



D16.3v0.1. WSML/XML - An XML Syntax for WSML

WSML Working Draft 9 September 2004

This version:

<http://www.wsmo.org/2004/d16/d16.3/v0.1/20040909/>

Latest version:

<http://www.wsmo.org/2004/d16/d16.3/v0.1/>

Previous version:

<http://www.wsmo.org/2004/d16/d16.3/v0.1/20040322/>

Editors:

Jos de Bruijn

Authors:

Jos de Bruijn

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

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

Abstract

This document defines WSML/XML - an XML syntax for WSML, based on the human-readable syntax. Furthermore, it provides an extensive example of the use of the syntax for ontology modeling.

Table of contents

- [1. Introduction](#)
- [2. WSML/XML Syntax](#)
- [3. Example](#)
- [4. Future Work](#)
- [References](#)
- [Appendix A. XML Schema for WSML/XML](#)
- [Appendix B. An example of a WSML/XML ontology](#)

1. Introduction

This document provides an XML syntax for the Web Service Modeling Language WSML, based on the WSMO conceptual model and the WSML BNF grammar provided by [\[Roman et al., 2004\]](#). This syntax, henceforth referred to as WSML/XML, can be used to encode arbitrary WSML and can be used to encode anything, which can be encoded in the BNF syntax provided by [\[Roman et al., 2004\]](#).

Currently, WSML/XML only covers the "umbrella" syntax presented in D2 and does not yet support specific WSML variants, such as WSML-Flight, WSML-Core and WSML-Rule.

2. WSML/XML Syntax

The XML Schema (see [Appendix A](#)) captures the syntax of WSML/XML, as described earlier.

3. Example

The XSLT stylesheet (see [Appendix B](#)) transforms WSML/XML syntax back to the original, "human-readable" WSML/BNF syntax.

4. Future Work

The example in Appendix B is to be expanded to include goal, web service and mediator descriptions, as well as richer logical expressions.

Based on the XML syntax presented in this document, an XML syntax for WSML-Core is to be defined. The hard deadline for the syntax to be (pre-)final is end of September 2004.

The use of built-in predicate and function symbols is to be evaluated. The current strategy is to rely on identifiers for these built-ins, but it would be possible to include some of these built-ins as XML elements.

The use of path expressions is to be evaluated. Currently, a path expression is seen as an ID. We need to evaluate whether we need additional XML elements for path expressions.

After the XML Schema has been finalized, converters need to be built to convert from and to the human-readable syntax of WSML. The latter can be done using an XSLT script, as was done for previous versions (March 2004) of WSML/XML.

References

[Roman et al., 2004] D. Roman, H. Lausen, and U. Keller (eds.): *Web Service Modeling Ontology - Standard (WSMO - Standard)*, WSMO deliverable D2 version 1.0 Working Draft 27 August 2004. available from <http://www.wsmo.org/2004/d2/v1.0/20040827/>.

Appendix A. XML Schema for WSML/XML

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- edited with XMLSPY v2004 rel. 3 U (http://www.xmlspy.com) by University
Innsbruck / Department of Computer Science (University Innsbruck / Department of Computer Science) -->
<xs:schema targetNamespace="http://www.wsmo.org/2004/d16/16.3/" xmlns="http://www.wsmo.org/2004/d16/16.3/"
xmlns:exp="http://www.wsmo.org/2004/d16/d16.3" xmlns:xs="http://www.w3.org/2001/XMLSchema"
elementFormDefault="qualified" attributeFormDefault="unqualified">
  <!-- version: 2004-09-09 -->
  <!-- author: Jos de Bruijn -->
  <!-- this document provides an XML Schema for the X
ML version of the WSML syntax. -->
  <!-- this document is based on the BNF syntax for WSML
(WSMO D2, Appendix B) v1.0 dated 2004-08-27 -->
  <!-- there is currently elementary support for logical expressions; path expressions
are not supported and no built-ins are defined. It is assumed that built-ins are specific
function or predicate symbols with a certain identifier. Path expressions are treated currently
as a kind of ID. -->
  <xs:element name="wsmml">
    <xs:annotation>
      <xs:documentation>the root element, containing the WSML document</xs:documentation>
    </xs:annotation>
    <xs:complexType>
      <xs:choice minOccurs="0" maxOccurs="unbounded">
        <xs:element name="ontology" type="ontologyType"/>
        <xs:element name="webService" type="webServiceType"/>
        <xs:element name="goal" type="goalType"/>
        <xs:element name="mediator" type="mediatorType"/>
      </xs:choice>
    </xs:complexType>
  </xs:element>
  <xs:complexType name="goalType">
    <xs:sequence>
      <xs:element name="nonFunctionalProperties" type="nonFunctionalPropertiesType" minOccurs="0"/>
      <xs:element name="importOntology" type="wsmmlID" minOccurs="0" maxOccurs="unbounded"/>
      <xs:element name="ooMediator" type="wsmmlID" minOccurs="0" maxOccurs="unbounded"/>
      <xs:element name="ggMediator" type="wsmmlID" minOccurs="0" maxOccurs="unbounded"/>
      <xs:element name="postCondition" type="axiomType" minOccurs="0" maxOccurs="unbounded"/>
      <xs:element name="effect" type="axiomType" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:attribute name="id" type="wsmmlID" use="required"/>
  </xs:complexType>
</xs:schema>
```

