiabilityDescription that details further the Negotiability attribute:

```
<xs:complexType name=”TermType” abstract=”true” >
   <xs:sequence>
   <xs:element name="NegotiabilityDescription"
               type="wsa:NegDescriptionType"
               minOccurs="0"/>
   </xs:sequence>
   <xs:attributeGroup ref="wsp:CompositorAndAssertionAttributes" />
   <xs:attribute name=”Name” type=”xsd:NCName” />
   <xs:attribute name=”Negotiability” type=”wsa:NegotiabilityType”/>
</xs:complexType>

<xs:simpleType name="NegotiabilityType">
   <xs:restriction base="xsd:QName">
      <xs:enumeration value="wsa:Fixed"/>
      <xs:enumeration value="wsa:Negotiable"/>
   </xs:restriction>
</xs:simpleType>
```

The new element NegotiabilityDescription has the type NegDescriptionType. Its specification is optional in case the Negotiability attribute is Negotiable and a NegotiabilityDescription element is defined, an acceptable Agreement Document observes the constraint of the NegotiabilityDescription. Existing values in the terms are interpreted as defaults. If the Negotiability attribute is Fixed and there is a NegotiabilityDescription, it only serves as a reminder of the constraint that has been applied during the negotiation. In this case, the constraint of the NegotiabilityDescription must hold. If no NegotiabilityDescription is defined, the term is either fixed or negotiable without further constraint.

NegDescriptionType and NegItemType define the negotiable parts of a term and their constraints:

```
<xs:complexType name="NegDescriptionType">
    <xs:sequence>
        <xs:element name="Item" type="wsa:NegItemType"
                    maxOccurs="unbounded">
    </xs:sequence>
</xs:complexType>

<xs:complexType name="NegItemType">
     <xs:sequence>
         <xs:group ref="xs:simpleRestrictionModel"/>
     </xs:sequence>
     <xs:attribute name="name" type="xs:string" use="required"/>
     <xs:attribute name="location" type="xs:string" use="required"/>
</xs:complexType>
```

The NegDescriptionType has at least one Item element of NegItemType.

An element of NegItemType, which corresponds what one would loosely refer to as a field, has the following elements and attributes:

- An xs:simpleRestrictionModel is a description of a constraint on a simple type as defined in the XML Schema definition [XMLSchema2].
- An attribute name that is a unique identifier of the item.
- A location is an XPath expression referring to the position of the field in the term definition.

The following example of a modified JSDL term statement illustrates the use of a NegotiabilityDescription:

```
<?xml version="1.0" encoding="UTF-8"?>

<wsa:Agreement>

      <!--  ... more terms here -->

      <!--  This example of a Term taken and modified from jsdl has two
            negotiable values to which a constraint can be applied. Those
            negotiability constraints are grouped in the wsa:Negotiability
            element
       -->
      <jsdl:CPUUtilization wsp:Usage="wsp:required"
                           wsa:Negotiability=”wsa:Negotiable”>
            <jsdl:lowerBound>0</jsdl:lowerBound>
            <jsdl:upperBound>100</jsdl:upperBound>
            <wsa:NegotiabilityDescription>
                  <wsa:Item name="lower" location="jsdl:lowerBound">
                        <xsd:restriction>
                              <xsd:minInclusive value="80">
                              <xsd:maxExclusive value="90">
                        </xsd:restriction>
                  <wsa:Item>
                  <wsa:Item name="upper" location="jsdl:upperBound">
                        <xsd:restriction>
                              <xsd:enumeration value="95">
                              <xsd:enumeration value="97">
                              <xsd:enumeration value="99">
                              <xsd:enumeration value="99.9">
                        </xsd:restriction>
                  <wsa:Item>
            </wsa:NegotiabilityDescription>
      </jsdl:CPUUtilization>

      <!-- ... further terms ... -->

</wsa:Agreement>
```

CPUUtilization, borrowed and modified from JSDL, is a term defining a lower and an upper bound in a domain-specific extension to the TermType of WS-Agreement. The negotiability description restricts the lower bound to values between 80 and 90. The upper bound is restricted in form of an enumeration with defined values. Both typos of restrictions are defined in the XML Schema language.

