WS-Agreement Structure

Version 0.1
9 January 2004

Authors

Asit Dan (IBM)

Heiko Ludwig (IBM)

John Rofrano (IBM)

Abstract

This document specifies enhancement to the structure of WS-Agreement. Current specification of WS-Agreement refers to all aspects of agreement content as terms, and all agreement terms are listed as WS-Policy assertions. It is left to the parties to make sure that the essential elements of an agreement are captured as agreement terms. The current proposal defines structures for the essential elements of an agreement, and only the agreement specific terms are listed as WS-Policy assertions.

Status

This document is a proposal for enhancing WS-Agreement structure. The document will be submitted to the GRAAP WG as input for the overall WS-Agreement specification.

Table of Contents

1. Introduction

Why new structure? The current specification of WS-Agreement refers to all aspects of agreement content as terms, and all agreement terms are listed as WS-Policy assertions. While this simplified the schema, it is not a true representation of what an agreement is. Everything in an agreement is not a negotiable term. There is meta-data associated with an agreement such as those items specified in the Context (i.e., involved parties, service reference, etc.) which are not policy and should not be expressed as WS-Policy assertions. Further some of this metadata is essential to all agreements and should be REQUIRED. This proposal defines structures for the essential elements of an agreement, and only the agreement specific terms are listed as WS-Policy assertions.

1.1 Goals

The goal of WS-Agreement is to provide the mechanisms needed to enable Web Services applications to specify agreement terms for the usage of their service. Specifically, this specification defines the following:

- An XML-based structure called an Agreement which specifies both the context and terms under which this agreement applies.
- An XML-based structure called an AgreementContext which contains metadata about the involved parties and services.
- An XML-based structure called an AgreementTerm which contains domain specific Web Service agreement information.
- A core set of grammar elements to indicate how the contained agreement terms apply.

1.2 Example

The following example illustrates an agreement:

```
<wsag:Agreement Name="job123" xmlns:wsag="...">
  <wsag:AgreementContext>
    <wsag:AgreementInitiator>xs:anyURI</wsag:AgreementInitiator>
    <wsag:AgreementProvider>xs:anyURI</wsag:AgreementProvider>
    <wsag:TerminationTime>xs:DateTime</wsag:TerminationTime>
    <wsag:ServiceReference>
       <wsa:EndpointReference xmlns:wsa="...">
          <wsa:Address>xs:anyURI</wsa:Address>
          <wsa:PortType>xs:QName</wsa:PortType>
       </wsa:EndpointReference>
    </wsag:ServiceReference>
    <wsag:RelatedAgreements>...</wsag:RelatedAgreements>
  </wsag:AgreementContext>
  <wsag:TerminationTerms>
    ...
  </wsag:TerminationTerms>
  <wsag:MonitoringTerms>
    ...
  </wsag:MonitoringTerms>
```

```
020   <wsag:GuaranteeTerms>
021     ...
022   </wsag:GuaranteeTerms>
023   <tns:domain specific terms>
024      ...
025   </tns:domain specific terms>
026 </wsag:Agreement>
```

In this example, we illustrate the expression of an agreement.

Lines 1 to 26 represent a set of terms for a single agreement.

Lines 2 to 13 represent the context for this agreement.

Lines 14 to 16 represent the terms which specify under what conditions an agreement can be terminated.

Lines 17 to 19 represent the terms which specify the monitoring criteria available for this agreement.

Lines 20 to 21 represent the terms which specify the guarantee semantics for other terms that are specified.

Lines 23 to 25 represent the terms which specify domain specific terms about the agreement.

2. Notations and Terminology

This section specifies the notations, namespaces, and terminology used in this specification.

2.1 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].

Namespace [XML-NS] URIs (of the general form "some-URI") represents some application-dependent or context-dependent URI as defined in RFC 2396 [RFC 2396].

WS-Agreement is designed to work with the general Web Services framework including WSDL service descriptions [WSDL], UDDI service registrations [UDDI], and SOAP message structure and message processing model [SOAP11, SOAP12].

2.2 Namespaces

The XML namespace URI that MUST be used by implementations of this specification is:

```
        http://www.gridforum.org/namespaces/2004/01/agreement
```

A normative copy of the XML Schema [XMLSchema1] for WS-Agreement constructs may be retrieved by resolving the URI "http://www.gridforum.org/namespaces/2003/01/agreement".