```

</xs:complexType>
<!-- start of the non-functional properties part -->
<xs:complexType name="nonFunctionalPropertiesType">
  <xs:choice minOccurs="0" maxOccurs="unbounded">
    <xs:element name="version"/>
    <xs:any/>
  </xs:choice>
</xs:complexType>
<xs:complexType name="nonFunctionalPropertiesWSType">
  <xs:complexContent>
    <xs:extension base="nonFunctionalPropertiesType">
      <xs:choice minOccurs="0" maxOccurs="unbounded">
        <xs:element name="performance"/>
        <xs:element name="reliability"/>
        <xs:element name="security"/>
        <xs:element name="scalability"/>
        <xs:element name="robustness"/>
        <xs:element name="accuracy"/>
        <xs:element name="transactional"/>
        <xs:element name="trust"/>
        <xs:element name="financial"/>
        <xs:element name="networkRelatedQoS"/>
      </xs:choice>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:complexType name="importOntologyType"/>
<!-- end of the non-functional properties part -->
<!-- start of the Web Services part -->
<xs:complexType name="webServiceType">
  <xs:sequence>
    <xs:element name="nonFunctionalProperties" type="nonFunctionalPropertiesWSType" minOccurs="0"/>
    <xs:element name="importOntology" type="wsmliID" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name="ooMediator" type="wsmliID" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name="capability" type="capabilityType"/>
    <xs:element name="interface" type="interfaceType" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:attribute name="id" type="wsmliID" use="required"/>
</xs:complexType>
<xs:complexType name="capabilityType">
  <xs:sequence>
    <xs:element name="nonFunctionalProperties" type="nonFunctionalPropertiesType" minOccurs="0"/>
    <xs:element name="importOntology" type="wsmliID" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name="ooMediator" type="ooMediatorType" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name="wgMediator" type="wgMediatorType" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name="preCondition" type="axiomType" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name="assumption" type="axiomType" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name="postCondition" type="axiomType" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name="effect" type="axiomType" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:attribute name="id" type="wsmliID" use="required"/>
</xs:complexType>
<xs:complexType name="interfaceType">
  <xs:sequence>
    <xs:element name="nonFunctionalProperties" type="nonFunctionalPropertiesType" minOccurs="0"/>
    <xs:element name="importOntology" type="wsmliID" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name="ooMediator" type="wsmliID" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name="choreography" type="wsmliID"/>
    <xs:element name="orchestration" type="wsmliID"/>
  </xs:sequence>
  <xs:attribute name="id" type="wsmliID" use="required"/>
</xs:complexType>
<!-- End of the Web Services part -->
<!-- start of the mediator part -->
<xs:complexType name="mediatorType">
  <xs:choice>
    <xs:element name="ooMediator" type="ooMediatorType"/>
    <xs:element name="ggMediator" type="ggMediatorType"/>
    <xs:element name="wgMediator" type="wgMediatorType"/>
    <xs:element name="wwMediator" type="wwMediatorType"/>
  </xs:choice>
</xs:complexType>
<xs:complexType name="ooMediatorType">
  <xs:sequence>
    <xs:element name="nonFunctionalProperties" type="nonFunctionalPropertiesWSType" minOccurs="0"/>
    <xs:element name="importOntology" type="wsmliID" minOccurs="0" maxOccurs="unbounded"/>
    <xs:choice minOccurs="0" maxOccurs="unbounded">
      <xs:element name="sourceOntology" type="wsmliID"/>
      <xs:element name="sourceooMediator" type="wsmliID"/>
    </xs:choice>
    <xs:choice minOccurs="0">
      <xs:element name="targetOntology" type="wsmliID"/>
      <xs:element name="targetGoal" type="wsmliID"/>
      <xs:element name="targetWebService" type="wsmliID"/>
      <xs:element name="targetMediator" type="wsmliID"/>
    </xs:choice>
    <xs:choice minOccurs="0">
      <xs:element name="goal" type="wsmliID"/>
      <xs:element name="wwMediator" type="wsmliID"/>
      <xs:element name="webService" type="wsmliID"/>
    </xs:choice>
  </xs:sequence>
  <xs:attribute name="id" type="wsmliID" use="required"/>
</xs:complexType>
<xs:complexType name="ggMediatorType">
  <xs:sequence>
    <xs:element name="nonFunctionalProperties" type="nonFunctionalPropertiesWSType" minOccurs="0"/>
    <xs:element name="importOntology" type="wsmliID" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name="usedMediator" type="wsmliID" minOccurs="0" maxOccurs="unbounded"/>
    <xs:choice minOccurs="0" maxOccurs="unbounded">
      <xs:element name="sourceGoal" type="wsmliID"/>
    </xs:choice>
  </xs:sequence>

```

