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

WSMO Working Draft 22 March 2004

This version:
http://www.wsmo.org/2004/d16/d16.3/v0.1/20040322/
Latest version:
http://www.wsmo.org/2004/d16/d16.3/
Editors:
Jos de Bruijn
Authors:
Jos de Bruijn
Michael Kifer

Abstract

This document defines WSML/XML - an XML syntax for WSML, based on the WSML/BNF syntax. Furthermore, it provides an XSLT transformation script to translate the WSML/XML syntax to WSML/BNF.

Table of contents

1. Introduction

This document provides an XML syntax for the Web Service Modeling Language WSML, based on the BNF syntax provided by [Oren, 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 [Oren, 2004].

In order to encode logical expressions, the XML Schema presented in this document imports the XML Schema for the F-Logic/XML syntax presented in [de Bruijn & Kifer, 2004].

2. XML Syntax

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

3. XSLT Stylesheet for Converting F-logic/XML to the Regular F-logic

Syntax

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

4. References

[de Bruijn & Kifer, 2004] J de Bruijn and M. Kifer (editors) (2004). *F-logic/XML - An XML Syntax for F-logic.* WSMO Working Draft v0.1, Digital Enterprise Research Insitute (DERI), available from http://www.wsmo.org/2004/d16/d16.2/v0.1/.

[Oren, 2004] E. Oren (editor) (2004). *BNF Syntax.* WSMO Working Draft v0.1, Digital Enterprise Research Insitute (DERI), available from http://www.wsmo.org/2004/d16/d16.1/.

Appendix A. XML Schema for WSML/XML

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema targetNamespace="http://www.wsmo.org/2004/d163/"
        xmlns:xs="http://www.w3.org/2001/XMLSchema"
        xmlns:flg="http://www.wsmo.org/2004/d162/"
        xmlns="http://www.wsmo.org/2004/d163/"
        elementFormDefault="qualified" attributeFormDefault="unqualified">
        <!-- version: 2004-03-18 -->
        <!-- author: Jos de Bruijn -->
        <!-- this document provides an XML Schema for the XML version of the WSML syntax. -->
        <!-- this document is based on the BNF syntax for WSML (WSMO D16.1) v0.1 dated 2004-03-15 -->
        <xs:import namespace="http://www.wsmo.org/2004/d162/"
                schemaLocation="http://www.wsmo.org/2004/d16/d16.2/flogic.xml"/>
        <xs:element name="wsml">
                <xs:annotation>
                        <xs:documentation>the root element, containing the WSML document</xs:documentation>
                </xs:annotation>
                <xs:complexType>
                        <xs:sequence>
                                <xs:element name="ontology" type="ontologyType" minOccurs="0" maxOccurs="unbounded"/>
                                <xs:element name="webService" type="webServiceType" minOccurs="0"
                                        maxOccurs="unbounded"/>
                                <xs:element name="goal" type="goalType" minOccurs="0" maxOccurs="unbounded"/>
                                <xs:element name="mediator" type="mediatorType" minOccurs="0" maxOccurs="unbounded"/>
                        </xs:sequence>
                </xs:complexType>
        </xs:element>
        <xs:complexType name="goalType">
                <xs:sequence>
                        <xs:element name="nonFunctionalProperties" type="nonFunctionalPropertiesType" minOccurs="0"/>
                        <xs:element name="ooMediator" type="ooMediatorType" minOccurs="0" maxOccurs="unbounded"/>
                        <xs:element name="ggMediator" type="ggMediatorType" 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="xs:QName" use="required"/>
        </xs:complexType>
        <!-- start of the non-functional properties part -->
        <xs:complexType name="nonFunctionalPropertiesType">
                <xs:sequence>
                        <xs:element name="title" type="xs:string" minOccurs="0"/>
                        <xs:element name="creator" type="xs:QName" minOccurs="0"/>
                        <xs:element name="subject" type="xs:QName" minOccurs="0" maxOccurs="unbounded"/>
                        <xs:element name="description" type="xs:QName" minOccurs="0"/>
                        <xs:element name="publisher" type="xs:QName" minOccurs="0"/>
                        <xs:element name="contributor" type="xs:QName" minOccurs="0"/>
                        <xs:element name="date" type="xs:date" minOccurs="0"/>
                        <xs:element name="type" type="xs:QName" minOccurs="0"/>
                        <xs:element name="format" type="xs:QName" minOccurs="0"/>
                        <xs:element name="source" type="xs:QName" minOccurs="0"/>
                        <xs:element name="language" type="xs:QName" minOccurs="0"/>
                        <xs:element name="relation" type="xs:QName" minOccurs="0"/>
                        <xs:element name="coverage" type="xs:QName" minOccurs="0"/>
                        <xs:element name="rights" type="xs:QName" minOccurs="0"/>
                        <xs:element name="version" type="xs:QName" minOccurs="0"/>
                </xs:sequence>
        </xs:complexType>
        <xs:complexType name="nonFunctionalPropertiesWSType">
```