The following namespaces are used in this document:

Prefix	Namespace
wsag	http://www.gridforum.org/namespaces/2004/01/agreement
ogsi	http://www.gridforum.org/namespaces/2003/03/OGSI
wsa	http://schemas.xmlsoap.org/ws/2003/03/addressing

wsp	http://schemas.xmlsoap.org/ws/2002/12/policy
xsd	http://www.w3.org/2001/XMLSchema
xsi	http://www.w3.org/2001/XMLSchema-instance

2.3 Terminology and Concepts

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

Agreement – An XML document that represents a contractual obligation between an agreement provider and an agreement initiator.

Agreement Provider – The party that is providing the ability to crate an agreement for using a web service

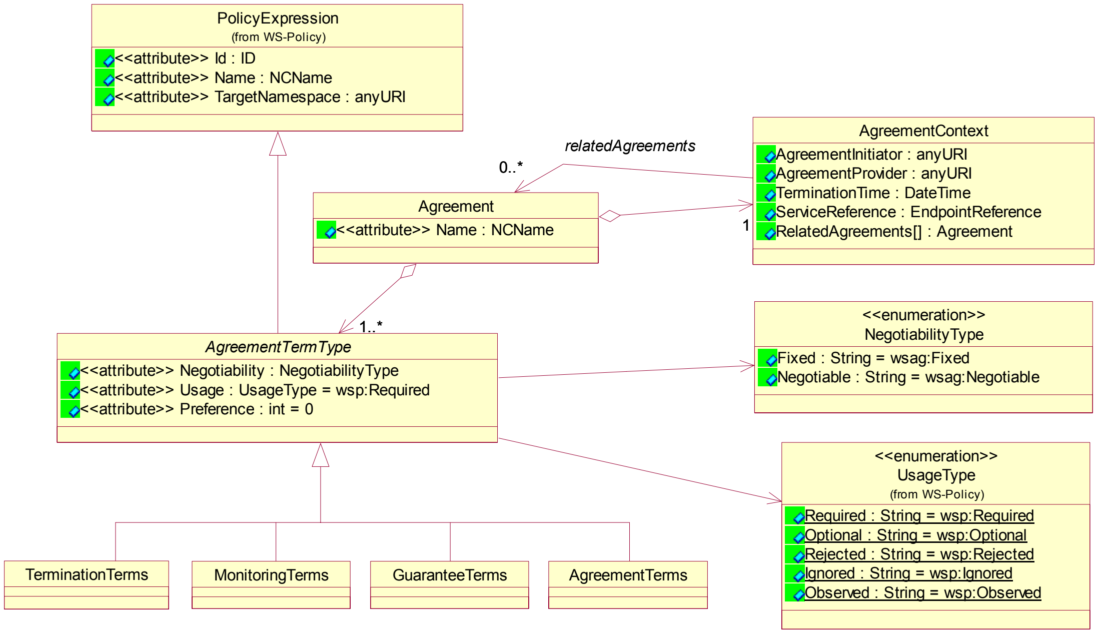
Agreement Initiator – The party that initiates the creation of an agreement

Service Provider – The party that provides the service specified in the agreement

Service Consumer – The party that uses the service specified in the agreement

3.0 Agreement Model

The UML for the agreement model can be seen in the figure below:


PolicyExpression
(from WS-Policy)
<<attribute>> Id : ID
<<attribute>> Name : NCName
<<attribute>> TargetNamespace : anyURI
relatedAgreements
0..*
AgreementContext
AgreementInitiator : anyURI
AgreementProvider : anyURI
TerminationTime : DateTime
ServiceReference : EndpointReference
RelatedAgreements[] : Agreement
1
Agreement
<<attribute>> Name : NCName
1..*
AgreementTermType
<<attribute>> Negotiability : NegotiabilityType
<<attribute>> Usage : UsageType = wsp:Required
<<attribute>> Preference : int = 0
<<enumeration>>
NegotiabilityType
Fixed : String = wsag:Fixed
Negotiable : String = wsag:Negotiable
<<enumeration>>
UsageType
(from WS-Policy)
Required : String = wsp:Required
Optional : String = wsp:Optional
Rejected : String = wsp:Rejected
Ignored : String = wsp:Ignored
Observed : String = wsp:Observed
TerminationTerms
MonitoringTerms
GuaranteeTerms
AgreementTerms