```

        <xs:element name="sourceeggMediator" type="wsmlID"/>
    </xs:choice>
    <xs:choice minOccurs="0">
        <xs:element name="targetGoal" type="wsmlID"/>
        <xs:element name="targetggMediator" type="wsmlID"/>
    </xs:choice>
    <xs:choice minOccurs="0">
        <xs:element name="goal" type="wsmlID"/>
        <xs:element name="wwMediator" type="wsmlID"/>
        <xs:element name="webService" type="wsmlID"/>
    </xs:choice>
</xs:sequence>
<xs:attribute name="id" type="wsmlID" use="required"/>
</xs:complexType>
<xs:complexType name="wgMediatorType">
    <xs:sequence>
        <xs:element name="nonFunctionalProperties" type="nonFunctionalPropertiesWSType" minOccurs="0"/>
        <xs:element name="importOntology" type="wsmlID" minOccurs="0" maxOccurs="unbounded"/>
        <xs:element name="usedMediator" type="wsmlID" minOccurs="0" maxOccurs="unbounded"/>
        <xs:choice minOccurs="0">
            <xs:element name="sourceWebService" type="wsmlID"/>
            <xs:element name="sourcewgMediator" type="wsmlID"/>
        </xs:choice>
        <xs:choice minOccurs="0">
            <xs:element name="targetGoal" type="wsmlID"/>
            <xs:element name="targetggMediator" type="wsmlID"/>
        </xs:choice>
        <xs:choice minOccurs="0">
            <xs:element name="goal" type="wsmlID"/>
            <xs:element name="wwMediator" type="wsmlID"/>
            <xs:element name="webService" type="wsmlID"/>
        </xs:choice>
    </xs:sequence>
    <xs:attribute name="id" type="wsmlID" use="required"/>
</xs:complexType>
<xs:complexType name="wwMediatorType">
    <xs:sequence>
        <xs:element name="nonFunctionalProperties" type="nonFunctionalPropertiesWSType" minOccurs="0"/>
        <xs:element name="importOntology" type="wsmlID" minOccurs="0" maxOccurs="unbounded"/>
        <xs:element name="usedMediator" type="wsmlID" minOccurs="0" maxOccurs="unbounded"/>
        <xs:choice minOccurs="0">
            <xs:element name="sourceWebService" type="webServiceType"/>
            <xs:element name="sourcewwMediator" type="wwMediatorType"/>
        </xs:choice>
        <xs:choice minOccurs="0">
            <xs:element name="targetWebService" type="webServiceType"/>
            <xs:element name="targetwwMediator" type="wwMediatorType"/>
        </xs:choice>
        <xs:choice minOccurs="0">
            <xs:element name="goal" type="wsmlID"/>
            <xs:element name="wwMediator" type="wsmlID"/>
            <xs:element name="webService" type="wsmlID"/>
        </xs:choice>
    </xs:sequence>
    <xs:attribute name="id" type="wsmlID" use="required"/>
</xs:complexType>
<!-- end of the mediator part -->
<!-- start of the ontology part -->
<xs:complexType name="ontologyType">
    <xs:sequence>
        <xs:element name="nonFunctionalProperties" type="nonFunctionalPropertiesType" minOccurs="0"/>
        <xs:element name="importOntology" type="wsmlID" minOccurs="0" maxOccurs="unbounded"/>
        <xs:element name="ooMediator" type="wsmlID" minOccurs="0" maxOccurs="unbounded"/>
        <xs:element name="concept" type="conceptType" minOccurs="0" maxOccurs="unbounded"/>
        <xs:element name="relation" type="relationType" minOccurs="0" maxOccurs="unbounded"/>
        <xs:element name="instance" type="instanceType" minOccurs="0" maxOccurs="unbounded"/>
        <xs:element name="function" type="functionType" minOccurs="0" maxOccurs="unbounded"/>
        <xs:element name="axiom" type="axiomType" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:attribute name="id" type="wsmlID" use="required"/>
</xs:complexType>
<!-- axiomType refers to flogicType defined in the F-Logic/XML syntax; WSMO D16.2 -->
<!-- some validators might break on this, because they do not have this schema available -->
<!-- axiomType is used as a basis for extension by many other types used in the ontology part -->
<xs:complexType name="axiomType">
    <xs:sequence>
        <xs:element name="nonFunctionalProperties" type="nonFunctionalPropertiesType" minOccurs="0"/>
        <xs:element name="definedBy" type="logicalExpressionType"/>
    </xs:sequence>
    <xs:attribute name="id" type="wsmlID" use="required"/>
</xs:complexType>
<xs:complexType name="conceptType">
    <xs:sequence>
        <xs:element name="nonFunctionalProperties" type="nonFunctionalPropertiesType" minOccurs="0"/>
        <xs:element name="superConcept" type="wsmlID" minOccurs="0" maxOccurs="unbounded"/>
        <xs:element name="attribute" type="attributeType" minOccurs="0" maxOccurs="unbounded"/>
        <xs:element name="definedBy" type="logicalExpressionType" minOccurs="0"/>
    </xs:sequence>
    <xs:attribute name="id" type="wsmlID" use="required"/>
</xs:complexType>
<xs:complexType name="relationType">
    <xs:sequence>
        <xs:element name="nonFunctionalProperties" type="nonFunctionalPropertiesType" minOccurs="0"/>
        <xs:element name="superRelation" type="wsmlID" minOccurs="0" maxOccurs="unbounded"/>
        <xs:element name="parameter" type="parameterType" minOccurs="0" maxOccurs="unbounded"/>
        <xs:element name="definedBy" type="logicalExpressionType" minOccurs="0"/>
    </xs:sequence>
    <xs:attribute name="id" type="wsmlID" use="required"/>
</xs:complexType>
<xs:complexType name="parameterType">
    <xs:sequence>

```