```
        <xs:complexContent>
                <xs:extension base="nonFunctionalPropertiesType">
                        <xs:sequence>
                                <xs:element name="performance" type="xs:QName" minOccurs="0"/>
                                <xs:element name="reliability" type="xs:QName" minOccurs="0"/>
                                <xs:element name="security" type="xs:QName" minOccurs="0"/>
                                <xs:element name="scalability" type="xs:QName" minOccurs="0"/>
                                <xs:element name="robustness" type="xs:QName" minOccurs="0"/>
                                <xs:element name="accuracy" type="xs:QName" minOccurs="0"/>
                                <xs:element name="transactional" type="xs:QName" minOccurs="0"/>
                                <xs:element name="trust" type="xs:QName" minOccurs="0"/>
                                <xs:element name="financial" type="xs:QName" minOccurs="0"/>
                                <xs:element name="networkRelatedQoS" type="xs:QName" minOccurs="0"/>
                        </xs:sequence>
                </xs:extension>
        </xs:complexContent>
</xs:complexType>
<!-- 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="ooMediator" type="ooMediatorType" minOccurs="0" maxOccurs="unbounded"/>
                <xs:element name="capability" type="capabilityType" minOccurs="0"/>
                <xs:element name="interface" type="interfaceType" minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
        <xs:attribute name="id" type="xs:QName" use="required"/>
</xs:complexType>
<xs:complexType name="capabilityType">
        <xs:sequence>
                <xs:element name="nonFunctionalProperties" type="nonFunctionalPropertiesType" minOccurs="0"/>
                <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="xs:QName" use="required"/>
</xs:complexType>
<xs:complexType name="interfaceType">
        <xs:sequence>
                <xs:element name="nonFunctionalProperties" type="nonFunctionalPropertiesType" minOccurs="0"/>
                <xs:element name="ooMediator" type="ooMediatorType" minOccurs="0" maxOccurs="unbounded"/>
                <xs:element name="choreography" type="xs:QName"/>
                <xs:element name="orchestration" type="xs:QName"/>
        </xs:sequence>
        <xs:attribute name="id" type="xs:QName" 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="sourceOntology" type="ontologyType" minOccurs="0" maxOccurs="unbounded"/>
                <xs:element name="sourceooMediator" type="ooMediatorType" minOccurs="0"
                        maxOccurs="unbounded"/>
                <xs:element name="targetOntology" type="ontologyType" minOccurs="0" maxOccurs="unbounded"/>
                <xs:element name="targetGoal" type="goalType" minOccurs="0" maxOccurs="unbounded"/>
                <xs:element name="targetWebService" minOccurs="0" maxOccurs="unbounded"/>
                <xs:element name="targetMediator" type="mediatorType" minOccurs="0" maxOccurs="unbounded"/>
                <xs:choice>
                        <xs:element name="goal" type="goalType"/>
                        <xs:element name="wwMediator" type="wwMediatorType"/>
                </xs:choice>
        </xs:sequence>
        <xs:attribute name="id" type="xs:QName" use="required"/>
</xs:complexType>
<xs:complexType name="ggMediatorType">
        <xs:sequence>
                <xs:element name="nonFunctionalProperties" type="nonFunctionalPropertiesWSType"
```