In this model we can see that the Agreement is composed of an AgreementContext and a collection of one or more AgreementTermTypes. AgreementTermType is a specialization of the PolicyExpression from WS-Policy and defines a Negotiability attribute as well as using the wsp:Usage and wsp:Preference attributes from WS-Policy. AgreementTermType is further specialized into TerminationTerms, MonitoringTerms, GuaranteeTerms, and AgreementTerms. Agreement providers should extend these specializations of AgreementTermType to describe the domain specific terms of their agreement. The AgreementContext contains a reference to the AgreementInitiator, the AgreementProvider, the service, which is an EndpointReference from WS-Addressing, and zero or more RelatedAgreements.

4.3 Extensible Agreement Language

An invocation to an AgreementFactory for creation of an Agreement service carries an agreement document containing a set of extensible terms, resulting in the formation of an agreement between the requester and the service provider, and embodiment of this agreement as an Agreement service. We define an agreement document type and a set of agreement term types that MUST be used both in creation of an agreement service and in expressing this agreement in an agreement document. Externally defined agreement languages MAY be used to extend WS-Agreement behaviors.

4.3.1 Agreement

An agreement document MUST be of the type wsag:AgreementType. The following describes the contents of this type (see Appendix I for XML Schema):

```
<wsag:Agreement Name="xs:NCName"?>
  <wsag:AgreementContext>wsag:AgreementContextType</wsag:AgreementContext>
  <wsag:TerminationTerms> ... </wsag:TerminationTerms> ?
  <wsag:MonitoringTerms> ... </wsag:MonitoringTerms> ?
  <wsag:GuaranteeTerms> ... </wsag:GuaranteeTerms> ?
  <wsag:AgreementTerms> ... </wsag:AgreementTerms> *
</wsag:Agreement>
```

The following describes the attributes and tags listed in the schema outlined above:

/wsag:Agreement

This is the outermost document tag which encapsulates the entire agreement. An agreement contains and agreement context and a collection of agreement terms.

/wsag:Agreement/@Name

This is an optional name that can be given to an agreement

/wsag:Agreement/AgreementContext

This is a REQUIRED element in the agreement and provides information about the agreement that is not specified in the terms such as who the involved parties are, what the services is that is being agree to, the length of the agreement, and references to any related agreements.

/wsag:Agreement/TerminationTerms

These terms specify how and under what conditions an agreement can be terminated. It may also include penalties for early termination.

/wsag:Agreement/MonitoringTerms

These terms specify what data will be available to monitor the agreement.

/wsag:Agreement/GuaranteeTerms

These terms specify the guarantees (both promises and penalties) that are associated with the other terms in the agreement.

/wsag:Agreement/AgreementTerms

These terms specify the domain specific terms that do not fit into one of the other categories. The majority of the terms that make up the agreement will fit into this category.

The term elements within an agreement document SHOULD be either one of the standard WS-Policy Compositors (wsp:OneOrMore, wsp:All, wsp:ExactlyOnce, wsp:Reference), or an element that extends the wsag:AgreementTermType. All agreements MUST have an AgreementContext and MAY have TerminationTerms, MonitoringTerms, GuaranteeTerms, and domain specific terms which derive from the base AgreementTerm.

4.3.2 AgreementContext

An agreement is scoped by its associated context that SHOULD include parties to an agreement, and additionally, SHOULD include reference to the service provided in support of the agreement. The context MAY also include other prior and/or related agreements. The new agreement thus augments prior related agreements, between the client and the service provider.