```

        <xs:element name="type" type="wsmlID"/>
    </xs:sequence>
    <xs:attribute name="id" type="wsmlID" use="required"/>
</xs:complexType>
<xs:complexType name="instanceType">
    <xs:sequence>
        <xs:element name="nonFunctionalProperties" type="nonFunctionalPropertiesType" minOccurs="0"/>
        <xs:element name="memberOf" type="wsmlID" minOccurs="0" maxOccurs="unbounded"/>
        <xs:element name="attributeValue" minOccurs="0" maxOccurs="unbounded">
            <xs:complexType>
                <xs:sequence>
                    <xs:element name="nonFunctionalProperties" type="nonFunctionalPropertiesType"
                        minOccurs="0"/>
                    <xs:element name="attributeID" type="wsmlID"/>
                    <xs:element name="value" type="wsmlIDorLiteral" maxOccurs="unbounded"/>
                <!-- can be instance, ID or literal -->
                </xs:sequence>
            </xs:complexType>
        </xs:element>
    </xs:sequence>
    <xs:attribute name="id" type="wsmlID" use="required"/>
</xs:complexType>
<xs:complexType name="relationInstanceType">
    <xs:sequence>
        <xs:element name="nonFunctionalProperties" type="nonFunctionalPropertiesType" minOccurs="0"/>
        <xs:element name="memberOf" type="wsmlID" minOccurs="0" maxOccurs="unbounded"/>
        <xs:element name="parameterValue" minOccurs="0" maxOccurs="unbounded">
            <xs:complexType>
                <xs:sequence>
                    <xs:element name="parameter" type="wsmlID"/>
                    <xs:element name="value" type="wsmlID"/>
                </xs:sequence>
            </xs:complexType>
        </xs:element>
    </xs:sequence>
    <xs:attribute name="id" type="wsmlID" use="required"/>
</xs:complexType>
<xs:complexType name="attributeType">
    <xs:sequence>
        <xs:element name="nonFunctionalProperties" type="nonFunctionalPropertiesType" minOccurs="0"/>
        <xs:element name="range" type="wsmlID"/>
    </xs:sequence>
    <xs:attribute name="id" type="wsmlID" use="required"/>
</xs:complexType>
<xs:complexType name="functionType">
    <xs:complexContent>
        <xs:extension base="relationType">
            <xs:sequence>
                <xs:element name="range" type="wsmlID"/>
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
<!-- End of the ontology part -->
<!-- auxiliary type definitions -->
<xs:simpleType name="wsmlID">
    <xs:union memberTypes="xs:anyURI xs:QName"/>
</xs:simpleType>
<xs:simpleType name="wsmlIDorLiteral-basic">
    <xs:union memberTypes="wsmlID xs:string"/>
</xs:simpleType>
<xs:complexType name="wsmlLiteral">
    <xs:simpleContent>
        <xs:extension base="xs:string">
            <xs:attribute name="type" type="wsmlID"/>
        </xs:extension>
    </xs:simpleContent>
</xs:complexType>
<xs:complexType name="wsmlIDorLiteral">
    <xs:simpleContent>
        <xs:extension base="wsmlIDorLiteral-basic">
            <xs:attribute name="type" type="wsmlID" use="optional"/>
        </xs:extension>
    </xs:simpleContent>
</xs:complexType>
<xs:complexType name="wsmlAnyID">
    <xs:simpleContent>
        <xs:extension base="wsmlIDorLiteral-basic">
            <xs:attribute name="type" type="wsmlID" use="optional"/>
            <xs:attribute name="kind" use="required">
                <xs:simpleType>
                    <xs:restriction base="xs:string">
                        <xs:enumeration value="variable"/>
                        <xs:enumeration value="wsmlID"/>
                        <xs:enumeration value="literal"/>
                        <xs:enumeration value="anonymousID"/>
                    </xs:restriction>
                </xs:simpleType>
            </xs:attribute>
        </xs:extension>
    </xs:simpleContent>
</xs:complexType>
<xs:simpleType name="wsmlVariable">
    <xs:restriction base="xs:string"/>
</xs:simpleType>
<!-- end of auxiliary type definitions -->
<!-- Start of logical expressions (umbrella syntax) -->
<xs:complexType name="logicalExpressionType">
    <xs:choice>
        <xs:element name="disjunction" type="disjunctionType"/>
        <xs:element name="constraint" type="disjunctionType"/>
    </xs:choice>

```