```
                        minOccurs="0"/>
                <xs:choice>
                        <xs:element name="sourceGoal" type="goalType"/>
                        <xs:element name="sourceggMediator" type="ggMediatorType"/>
                </xs:choice>
                <xs:choice>
                        <xs:element name="targetGoal" type="goalType"/>
                        <xs:element name="targetggMediator" type="ggMediatorType"/>
                </xs:choice>
                <xs:element name="ooMediator" type="ooMediatorType" minOccurs="0" maxOccurs="unbounded"/>
                <xs:element name="reduction" type="axiomType" minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
        <xs:attribute name="id" type="xs:QName" use="required"/>
</xs:complexType>
<xs:complexType name="wgMediatorType">
        <xs:sequence>
                <xs:element name="nonFunctionalProperties" type="nonFunctionalPropertiesWSType"
                        minOccurs="0"/>
                <xs:choice>
                        <xs:element name="sourceWebService" type="webServiceType"/>
                        <xs:element name="sourcewgMediator" type="wgMediatorType"/>
                </xs:choice>
                <xs:choice>
                        <xs:element name="targetGoal" type="goalType"/>
                        <xs:element name="targetwgMediator" type="wgMediatorType"/>
                </xs:choice>
                <xs:element name="ooMediator" type="ooMediatorType" minOccurs="0" maxOccurs="unbounded"/>
                <xs:element name="reduction" type="axiomType" minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
        <xs:attribute name="id" type="xs:QName" use="required"/>
</xs:complexType>
<xs:complexType name="wwMediatorType">
        <xs:sequence>
                <xs:element name="nonFunctionalProperties" type="nonFunctionalPropertiesWSType"
                        minOccurs="0"/>
                <xs:choice>
                        <xs:element name="sourceWebService" type="webServiceType"/>
                        <xs:element name="sourcewwMediator" type="wwMediatorType"/>
                </xs:choice>
                <xs:choice>
                        <xs:element name="targetWebService" type="webServiceType"/>
                        <xs:element name="targetwwMediator" type="wwMediatorType"/>
                </xs:choice>
                <xs:element name="ooMediator" type="ooMediatorType" minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
        <xs:attribute name="id" type="xs:QName" 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="ooMediator" type="ooMediatorType" minOccurs="0" maxOccurs="unbounded"/>
                <xs:element name="axiom" type="axiomType" 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:sequence>
        <xs:attribute name="id" type="xs:QName" 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="flogicExpression" type="flg:flogicType"/>
        </xs:sequence>
        <xs:attribute name="id" type="xs:QName" use="required"/>
</xs:complexType>
<xs:complexType name="conceptType">
        <xs:complexContent>
                <xs:extension base="axiomType">
                        <xs:sequence>
                                <xs:element name="superConcept" type="conceptType" minOccurs="0"
                                        maxOccurs="unbounded"/>
                                <xs:element name="attribute" type="attributeType" minOccurs="0"
                                        maxOccurs="unbounded"/>
                                <xs:element name="method" type="methodType" minOccurs="0"
                                        maxOccurs="unbounded"/>
                        </xs:sequence>
```