The <wsag:AgreementContext> element is used to describe the involved parties and the identify the service that the agreement is about. It can also optionally contain references to other related agreements.

```
<wsag:AgreementContext>
  <wsag:AgreementInitiator>xs:AnyURI</wsag:AgreementInitiator>
  <wsag:AgreementProvider>xs:anyURI</wsag:AgreementProvider>
  <wsag:TerminationTime>xs:DateTime</wsag:TerminationTime>
  <wsag:ServiceReference>
     <wsa:EndpointReference xmlns:wsa="..." xmlns:fabrikam="...">
        <wsa:Address>http://www.fabrikam123.com/acct</wsa:Address>
        <wsa:PortType>fabrikam:JobSubmissionPortType</wsa:PortType>
     </wsa:EndpointReference>
  </wsag:ServiceReference>
  <wsag:RelatedAgreements>...</wsag:RelatedAgreements>
</wsag:AgreementContext>
```

The following describes the attributes and tags listed in the schema outlined above:

/wsag:AgreementContext

This is the outermost tag which encapsulates the entire agreement context

/wsag:AgreementContext/AgreementInitiator

This is element identifies of the initiator of the agreement. The URI for an agreement provider MAY be an wsa:EndpointReference from WS-Addressing or MAY identify the initiator by more abstract naming, e.g. by security identity of the owner or operator.

/wsag:AgreementContext/AgreementProvider

This is element identifies the provider of the agreement. The URI for an agreement provider MAY be an wsa:EndpointReference from WS-Addressing or a Grid Service Handle (GSH) [OGSI] to an existing service or MAY instead identify the provider by more abstract naming, e.g. by security identity of the owner or operator.

/wsag:AgreementContext/TerminationTime

This element specifies the time at which this agreement is no longer valid. Agreement initiators MAY use this mechanism to negotiate Agreement service lifetime. Extended negotiation languages MAY define other mechanisms to negotiate lifetime integrated with other negotiation terms. The resulting negotiated lifetime MUST be exposed as wsag:TerminationTime and further negotiation MUST be possible through the basic OGSI mechanisms.

/wsag:AgreementContext/ServiceReference

This element defines references to the provided service for which the agreement terms are defined.

/wsag:AgreementContext/RelatedAgreements

This element defines references to any number of related agreements that define existing agreement terms which are being augmented via this agreement. The related agreements are represented in the agreement service as related agreement services (see Section 4.6).

/wsp:AgreementContext/{any}

Additional child elements MAY be specified to make additional agreement contexts but MUST NOT contradict the semantics of the parent element; if an element is not recognized, it SHOULD be ignored.

/wsp:AgreementContext/@{any}

Additional attributes MAY be specified but MUST NOT contradict the semantics of the owner element; if an attribute is not recognized, it SHOULD be ignored.

A wsag:AgreementContext element of type wsag:AgreementContextType MAY be used in an agreement to define an agreement context. Alternatively, the agreement context MAY be extended, through XSD extension of wsag:AgreementContextType, to define other attributes of the parties or services to an agreement.

4.3.3 AgreementTerms and AgreementTermType

Agreement terms are an extensible and interoperable means to author and convey the nature of an agreement. All of the terms are derived from AgreementTermType. AgreementTerm itself is where implementers can specify domain specific terms. These terms can be used to convey meaning during a negotiation exchange or afterwards to characterize an existing agreement. Not all agreement terms have the same obligations to be met. There are some agreement terms that are negotiable, and others that are non-negotiable, meaning that the expressing party is unable or unwilling to consider alternatives for the term. Some agreement terms may be required while others may be optional. Once agreed to by both parties, terms are considered to be observed with the exception of optional terms which can be ignored by the provider. These status concepts are common to all agreement terms and if expressed in a common way, can facilitate the understanding of the agreement and what is required to satisfy it.

```
<wsag:AgreementTerm Name=”xs:NCName” ?
                    wsp:Usage=”wsp:Required” ?
                    wsp:Preference=1 ?
                    wsag:Negotiability=”wsag:Fixed” ? >
   ...
</wsag:AgreementTerm>
```

The following describes the attributes and tags listed in the schema outlined above:

/wsag:AgreementTerm

This is element contains all of the terms which are domain specific and don't fit into the other term elements in the agreement document. Implementers should use it as the base tag from which all agreement terms are derived.