```

        <xs:element name="implication" type="implicationType"/>
    </xs:choice>
</xs:complexType>
<xs:complexType name="disjunctionType">
    <xs:sequence>
        <xs:element name="quantification" type="quantificationType" minOccurs="0" maxOccurs="unbounded"/>
        <xs:choice maxOccurs="unbounded">
            <xs:element name="conjunction" type="conjunctionType"/>
        </xs:choice>
    </xs:sequence>
    <xs:attribute name="negated" type="xs:boolean" default="false"/>
</xs:complexType>
<xs:complexType name="conjunctionType">
    <xs:choice maxOccurs="unbounded">
        <xs:element name="disjunction" type="disjunctionType"/>
        <xs:element name="molecule" type="moleculeType"/>
        <xs:element name="predicate" type="predicateType"/>
    </xs:choice>
    <xs:attribute name="negated" type="xs:boolean" default="false"/>
</xs:complexType>
<xs:complexType name="implicationType">
    <xs:sequence>
        <xs:element name="quantification" type="quantificationType" minOccurs="0" maxOccurs="unbounded"/>
        <xs:element name="leftHandSide" type="disjunctionType" minOccurs="0"/>
        <xs:element name="rightHandSide" type="disjunctionType" minOccurs="0"/>
    </xs:sequence>
    <xs:attribute name="type" use="required">
        <xs:simpleType>
            <xs:restriction base="xs:string">
                <xs:enumeration value="left"/>
                <xs:enumeration value="right"/>
                <xs:enumeration value="dual"/>
            </xs:restriction>
        </xs:simpleType>
    </xs:attribute>
    <xs:attribute name="negated" type="xs:boolean" default="false"/>
</xs:complexType>
<xs:complexType name="moleculeType">
    <xs:sequence>
        <xs:element name="id" type="termType"/>
        <xs:element name="attributeDefinition" minOccurs="0" maxOccurs="unbounded">
            <xs:complexType>
                <xs:sequence>
                    <xs:element name="attributeID" type="termType"/>
                    <xs:element name="type" type="termType"/>
                </xs:sequence>
                <xs:attribute name="type">
                    <xs:simpleType>
                        <xs:restriction base="xs:string">
                            <xs:enumeration value="singleValued"/>
                            <xs:enumeration value="setValued"/>
                        </xs:restriction>
                    </xs:simpleType>
                </xs:attribute>
            </xs:complexType>
        </xs:element>
        <xs:element name="attributeValueDefinition" minOccurs="0" maxOccurs="unbounded">
            <xs:complexType>
                <xs:sequence>
                    <xs:element name="attributeID" type="termType"/>
                    <xs:element name="value" type="termType"/>
                </xs:sequence>
                <xs:attribute name="type">
                    <xs:simpleType>
                        <xs:restriction base="xs:string">
                            <xs:enumeration value="singleValued"/>
                            <xs:enumeration value="setValued"/>
                        </xs:restriction>
                    </xs:simpleType>
                </xs:attribute>
            </xs:complexType>
        </xs:element>
    </xs:sequence>
    <xs:element name="isa" minOccurs="0">
        <xs:complexType>
            <xs:simpleContent>
                <xs:extension base="xs:string">
                    <xs:attribute name="type">
                        <xs:simpleType>
                            <xs:restriction base="xs:string">
                                <xs:enumeration value="memberOf"/>
                                <xs:enumeration value="subConceptOf"/>
                            </xs:restriction>
                        </xs:simpleType>
                    </xs:attribute>
                </xs:extension>
            </xs:simpleContent>
        </xs:complexType>
    </xs:element>
</xs:sequence>
    <xs:attribute name="negated" type="xs:boolean" default="false"/>
</xs:complexType>
<xs:complexType name="predicateType">
    <xs:sequence>
        <xs:element name="id" type="termType"/>
        <xs:element name="argument" type="termType" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:attribute name="negated" type="xs:boolean" default="false"/>
</xs:complexType>
<xs:complexType name="termType" mixed="true">
    <xs:complexContent>
        <xs:extension base="wsm:AnyID">

```