```
                </xs:extension>
            </xs:complexContent>
        </xs:complexType>
        <xs:complexType name="relationType">
            <xs:complexContent>
                <xs:extension base="axiomType">
                    <xs:sequence>
                        <xs:element name="parameter" type="parameterType" minOccurs="0"
                            maxOccurs="unbounded">
                        </xs:element>
                    </xs:sequence>
                </xs:extension>
            </xs:complexContent>
        </xs:complexType>
        <xs:complexType name="parameterType">
            <xs:complexContent>
                <xs:extension base="axiomType">
                    <xs:sequence>
                        <xs:element name="domain" type="axiomType"/>
                    </xs:sequence>
                </xs:extension>
            </xs:complexContent>
        </xs:complexType>
        <xs:complexType name="instanceType">
            <xs:complexContent>
                <xs:extension base="axiomType">
                    <xs:sequence>
                        <xs:element name="instanceOf" type="conceptType" minOccurs="0"
                            maxOccurs="unbounded"/>
                        <xs:element name="attributeValue" minOccurs="0" maxOccurs="unbounded">
                            <xs:complexType>
                                <xs:complexContent>
                                    <xs:extension base="axiomType">
                                        <xs:sequence>
                                            <xs:element name="value"
                                                type="axiomType"/>
                                        </xs:sequence>
                                    </xs:extension>
                                </xs:complexContent>
                            </xs:complexType>
                        </xs:element>
                    </xs:sequence>
                </xs:extension>
            </xs:complexContent>
        </xs:complexType>
        <xs:complexType name="attributeType">
            <xs:complexContent>
                <xs:extension base="axiomType">
                    <xs:sequence>
                        <xs:element name="range" type="axiomType"/>
                    </xs:sequence>
                </xs:extension>
            </xs:complexContent>
        </xs:complexType>
        <xs:complexType name="methodType">
            <xs:complexContent>
                <xs:extension base="axiomType">
                    <xs:sequence>
                        <xs:element name="range" type="axiomType"/>
                        <xs:element name="parameter" type="parameterType" maxOccurs="unbounded"/>
                    </xs:sequence>
                </xs:extension>
            </xs:complexContent>
        </xs:complexType>
        <!-- End of the ontology part -->
</xs:schema>
```

Appendix B. XSLT for Converting WSML/XML to WSML/BNF

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
        <xsl:output method="text"/>
        <!-- XSLT document for the translation of WSML/XML into the WSML/BNF syntax

                Note that this is the first version of the transformation and does not take tabbing into account
                        when generating the WSML/BNF syntax.
                Furthermore, not all transformations have been tested.

        creator: Jos de Bruijn
```

```
        date: 2004-03-18 -->
        <!-- source XML documents are assumed to be valid according to the schema
                wsml-xml-syntax.xsd with date 2004-03-18 -->
        <!-- destination documents should be valid according to the grammar defined in
                WSMO D16.1, the version published 2004-03-15 -->
        <!-- writes the header information -->
        <xsl:template match="/">
% WSML/BNF generated from WSML/XML syntax using the WSMO XSLT
%
%

<xsl:apply-templates/>
        </xsl:template>

        <xsl:template match="goal" name="goal">
        GOAL
                <xsl:call-template name="id"/><xsl:apply-templates/>
        END
        </xsl:template>
        <xsl:template match="postCondition">
        POSTCONDITION
                <xsl:call-template name="id"/><xsl:apply-templates/>
        END
        </xsl:template>
        <xsl:template match="effect">
        EFFECT
                <xsl:call-template name="id"/><xsl:apply-templates/>
        END
        </xsl:template>
        <xsl:template match="preCondition">
        PRECONDITION
                <xsl:call-template name="id"/><xsl:apply-templates/>
        END
        </xsl:template>
        <xsl:template match="assumption">
        ASSUMPTION
                <xsl:call-template name="id"/><xsl:apply-templates/>
        END
        </xsl:template>

        <!-- start of the web service part -->