/wsag:AgreementTerm/@Name

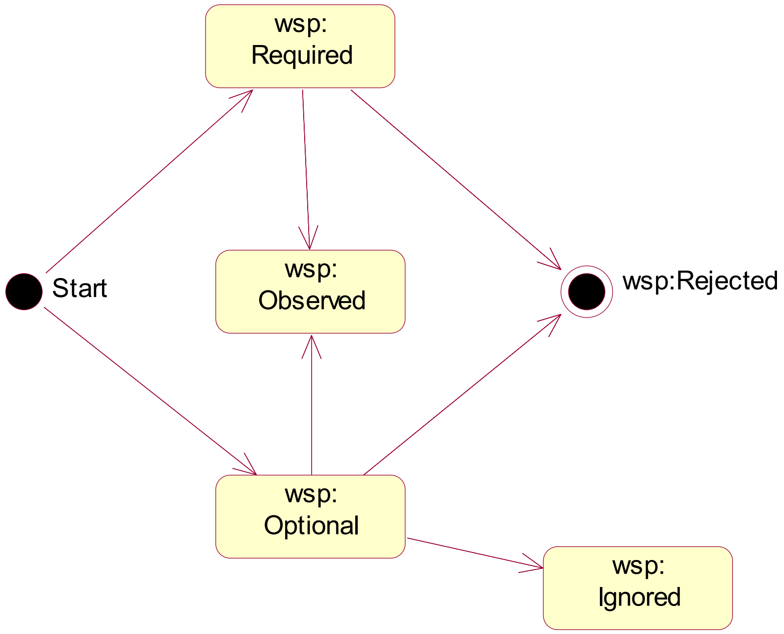
This optional NCName attribute allows a non-qualified name to be given to the term. This could be used to refer to terms within the same agreement.

/wsag:AgreementTerm/@wsp:Usage

This optional QName attribute identifies how the agreement term is processed. The following table describes the different values of this attribute. Additionally, the attribute is extensible; additional QName values in addition to those listed may be used.

Value	Meaning
wsp:Required	The term is required by the initiator for an agreement to be created. If the term cannot be met, the agreement provider MUST return a fault and change the wsp:Usage value to wsp:Rejected. If the agreement term is acceptable to the provider then the provider MUST change the wsp:Usage value to wsp:Observed to indicate its acceptance. It may also need to select a value for the term if the term was negotiable. If the term cannot be met, the provider SHOULD add information in the fault as to why this term could not be accepted and what it possibly can be accept.
wsp:Observed	The term has been accepted by both parties. Terms can transition their state to wsp:Observered from wsp:Required or wsp:Optional. This is true for terms specified by either the initiator or the provider. Terms that have a wsp:Usage value of wsp:Observed and a wsag:Negotiability value of wsag:Negotiable can still be renegotiated even though they are accepted.
wsp:Optional	The Term is desired but not required for an agreement to be created. An optional term implies the agreement term could be either ignored or observed. If the term cannot be met, the agreement provider MUST change the wsp:Usage value to wsp:Ignored to indicate it is not part of the agreement. If the agreement term is acceptable to the provider then the provider MAY change the wsp:Usage value to wsp:Observed to indicate its acceptance. Alternately the provider MAY keep the state as wsp:Optional to indicate it wants to keep it open as a future option to be observed.
wsp:Ignored	The term is not being used in this agreement. Only terms that were wsp:Optional can transition to the wsp:Ignored state.
wsp:Rejected	The term cannot be met by one of the parties.

The following state diagram shows the relationship between these states:



State Diagram for wsp:Usage in WS-Agreement

/wsag:AgreementTerm/@wsp:Preference

This optional attribute specifies the preference of this particular alternative. The preference is expressed as an xs:int. The higher the value of the preference, the greater the weighting of the expressed preference. If no preference is specified, a value of zero is assumed.

/wsag:AgreementTerm/@wsag:Negotiability

This optional attribute specifies whether or not this term is fixed (wsag:Fixed) or negotiable (wsag:Negotiable). Terms which are fixed must be met or a fault MUST be thrown. Terms which are negotiable allow the provider to create an agreement with alternate values than the ones specified. If this attribute is not specified, the default value is wsp:Fixed.