```

        <xs:sequence>
        <xs:element name="argument" type="wsmlAnyID" minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
    </xs:extension>
</xs:complexContent>
</xs:complexType>
<xs:complexType name="quantificationType">
    <xs:simpleContent>
        <xs:extension base="wsmlVariable">
            <xs:attribute name="type">
                <xs:simpleType>
                    <xs:restriction base="xs:string">
                        <xs:enumeration value="universal"/>
                        <xs:enumeration value="existential"/>
                    </xs:restriction>
                </xs:simpleType>
            </xs:attribute>
        </xs:extension>
    </xs:simpleContent>
</xs:complexType>
<!-- End of logical expressions (umbrella syntax) -->
</xs:schema>

```

Appendix B. An example of a WSMML/XML ontology

```

<?xml version="1.0" encoding="UTF-8"?>
<wsml:wsml xmlns:wsml="http://www.wsmo.org/2004/d16/16.3/" xmlns:exp="http://www.wsmo.org/2004/d16/d16.3"
xmlns="http://www.wsmo.org/ontologies/dateTime#" xmlns:dc="http://purl.org/dc/elements/1.1#"
xmlns:xsd="http://www.w3.org/2001/XMLSchema#">
  <wsml:ontology id="http://www.example.org/example.wsml">
    <!-- this example WSMML/XML ontology is based on the example given with the WSMML validator -->

    <!-- Notice that the ontology is rather complex when it comes to logical expressions;
    however, XML is just an exchange syntax -->
    <wsml:nonFunctionalProperties>
      <dc:title>WSMML example collection</dc:title>
      <dc:subject>family</dc:subject>
      <dc:description>Fragments of a family ontology to provide WSMML examples</dc:description>
      <dc:date>2004-06-28</dc:date>
      <dc:type>&lt;http://www.wsmo.org/2004/d2/v0.3/20040329/#ontos&gt;</dc:type>
      <dc:format>text/plain</dc:format>
      <dc:language>en-US</dc:language>
      <dc:rights>&lt;http://www.deri.org/privacy.html&gt;</dc:rights>
      <wsml:version>$Revision: 1.2 $</wsml:version>
    </wsml:nonFunctionalProperties>
    <wsml:concept id="Human">
      <wsml:nonFunctionalProperties>
        <dc:description>concept of a human being</dc:description>
      </wsml:nonFunctionalProperties>
      <wsml:attribute id="name">
        <wsml:range>xsd:string</wsml:range>
      </wsml:attribute>
      <wsml:attribute id="age">
        <wsml:range>xsd:string</wsml:range>
      </wsml:attribute>
    </wsml:concept>
    <wsml:concept id="Man">
      <wsml:nonFunctionalProperties>
        <dc:description>concept of a man</dc:description>
      </wsml:nonFunctionalProperties>
      <wsml:superConcept>Human</wsml:superConcept>
    </wsml:concept>
    <wsml:concept id="Woman">
      <wsml:nonFunctionalProperties>
        <dc:description>concept of a woman</dc:description>
      </wsml:nonFunctionalProperties>
      <wsml:superConcept>Human</wsml:superConcept>
      <wsml:definedBy>
        <wsml:implication type="right">
          <wsml:quantification type="universal">?x</wsml:quantification>
          <wsml:leftHandSide>
            <wsml:conjunction>
              <wsml:molecule>
                <wsml:id kind="variable">?x</wsml:id>
                <wsml:isa type="memberOf">Woman</wsml:isa>
              </wsml:molecule>
            </wsml:conjunction>
          </wsml:leftHandSide>
          <wsml:rightHandSide>
            <wsml:conjunction negated="true">
              <wsml:molecule>
                <wsml:id kind="variable">?x</wsml:id>
                <wsml:isa type="memberOf">Man</wsml:isa>
              </wsml:molecule>
            </wsml:conjunction>
          </wsml:rightHandSide>
        </wsml:implication>
      </wsml:definedBy>
    </wsml:concept>
    <wsml:concept id="Boy">
      <wsml:nonFunctionalProperties>
        <dc:description>concept of a man not older than 14</dc:description>
      </wsml:nonFunctionalProperties>
      <wsml:superConcept>Man</wsml:superConcept>
      <wsml:definedBy>
        <wsml:implication type="dual">
          <wsml:quantification type="universal">?x</wsml:quantification>
          <wsml:leftHandSide>

```