        <xsl:template match="webService" name="webService">
        WEBSERVICE
                <xsl:call-template name="id"/><xsl:apply-templates/>
        END
        </xsl:template>
        <xsl:template match="capability">
        CAPABILITY
                <xsl:call-template name="id"/><xsl:apply-templates/>
        END
        </xsl:template>
        <xsl:template match="interface">
        INTERFACE
                <xsl:call-template name="id"/><xsl:apply-templates/>
        END
        </xsl:template>
        <xsl:template match="choreography">
                PLACEHOLDERCHOREOGRAPHY
        </xsl:template>
        <xsl:template match="orchestration">
                PLACEHOLDERORCHESTRATION
        </xsl:template>
        <!-- end of the web service part -->

        <!-- start of the mediator part -->
        <xsl:template match="mediator" name="mediator">
                <xsl:apply-templates/>
        </xsl:template>
        <xsl:template match="ooMediator" name="ooMediator">
        OOMEDIATOR
                <xsl:call-template name="id"/>
                <xsl:apply-templates select="nonFunctionalProperties"/>
                <xsl:apply-templates select="sourceOntology"/>
                <xsl:apply-templates select="sourceooMediator"/>
                <xsl:apply-templates select="targetOntology"/>
                <xsl:apply-templates select="targetGoal"/>
                <xsl:apply-templates select="targetWebService"/>
                <xsl:apply-templates select="targetMediator"/>
                MEDIATIONSERVICE
                        <xsl:apply-templates select="goal"/>
                        <xsl:apply-templates select="wwMediator"/>
```

```
        END
END
</xsl:template>
<xsl:template match="ggMediator" name="ggMediator">
GGMEDIATOR
        <xsl:call-template name="id"/>
        <xsl:apply-templates select="nonFunctionalProperties"/>
        <xsl:apply-templates select="sourceGoal"/>
        <xsl:apply-templates select="sourceggMediator"/>
        <xsl:apply-templates select="targetGoal"/>
        <xsl:apply-templates select="targetggMediator"/>
        <xsl:if test="ooMediator">USEDMEDIATOR <xsl:apply-templates select="ooMediator"/> END</xsl:if>
        <xsl:if test="reduction">REDUCTION <xsl:apply-templates select="reduction"/> END</xsl:if>
END
</xsl:template>
<xsl:template match="wgMediator" name="wgMediator">
WGMEDIATOR
        <xsl:call-template name="id"/>
        <xsl:apply-templates select="nonFunctionalProperties"/>
        <xsl:apply-templates select="sourceWebService"/>
        <xsl:apply-templates select="sourcewgMediator"/>
        <xsl:apply-templates select="targetGoal"/>
        <xsl:apply-templates select="targetwgMediator"/>
        <xsl:if test="ooMediator">USEDMEDIATOR <xsl:apply-templates select="ooMediator"/> END</xsl:if>
        <xsl:if test="reduction">REDUCTION <xsl:apply-templates select="reduction"/> END</xsl:if>
END
</xsl:template>
<xsl:template match="wwMediator" name="wwMediator">
WWMEDIATOR
        <xsl:call-template name="id"/>
        <xsl:apply-templates select="nonFunctionalProperties"/>
        <xsl:apply-templates select="sourceWebService"/>
        <xsl:apply-templates select="sourcewwMediator"/>
        <xsl:apply-templates select="targetWebService"/>
        <xsl:apply-templates select="targetwwMediator"/>
        <xsl:if test="ooMediator">USEDMEDIATOR <xsl:apply-templates select="ooMediator"/> END</xsl:if>
        <xsl:if test="reduction">REDUCTION <xsl:apply-templates select="reduction"/> END</xsl:if>
END
</xsl:template>
<xsl:template match="sourceOntology">
SOURCECOMPONENT
        <xsl:call-template name="ontology"/><xsl:apply-templates/>
END
</xsl:template>
<xsl:template match="sourceooMediator">
SOURCECOMPONENT
        <xsl:call-template name="ooMediator"/><xsl:apply-templates/>
END
</xsl:template>
<xsl:template match="targetOntology">
TARGETCOMPONENT
        <xsl:call-template name="ontology"/><xsl:apply-templates/>
END
</xsl:template>
<xsl:template match="targetGoal">
TARGETCOMPONENT
        <xsl:call-template name="goal"/><xsl:apply-templates/>
END
</xsl:template>
<xsl:template match="targetWebService">
TARGETCOMPONENT
        <xsl:call-template name="webService"/><xsl:apply-templates/>
END
</xsl:template>
<xsl:template match="targetMediator">
TARGETCOMPONENT
        <xsl:call-template name="mediator"/><xsl:apply-templates/>
END
</xsl:template>
<xsl:template match="sourceggMediator">
SOURCECOMPONENT
        <xsl:call-template name="ggMediator"/><xsl:apply-templates/>
END
</xsl:template>
<xsl:template match="sourceGoal">
SOURCECOMPONENT
        <xsl:call-template name="goal"/><xsl:apply-templates/>
END
</xsl:template>
<xsl:template match="targetggMediator">
TARGETCOMPONENT
```

```
        <xsl:call-template name="ggMediator"/><xsl:apply-templates/>
END
</xsl:template>
<xsl:template match="reduction">