The base type of all agreement terms in WS-Agreement MUST be wsag:AgreementTermType, which is an extension of the WS-Policy wsp:PolicyExpression type which makes it a policy assertion. This base type uses the wsp:Usage attribute for specifying if a term is Required, Optional, Observed, Ignored, or Rejected. It also uses the <wsp:All>, <wsp:OneOrMore>, and <wsp:ExactlyOne> operators to group terms. Finally it uses the wsp:Preference attribute to specify preferences when groups have one or more selectable terms. A well defined agreement MAY include terms on the service guarantees, such as service level goals to be supported by the provider [Ludwig SLA], importance of this term and/or business impact for violating these terms, as well as terms on the manageability of this agreement, such as termination time, monitoring criteria, and termination criteria.

The wsp:Usage and wsag:Negotiability attributes of an agreement term define the status of this agreement term. The wsp:Usage status specifies whether a term is agreed by both the client and the provider (wsp:Observed), whether the term is required by the client or provider (wsp:Required), or whether the term is optional to the client or ignored by the provider (wsp:Optional or wsp:Ignored), or has been rejected (wsp:Rejected). The negotiability attribute specifies whether the term is fixed (i.e., required by client or provider, or agreed by both), or negotiable.

4.3.3.1 Term State Definitions for wsag:Negotiability

The table below illustrates how the wsp:Usage states relate to the wsa:Negotiability states:

Negotiability State Combinations

	wsag:Fixed	wsag:Negotiable
wsp:Required	The term is required but no agreement has been reached on it yet. The contents of the term cannot be changed and MUST be agreed to (i.e., changed to wsp:Observed) for the agreement to be valid.	The term is required but no agreement has been reached on it yet. The contents of the term can be changed within the range of specified negotiable parameters and MUST be agreed to (i.e., changed to wsp:Observed) for the agreement to be valid.
wsp:Observed	There is an agreement to provide the term. The contents of the term cannot be changed.	There is an agreement to provide the term. The contents of the term can still be renegotiated.
wsp:Optional	The term is desired but no agreement has been reached on it yet. The contents of the term cannot be changed but the term MAY be ignored. If the term is not ignored it MUST be agreed to (i.e., changed to wsp:Observed) for the agreement to be valid..	The term is desired but no agreement has been reached on it yet. The contents of the term can be changed within the range of specified negotiable parameters but the term MAY be ignored. If the term is not ignored it MUST be agreed to (i.e., changed to wsp:Observed) for the agreement to be valid..
wsp:Ignored	The term is not being used.	The term is not being used.
Wsp:Rejected	The term has been rejected.	The term has been rejected.

By modeling agreement terms as WS-Policy assertions, more complex term associations can be expressed. The use of wsp:Preference in conjunction with <wsp:All>, <wsp:ExactlyOne>, and <wsp:OneOrMore> can express groups of terms which represent choices to select from.

The following example illustrates the use of these WS-Policy artifacts:

```
<wsag:Agreement>
   <wsp:ExactlyOne wsp:Usage="wsp:Required">
      <wsag:term1 wsp:Preference="1" wsa:Negotiability="wsa:Fixed">
```

```
         ...
      </wsag:term1>
      <wsag:term2 wsp:Preference="100" wsa:Negotiability="wsa:Fixed">
         ...
      </wsag:term2>
   <wsp:ExactlyOne>
</wsag:Agreement>
```

The example above, when specified by a requestor, specifies two term which are not negotiable (wsa:Negotiability="wsa:Fixed") but are a disjunction in that the provider only has to satisfy one of them (<wsp:ExactlyOne>) and the requestor prefers that the provider satisfy the second one (wsp:Preference="100") if possible. Combinations of these WS-Policy assertion tags, can be used to express agreements containing complex conjunctions and disjunctions of terms.

4.3.4 TerminationTerms

The base type TerminationTermType is an abstract type that can be extended to define a terms for termination of agreement and details of the finalization process. A well defined agreement termination term may include who can terminate this agreement and under what conditions the agreement may be terminated (say, multiple violations within a time window, client exceeding an agreed upon request rate, presence of a higher valued agreement, etc). Note that both the client as well as provider MAY initiate an early termination. The termination terms MAY also include penalty assessment on the party responsible for this termination.

```
<wsag:Agreement>
   ...
   <wsag:TerminationTerms>
      ... (need to be more specific here)
     ⇨ Proposal: Define -
           o  Who (initiator, provider, or both) can modify or terminate
              this agreement
           o  Under what condition -
                 ▪  reference to guarantee terms, if violated multiple
                    times during lifetime and/or a time window
                 ▪  Violation of terms agreed by service clients, e.g.,
                    exceeding request rate
   </wsag:TerminationTerms>
</wsag:Agreement>
```

/wsag:TerminationTerms

This element delineates the section of the agreement document that contains the termination terms.