4. Extension of the Agreement Factory Port Type

The extension of the AgreementFactory PortType comprises a new ServiceData element representing a named agreement template.

```
<xs:serviceData name=”supportedTemplate”
     type=”wsa:SupportedTemplateType”
     minOccurs=”0” maxOccurs=”unbounded”
     mutability=”mutable”
     modifiable=”false”
     nillable=”false”/>

<xs:complexType name=”SupportedTemplateType”>
   <xs:attribute name=”qname” type=”xsd:QName” />
   <xs:sequence>
      <xs:element name=”Template” type=”wsa:AgreementType” />
   </xsd:sequence>
</xsd:complexType>
```

An AgreementFactory MAY have an arbitrary number of service data elements (SDEs) supportedTemplate. An SDE contains an Agreement that is interpreted as a template and has a name assigned to it.

5. Acknowledgements

We would like to thank the following people for their contributions towards this specification: Karl Czajkowski, Robert Kearney, .. (TBD)

6. References

[RFC 2119]
“Key words for use in RFCs to Indicate Requirement Levels,” RFC 2119, S. Bradner (editor), March 1997.

[RFC 2396]
“Uniform Resource Identifiers (URI): Generic Syntax,” RFC 2396, T. Berners-Lee, R. Fielding, and L. Masinter (editors), August 1998.

[WS-Policy]
“Web Services Policy Framework (WS-Policy),” Don Box, Francisco Curbera, Maryann Hondo (Editor), Chris Kaler (Editor), Dave Langworthy, Anthony Nadalin, Nataraj Nagaratnam, Mark Nottingham, Claus von Riegen, and John Shewchuk, March 2003.

[XML-NS]
“Namespaces in XML,” W3C Recommendation, Tim Bray, Dave Hollander, and Andrew Layman (editors), 14 January 1999.

[XMLSchema1]

“XML Schema Part 1: Structures,” W3C Recommendation, Henry S. Thompson, David Beech, Murray Maloney, and Noah Mendelsohn (editors), 2 May 2001.

[XMLSchema2]

“XML Schema Part 2: Datatypes,” W3C Recommendation, Paul V. Biron, Ashok Malhotra (editors), 2 May 2001.

Appendix I. Schema Extensions

A normative copy of the XML Schema [XMLSchema1]:

```
<?xml version="1.0" encoding="UTF-8"?>

<xs:schema targetNamespace="http://www.ggf.org/wS-Agreement"
        elementFormDefault="qualified"
          xmlns:wsa="http://www.ggf.org/wS-Agreement"
          xmlns:xs="http://www.w3.org/2001/XMLSchema">


      <!-- other WS-Agreement specification

            .................

      -->

<xs:complexType name=”TermType” abstract=”true” >
   <xs:sequence>
   <xs:element name="NegotiabilityDescription"
               type="wsa:NegDescriptionType"
               minOccurs="0"/>
   </xs:sequence>
   <xs:attributeGroup ref="wsp:CompositorAndAssertionAttributes" />
   <xs:attribute name=”Name” type=”xsd:NCName” />
   <xs:attribute name=”Negotiability” type=”wsa:NegotiabilityType”/>
</xs:complexType>

<xs:complexType name="NegDescriptionype">
    <xs:sequence>
        <xs:element name="Item" type="wsa:NegItemType"
                    maxOccurs="unbounded">
    </xs:sequence>
</xs:complexType>

<!--
      Any xsd restriction can be applied to the field.

      attribut name is the identifier of the field to negotiate
      location is a relative xpath to the loction of the field in the term
 -->
<xs:complexType name="NegItemType">
     <xs:sequence>
         <xs:group ref="xs:simpleRestrictionModel"/>
     </xs:sequence>
     <xs:attribute name="name" type="xs:string" use="required"/>
     <xs:attribute name="location" type="xs:string" use="required"/>
</xs:complexType>

</xs:schema>
```