```

        <wsmml:conjunction>
          <wsmml:molecule>
            <wsmml:id kind="variable"?x</wsmml:id>
            <wsmml:isa type="memberOf">Boy</wsmml:isa>
          </wsmml:molecule>
        </wsmml:conjunction>
      </wsmml:leftHandSide>
      <wsmml:rightHandSide>
        <wsmml:conjunction>
          <wsmml:predicate>
            <wsmml:id kind="wsmmlID">xquery:numeric-less-than</wsmml:id>
            <wsmml:argument kind="variable"?x.age</wsmml:argument>
            <wsmml:argument kind="literal" type="xsd:integer">14</wsmml:argument>
          </wsmml:predicate>
          <wsmml:molecule>
            <wsmml:id kind="variable"?x</wsmml:id>
            <wsmml:isa type="memberOf">Man</wsmml:isa>
          </wsmml:molecule>
        </wsmml:conjunction>
      </wsmml:rightHandSide>
    </wsmml:implication>
  </wsmml:definedBy>
</wsmml:concept>
<wsmml:concept id="Parent">
  <wsmml:nonFunctionalProperties>
    <dc:description>concept of a man not older than 14</dc:description>
  </wsmml:nonFunctionalProperties>
  <wsmml:superConcept>Human</wsmml:superConcept>
  <wsmml:attribute id="children">
    <wsmml:range>Human</wsmml:range>
  </wsmml:attribute>
  <wsmml:definedBy>
    <wsmml:implication type="right">
      <wsmml:quantification type="universal">x</wsmml:quantification>
      <wsmml:leftHandSide>
        <wsmml:conjunction>
          <wsmml:molecule>
            <wsmml:id kind="variable"?x</wsmml:id>
            <wsmml:isa type="memberOf">Parent</wsmml:isa>
          </wsmml:molecule>
        </wsmml:conjunction>
      </wsmml:leftHandSide>
      <wsmml:rightHandSide>
        <wsmml:conjunction>
          <wsmml:molecule>
            <wsmml:id kind="variable"?x</wsmml:id>
            <wsmml:isa type="memberOf">Human</wsmml:isa>
          </wsmml:molecule>
          <wsmml:disjunction>
            <wsmml:quantification type="existential">y</wsmml:quantification>
            <wsmml:conjunction>
              <wsmml:molecule>
                <wsmml:id kind="variable"?y</wsmml:id>
                <wsmml:attributeValueDefinition>
                  <wsmml:attributeID kind="wsmmlID">parent</wsmml:attributeID>
                  <wsmml:value kind="variable"?x</wsmml:value>
                </wsmml:attributeValueDefinition>
                <wsmml:attributeValueDefinition>
                  <wsmml:attributeID kind="wsmmlID">child</wsmml:attributeID>
                  <wsmml:value kind="variable"?y</wsmml:value>
                </wsmml:attributeValueDefinition>
              </wsmml:molecule>
            </wsmml:conjunction>
          </wsmml:disjunction>
        </wsmml:conjunction>
      </wsmml:rightHandSide>
    </wsmml:implication>
  </wsmml:definedBy>
</wsmml:concept>
<wsmml:relation id="hasChild">
  <wsmml:parameter id="parent">
    <wsmml:type>Human</wsmml:type>
  </wsmml:parameter>
  <wsmml:parameter id="child">
    <wsmml:type>Human</wsmml:type>
  </wsmml:parameter>
</wsmml:relation>
<wsmml:relation id="childOf">
  <wsmml:parameter id="child">
    <wsmml:type>Human</wsmml:type>
  </wsmml:parameter>
  <wsmml:parameter id="parent">
    <wsmml:type>Human</wsmml:type>
  </wsmml:parameter>
  <wsmml:definedBy>
    <wsmml:implication type="dual">
      <wsmml:quantification type="universal">x</wsmml:quantification>
      <wsmml:quantification type="universal">y</wsmml:quantification>
      <wsmml:leftHandSide>
        <wsmml:conjunction>
          <wsmml:molecule>
            <wsmml:id kind="wsmmlID">child</wsmml:id>
            <wsmml:attributeValueDefinition>
              <wsmml:attributeID kind="wsmmlID">child</wsmml:attributeID>
              <wsmml:value kind="variable"?x</wsmml:value>
            </wsmml:attributeValueDefinition>
          </wsmml:molecule>
        </wsmml:conjunction>
      </wsmml:leftHandSide>
    </wsmml:implication>
  </wsmml:definedBy>
</wsmml:relation>

```