4.3.5 MonitoringTerms

The base type wsa:MonitoringTermType can be extended to define criteria for monitoring agreement terms. A set of well defined agreement monitoring criteria MAY include both the set of terms to be monitored by the client, as well as monitoring details, such as how often the monitoring data can be received by the client (without causing undue burden on the provider) or the details on receiving audit data after the agreement service becomes unavailable. The set of values to be monitored MAY include status with respect to agreement terms (say, violation of a response time goal) as well as values over which the guarantees are expressed (say, response time itself). The values to be monitored dynamically becomes available as monitoredValue service data (see section 6.2.7).

```
<wsag:Agreement>
   ...
   <wsag:MonitoringTerms>
      ... (need to be more specific here)
     ⇨ Proposal: Define -
           o  What service attributes/state will be exposed to initiator
              for monitoring,
                 ▪  List of attributes, e.g., response time, memory size
                 ▪  List of guarantee terms whose status will be monitored
                    e.g., violation
           o  If via notification, how often?
           o  If via query, form of aggregation, e.g., current value, or
              history for a specific time period
   </wsag:MonitoringTerms>
</wsag:Agreement>
```

/wsag:MonitoringTerms

This element delineates the section of the agreement document that contains the monitoring terms.

4.3.6 GuaranteeTerms

... this section needs to be written ...

```
<wsag:Agreement>
   ...
   <wsag:GuarenteeTerms>
      ... (need to be more specific here)
     ⇨ Proposal: Define -
           o  What's being guaranteed, i.e., service level objective
           o  Under what qualifying condition
           o  Importance to initiator and/or impact on provider for
              meeting this guarantee
   </wsag:GuarenteeTerms>
</wsag:Agreement>
```

/wsag:GuarenteeTerms

This element delineates the section of the agreement document that contains the guarantee terms.

5. Acknowledgements

We would like to thank the following people for their contributions towards this specification:

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.

[SOAP11]

“SOAP: Simple Object Access Protocol 1.1,” W3C Note, Don Box, David Ehnebuske, Gopal Kakivaya, Andrew Layman, Noah Mendelsohn, Henrik Frystyk Nielsen, Satish Thatte, and Dave Winer, 08 May 2000.

[SOAP12]

“SOAP Version 1.2 Part 1: Messaging Framework,” W3C Candidate Recommendation, Martin Gudgin, Marc Hadley, Noah Mendelsohn, Jean-Jacques Moreau, and Henrik Frystyk Nielsen (editors), 19 December 2002.

[UDDI]

“UDDI Version 3.0,” OASIS Committee Specification, Tom Bellwood, Luc Clément, David Ehnebuske, Andrew Hately, Maryann Hondo, Yin Leng Husband, Karsten Januszewski, Sam Lee, Barbara McKee, Joel Munter, and Claus von Riegen (editors), 19 July 2002.

[WS-Addressing]

“Web Services Addressing (WS-Addressing),” Adam Bosworth, Don Box (Editor), Erik Christensen, Francisco Curbera (Editor), Donald Ferguson, Jeffrey Frey, Chris Kaler, David Langworthy, Frank Leymann, Steve Lucco, Steve Millet, Nirmal Mukhi, Mark Nottingham, David Orchard, John Shewchuk, Tony Storey, and Sanjiva Weerawarana, 13 March 2003.

[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.

[WSDL]

“Web Services Description Language (WSDL) 1.1,” W3C Note, Erik Christensen, Francisco Curbera, Greg Meredith, and Sanjiva Weerawarana, 15 March 2001.

[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.