        <xsl:call-template name="id"/><xsl:apply-templates/>
</xsl:template>
<xsl:template match="sourceWebService">
SOURCECOMPONENT
        <xsl:call-template name="webService"/><xsl:apply-templates/>
END
</xsl:template>
<xsl:template match="sourcewgMediator">
SOURCECOMPONENT
        <xsl:call-template name="wgMediator"/><xsl:apply-templates/>
END
</xsl:template>
<xsl:template match="targetwgMediator">
TARGETCOMPONENT
        <xsl:call-template name="wgMediator"/><xsl:apply-templates/>
END
</xsl:template>
<xsl:template match="sourcewwMediator">
SOURCECOMPONENT
        <xsl:call-template name="wwMediator"/><xsl:apply-templates/>
END
</xsl:template>
<xsl:template match="targetwwMediator">
TARGETCOMPONENT
        <xsl:call-template name="wwMediator"/><xsl:apply-templates/>
END
</xsl:template>
<!-- end of the mediator part -->


<!-- this is the start of the ontology part -->
<xsl:template match="ontology" name="ontology">
ONTOLOGY
        <xsl:call-template name="id"/><xsl:apply-templates/>
END
</xsl:template>
<xsl:template match="concept">
CONCEPT
        <xsl:call-template name="id"/><xsl:apply-templates/>
END
</xsl:template>
<xsl:template match="superConcept">
SUPERCONCEPT
        <xsl:call-template name="id"/><xsl:apply-templates/>
END
</xsl:template>
<xsl:template match="relation">
RELATION
        <xsl:call-template name="id"/><xsl:apply-templates/>
END
</xsl:template>
<xsl:template match="instance">
INSTANCE
        <xsl:call-template name="id"/><xsl:apply-templates/>
END
</xsl:template>
<xsl:template match="attribute">
ATTRIBUTE
        <xsl:call-template name="id"/><xsl:apply-templates/>
END
</xsl:template>
<xsl:template match="method">
METHOD
        <xsl:call-template name="id"/><xsl:apply-templates/>
END
</xsl:template>
<xsl:template match="range">
RANGE
        <xsl:call-template name="id"/><xsl:apply-templates/>
END
</xsl:template>
<xsl:template match="parameter">
PARAMETER
        <xsl:call-template name="id"/><xsl:apply-templates/>
END
</xsl:template>
<xsl:template match="domain">
```

```
DOMAIN
        <xsl:call-template name="id"/><xsl:apply-templates/>
END
</xsl:template>
<xsl:template match="instanceOf">
INSTANCEOF
        <xsl:call-template name="id"/><xsl:apply-templates/>
END
</xsl:template>
<xsl:template match="attributeValue">
ATTRIBUTEVALUE
        <xsl:call-template name="id"/><xsl:apply-templates/>
END
</xsl:template>
<xsl:template match="value">
VALUE
        <xsl:call-template name="id"/><xsl:apply-templates/>
END
</xsl:template>
<!-- this is the end of the ontology part -->

<xsl:template match="nonFunctionalProperties">
        <xsl:apply-templates/>
</xsl:template>

<xsl:template name="id">
        <xsl:if test="@id">IDENTIFIER <xsl:value-of select="@id"/> END</xsl:if>
</xsl:template>

<!-- these are all the non-functional properties -->
<xsl:template match="title">
        TITLE <xsl:value-of select="."/> END
</xsl:template>
<xsl:template match="creator">
        CREATOR <xsl:value-of select="."/> END
</xsl:template>
<xsl:template match="subject">
        TITLE <xsl:value-of select="."/> END
</xsl:template>
<xsl:template match="description">
        DESCRIPTION <xsl:value-of select="."/> END
</xsl:template>
<xsl:template match="publisher">
        PUBLISHER <xsl:value-of select="."/> END
</xsl:template>
<xsl:template match="contributor">
        CONTRIBUTOR <xsl:value-of select="."/> END
</xsl:template>
<xsl:template match="date">
        DATE <xsl:value-of select="."/> END
</xsl:template>
<xsl:template match="type">
        TYPE <xsl:value-of select="."/> END
</xsl:template>
<xsl:template match="format">
        FORMAT <xsl:value-of select="."/> END
</xsl:template>
<xsl:template match="source">
        SOURCE <xsl:value-of select="."/> END
</xsl:template>
<xsl:template match="language">
        LANGUAGE <xsl:value-of select="."/> END
</xsl:template>
<xsl:template match="relation">
        RELATION <xsl:value-of select="."/> END
</xsl:template>
<xsl:template match="coverage">
        COVERAGE <xsl:value-of select="."/> END
</xsl:template>
<xsl:template match="rights">
        RIGHTS <xsl:value-of select="."/> END
</xsl:template>
<xsl:template match="version">
        VERSION <xsl:value-of select="."/> END
</xsl:template>
<xsl:template match="performance">
        PERFORMANCE <xsl:value-of select="."/> END
</xsl:template>
<xsl:template match="reliability">
        RELIABILITY <xsl:value-of select="."/> END
</xsl:template>
```

```
        <xsl:template match="security">
                SECURITY <xsl:value-of select="."/> END
        </xsl:template>
        <xsl:template match="scalability">
                SCALABILITY <xsl:value-of select="."/> END
        </xsl:template>
        <xsl:template match="robustness">
                ROBUSTNESS <xsl:value-of select="."/> END
        </xsl:template>