```

        <wsml:attributeValueDefinition>
          <wsml:attributeID kind="wsmlID">parent</wsml:attributeID>
          <wsml:value kind="variable">?y</wsml:value>
        </wsml:attributeValueDefinition>
      </wsml:molecule>
    </wsml:conjunction>
  </wsml:leftHandSide>
  <wsml:rightHandSide>
    <wsml:conjunction>
      <wsml:molecule>
        <wsml:id kind="wsmlID">hasChild</wsml:id>
        <wsml:attributeValueDefinition>
          <wsml:attributeID kind="wsmlID">parent</wsml:attributeID>
          <wsml:value kind="variable">?y</wsml:value>
        </wsml:attributeValueDefinition>
        <wsml:attributeValueDefinition>
          <wsml:attributeID kind="wsmlID">child</wsml:attributeID>
          <wsml:value kind="variable">?x</wsml:value>
        </wsml:attributeValueDefinition>
      </wsml:molecule>
    </wsml:conjunction>
  </wsml:rightHandSide>
</wsml:implication>
</wsml:definedBy>
</wsml:relation>
<wsml:relation id="hasSon">
  <wsml:superRelation>hasChild</wsml:superRelation>
  <wsml:parameter id="parent">
    <wsml:type>Human</wsml:type>
  </wsml:parameter>
  <wsml:parameter id="child">
    <wsml:type>Man</wsml:type>
  </wsml:parameter>
</wsml:relation>
<wsml:instance id="Mary">
  <wsml:memberOf>Parent</wsml:memberOf>
  <wsml:attributeValue>
    <wsml:attributeID>name</wsml:attributeID>
    <wsml:value type="xsd:string">Maria Smith</wsml:value>
  </wsml:attributeValue>
  <wsml:attributeValue>
    <wsml:attributeID>age</wsml:attributeID>
    <wsml:value type="xsd:integer">50</wsml:value>
  </wsml:attributeValue>
  <wsml:attributeValue>
    <wsml:attributeID>children</wsml:attributeID>
    <wsml:value>Michael</wsml:value>
    <wsml:value>Susan</wsml:value>
  </wsml:attributeValue>
</wsml:instance>
<wsml:instance id="Paul">
  <wsml:memberOf>Man</wsml:memberOf>
  <wsml:attributeValue>
    <wsml:attributeID>name</wsml:attributeID>
    <wsml:value type="xsd:string">Paul Jones</wsml:value>
  </wsml:attributeValue>
  <wsml:attributeValue>
    <wsml:attributeID>age</wsml:attributeID>
    <wsml:value type="xsd:integer">25</wsml:value>
  </wsml:attributeValue>
</wsml:instance>
<wsml:function id="hasAge">
  <wsml:parameter id="person">
    <wsml:type>Human</wsml:type>
  </wsml:parameter>
  <wsml:definedBy>
    <wsml:implication type="dual">
      <wsml:quantification type="universal">?x</wsml:quantification>
      <wsml:quantification type="universal">?y</wsml:quantification>
      <wsml:leftHandSide>
        <wsml:conjunction>
          <wsml:molecule>
            <wsml:id kind="wsmlID">hasAge</wsml:id>
            <wsml:attributeValueDefinition>
              <wsml:attributeID kind="wsmlID">person</wsml:attributeID>
              <wsml:value kind="variable">?x</wsml:value>
            </wsml:attributeValueDefinition>
            <wsml:attributeValueDefinition>
              <wsml:attributeID kind="wsmlID">result</wsml:attributeID>
              <wsml:value kind="variable">?y</wsml:value>
            </wsml:attributeValueDefinition>
          </wsml:molecule>
        </wsml:conjunction>
      </wsml:leftHandSide>
      <wsml:rightHandSide>
        <wsml:conjunction>
          <wsml:predicate>
            <wsml:id kind="wsmlID">xquery:numeric-equal</wsml:id>
            <wsml:argument kind="variable">?x.age</wsml:argument>
            <wsml:argument kind="variable">?y</wsml:argument>
          </wsml:predicate>
        </wsml:conjunction>
      </wsml:rightHandSide>
    </wsml:implication>
  </wsml:definedBy>
  <wsml:range>xsd:integer</wsml:range>
</wsml:function>
<wsml:axiom id="disjointManWoman">
  <wsml:nonFunctionalProperties>
    <dc:description>Man and Woman are two disjoint concepts
    without constraint notation. Notice that this is modeled here as an integrity constraint,

```

```
rather than an implication as was done in the example</dc:description>
</wsml:nonFunctionalProperties>
<wsml:definedBy>
  <wsml:constraint>
    <wsml:conjunction>
      <wsml:molecule>
        <wsml:id kind="variable"?x</wsml:id>
        <wsml:isa type="memberOf">Man</wsml:isa>
      </wsml:molecule>
      <wsml:molecule>
        <wsml:id kind="variable"?x</wsml:id>
        <wsml:isa type="memberOf">Woman</wsml:isa>
      </wsml:molecule>
    </wsml:conjunction>
  </wsml:constraint>
</wsml:definedBy>
</wsml:axiom>
</wsml:ontology>
</wsml:wsml>
```

Acknowledgements

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

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

\$Date: 2004/09/09 16:24:32 \$

[webmaster](#)