Appendix I. XML Schema

A normative copy of the XML Schema [XMLSchema1] for WS-Agreement constructs may be retrieved by resolving the URI “http://www.gridforum.org/namespaces/2004/01/agreement”. A non-normative copy of that Schema is included in-line below for convenient reference.

```
<xs:schema
  targetNamespace="http://www.gridforum.org/namespaces/2004/01/agreement"
    xmlns:xs="http://www.w3.org/2001/XMLSchema"
    xmlns:wsp="http://schemas.xmlsoap.org/ws/2002/12/policy"
    xmlns:wsu="http://schemas.xmlsoap.org/ws/2002/07/utility"
    elementFormDefault="qualified"
    blockDefault="#all">

<!-- wsag:AgreementType -->
<xsd:complexType name="AgreementType">
   <xsd:complexContent>
      <xsd:sequence>
         <xsd:element name="AgreementContext"
                      type="AgreementContextType" use="required"/>
         <xsd:element name="TerminationTerms"
                      type="TerminationTermType" use="optional"/>
         <xsd:element name="MonitoringTerms"
                      type="MonitoringTermType" use="optional"/>
         <xsd:element name="GuaranteeTerms"
                      type="GuaranteeTermType" use="optional"/>
         <xsd:element name="AgreementTerms"
                      type="AgreementTermType" use="optional"
                      minOccurs="0" maxOccurs="unbounded"/>
      </xsd:sequence>
   </xsd:complexContent>
</xsd:complexType>
<xsd:element name=”Agremeent” type=”AgreementType” />

<!-- wsag:AgreementContextType -->
<xsd:complexType name=”AgreementContextType”>
  <xsd:complexcontent>
    <xsd:sequence>
      <xsd:element name=”AgreementInitiator” type=”xsd:anyURI” />
      <xsd:element name=”AgreementProvider” type=”xsd:anyURI” />
      <xsd:element name="TerminationTime" type="xsd:dateTime"
                   xsi:nillable="true"/>
      <xsd:element name=”Service” type=”wsa:EndpointReference” />
      <xsd:element name=”RelatedAgreements”
                   type=”wsa:EndpointReference”
                   minOccurs=”0”
                   maxOccurs=”unbounded”/>
      <xsd:any namespace="##other"
                   minOccurs="0" maxOccurs="unbounded"/>
     </xsd:sequence>
   </xsd:complexContent>
</xsd:complexType>
<xsd:element name=”AgreementContext” type=”wsag:AgreementContextType” />

<!-- wsag:AgreementTermType -->
<xsd:complexType name=”AgreementTermType” abstract=”true” >
   <xsd:attributeGroup ref="wsp:CompositorAndAssertionAttributes" />
   <xsd:attribute name=”Name” type=”xsd:NCName” />
   <xsd:attribute name=”Negotiability” type=”wsag:NegotiabilityType”/>
   <xsd:any namespace="##other"
            minOccurs="0" maxOccurs="unbounded"/>
</xsd:complexType>
<xsd:element name=”AgreementTerms” type=”wsag:AgreementTermType” />

<!-- wsag:NegotiabilityType -->
<xsd:simpleType name="NegotiabilityType">
   <xsd:restriction base="xsd:QName">
      <xsd:enumeration value="wsag:Fixed"/>
      <xsd:enumeration value="wsag:Negotiable"/>
   </xsd:restriction>
</xsd:simpleType>

<!-- wsag:TerminationTermType -->
<xsd:complexType name=”TerminationTermType” >
   <xsd:complexcontent>
      <xsd:extension base=”wsa:AgreementTermType”
      </xsd:extension>
   </xsd:complexContent>
</xsd:complexType>
<xsd:element name=”TerminationTerms” type=”wsag:TerminationTermType” />

<!-- wsag:MonitoringTermType -->
<xsd:complexType name=”MonitoringTermType” >
   <xsd:complexcontent>
      <xsd:extension base=”wsa:AgreementTermType”>
      </xsd:extension>
   </xsd:complexContent
</xsd:complexType>
<xsd:element name=”MonitoringTerms” type=”wsag:MonitoringTermType” />
```

```
<!-- wsag:GuaranteeTermType -->
<xsd:complexType name=”GuaranteeTermType” >
   <xsd:complexcontent>
      <xsd:extension base=”wsa:AgreementTermType”>
      </xsd:extension>
   </xsd:complexContent
</xsd:complexType>
<xsd:element name=”GuaranteeTerms” type=”wsag:GuaranteeTermType” />

</xs:schema>
```