        <xsl:template match="accuracy">
                ACCURACY <xsl:value-of select="."/> END
        </xsl:template>
        <xsl:template match="transactional">
                TRANSACTIONAL <xsl:value-of select="."/> END
        </xsl:template>
        <xsl:template match="trust">
                TRUST <xsl:value-of select="."/> END
        </xsl:template>
        <xsl:template match="financial">
                FINANCIAL <xsl:value-of select="."/> END
        </xsl:template>
        <xsl:template match="networkRelatedQoS">
                NETWORKRELATEDQOS <xsl:value-of select="."/> END
        </xsl:template>
        <!-- end of the non-functional properties -->


        <!-- ********************************* start of the F-Logic transformations ***************************** -

        <!-- writes the header information for F-Logic expressions -->
        <xsl:template match="flogicExpression">
<xsl:apply-templates select="rule"/>
        </xsl:template>

        <!-- the following is copied from the flogic-transform.xslt document, which provides the translation from
                F-Logic/XML into F-Logic/BNF -->
<!-- In the generated F-Logic syntax, the AND has precedence over OR and since both AND and OR are
        commutative, no brackets are needed -->

<!-- writes the rule itself; the building block for F-Logic statements -->
        <xsl:template match="rule">
<xsl:for-each select="head"><xsl:call-template name="conjunct"/><xsl:if
        test="position()!=last()"> AND </xsl:if></xsl:for-each><xsl:if
        test="body"> <- </xsl:if><xsl:for-each select="body"><xsl:call-template
        name="conjunct"/><xsl:if test="position()!=last()"> AND </xsl:if></xsl:for-each>.
</xsl:template>
        <xsl:template match="conjunct" name="conjunct">
                <xsl:if test="variable"><xsl:if
                        test="ancestor-or-self::body">FORALL </xsl:if><xsl:if
                        test="ancestor-or-self::head">EXISTS </xsl:if><xsl:for-each
                        select="variable"><xsl:apply-templates select="."/><xsl:if
                        test="position()!=last()">,</xsl:if></xsl:for-each> (</xsl:if><xsl:for-each
                        select="molecule"><xsl:apply-templates select="."/><xsl:if
                        test="position()!=last()"> AND </xsl:if></xsl:for-each><xsl:for-each
                        select="disjunct"><xsl:apply-templates select="."/><xsl:if
                        test="position()!=last()"> OR </xsl:if></xsl:for-each><xsl:if
                        test="variable">)</xsl:if></xsl:template>
        <xsl:template match="functor">
                <xsl:apply-templates/>(<xsl:for-each select="../argument"><xsl:apply-templates
                        select="*"/><xsl:if test="position()!=last()">, </xsl:if></xsl:for-each>)</xsl:template>
        <xsl:template match="superclass">
                <xsl:value-of select="@isaType"/>
                <xsl:apply-templates/>
        </xsl:template>
        <xsl:template match="methodSpec">[<xsl:apply-templates select="name"/> <xsl:value-of
                select="@arrow"/><xsl:apply-templates select="result"/>]</xsl:template>
        <xsl:template match="variable">
                <xsl:value-of select="@name"/>
        </xsl:template>
        <xsl:template match="molecule">
                <xsl:apply-templates/>
        </xsl:template>
        <xsl:template match="disjunct">
                <xsl:if test="variable"><xsl:if test="ancestor-or-self::body">FORALL </xsl:if><xsl:if
                        test="ancestor-or-self::head">EXISTS </xsl:if><xsl:for-each
                        select="variable"><xsl:apply-templates select="."/><xsl:if
                        test="position()!=last()">, </xsl:if></xsl:for-each></xsl:if><xsl:for-each
                        select="molecule"><xsl:apply-templates select="."/><xsl:if
                        test="position()!=last()"> AND </xsl:if></xsl:for-each><xsl:for-each
                        select="conjunct"><xsl:apply-templates select="."/><xsl:if
                        test="position()!=last()"> AND </xsl:if></xsl:for-each><xsl:if
```

```
                        test="variable">)</xsl:if></xsl:template>
        <xsl:template match="constant">
                <xsl:value-of select="@name"/>
        </xsl:template>
        <xsl:template match="argument"></xsl:template>
</xsl:stylesheet>
```

Acknowledgements

The work is funded by the European Commission under the projects DIP, Knowledge Web, Ontoweb, SEKT, SWWS, Esperonto, COG and h-TechSight; by Science Foundation Ireland under the DERI-Lion project; and by the Vienna city government under the CoOperate programme.

The editors would like to thank to all the members of the WSMO working group for their advises and inputs to this document.

$Date: 2004/03/22 10:37:33 $

webmaster