OWL 2 Web Ontology Language Quick Reference Guide

W3C Editor's Draft 17 September 2009

This version:
http://www.w3.org/2007/OWL/draft/ED-owl2-quick-reference-20090917/
Latest editor's draft:
http://www.w3.org/2007/OWL/draft/owl2-quick-reference/
Previous version:
http://www.w3.org/2007/OWL/draft/ED-owl2-quick-reference-20090914/
(color-coded diff)

Editors:
Jie Bao, Rensselaer Polytechnic Institute
Elisa F. Kendall, Sandpiper Software, Inc.
Deborah L. McGuinness, Rensselaer Polytechnic Institute
Peter F. Patel-Schneider, Bell Labs Research, Alcatel-Lucent
Contributors:
Li Ding, Rensselaer Polytechnic Institute
Ankesh Khandelwal, Rensselaer Polytechnic Institute

This document is also available in these non-normative formats: PDF version.

Abstract

The OWL 2 Web Ontology Language, informally OWL 2, is an ontology language for the Semantic Web with formally defined meaning. OWL 2 ontologies provide classes, properties, individuals, and data values and are stored as Semantic Web documents. OWL 2 ontologies can be used along with information written in RDF, and OWL 2 ontologies themselves are primarily exchanged as RDF documents. The OWL 2 Document Overview describes the overall state of OWL 2, and should be read before other OWL 2 documents.

Status of this Document

May Be Superseded

This section describes the status of this document at the time of its publication. Other documents may supersede this document. A list of current W3C publications and the latest revision of this technical report can be found in the W3C technical reports index at http://www.w3.org/TR/.

XML Schema Datatypes Dependency

OWL 2 is defined to use datatypes defined in the XML Schema Definition Language (XSD). As of this writing, the latest W3C Recommendation for XSD is version 1.0, with version 1.1 progressing toward Recommendation. OWL 2 has been designed to take advantage of the new datatypes and clearer explanations available in XSD 1.1, but for now those advantages are being partially put on hold. Specifically, until XSD 1.1 becomes a W3C Recommendation, the elements of OWL 2 which are based on it should be considered *optional*, as detailed in Conformance, section 2.3. Upon the publication of XSD 1.1 as a W3C Recommendation, those elements cease to be optional and are to be considered required as otherwise specified.

We suggest that for now developers and users follow the XSD 1.1 Candidate Recommendation. Based on discussions between the Schema and OWL Working Groups, we do not expect any implementation changes will be necessary as XSD 1.1 advances to Recommendation.

Summary of Changes

There have been no substantive changes since the previous version. For details on the minor changes see the change log and color-coded diff.

W3C Members Please Review By 15 October 2009

The W3C Director seeks review and feedback from W3C Advisory Committee representatives, via their review form by 15 October 2009. This will allow the Director to assess consensus and determine whether to issue this document as a W3C Recommendation.

Others are encouraged by the OWL Working Group to continue to send reports of implementation experience, and other feedback, to public-owl-comments@w3.org (public archive). Reports of any success or difficulty with the test cases are encouraged. Open discussion among developers is welcome at public-owl-dev@w3.org (public archive).

Support

The advancement of this Proposed Recommendation is supported by the disposition of comments on the Candidate Recommendation, the Test Suite with Test Results, and the list of implementations.

No Endorsement

Publication as a Editor's Draft does not imply endorsement by the W3C Membership. This is a draft document and may be updated, replaced or obsoleted by other documents at any time. It is inappropriate to cite this document as other than work in progress.

Patents

This document was produced by a group operating under the 5 February 2004 W3C Patent Policy. This document is informative only. W3C maintains a public list of any patent disclosures made in connection with the deliverables of the group; that page also includes instructions for disclosing a patent. An individual who has actual knowledge of a patent which the individual believes contains Essential Claim(s) must disclose the information in accordance with section 6 of the W3C Patent Policy.

Table of Contents

1 Names, Prefixes, and Notation

Names in OWL 2 are IRIs, often written in a shorthand `prefix:local_name`, where `prefix:` is a prefix name that expands to an IRI, and `local_name` is the remainder of the name. The standard prefix names in OWL 2 are:

Prefix Name	Expansion
rdf:	http://www.w3.org/1999/02/22-rdf-syntax-ns#
rdfs:	http://www.w3.org/2000/01/rdf-schema#
owl:	http://www.w3.org/2002/07/owl#
xsd:	http://www.w3.org/2001/XMLSchema#

We use notation conventions in the following table*:

Letters	Meaning	Letters	Meaning	Letters	Meaning	Letters	Meaning
C	class expression	CN	class name	D	data range	DN	datatype name
P	object property expression	PN	object property name	R	data property	A	annotation property
a	individual	aN	individual name	_:a	anonymous individual (a blank node label)	v	literal
n	non-negative integer**	f	facet	ON	ontology name	U	IRI
s	IRI or anonymous individual	t	IRI, anonymous individual, or literal	p	prefix name	_:x	blank node
(a_1 … a_n)	RDF list						

* All of the above can have subscripts. ** as a shorthand for "n"^^xsd:nonNegativeInteger

2 OWL 2 constructs and axioms

For an OWL 2 DL ontology, there are some global restrictions on axioms.

In the following tables the first column provides links to the Primer (if applicable) and the 2nd column provides links to the Functional Syntax.

2.1 Class Expressions

Predefined and Named Classes

Language Feature	Functional Syntax	RDF Syntax
named class	CN	CN
universal class	owl:Thing	owl:Thing
empty class	owl:Nothing	owl:Nothing

Boolean Connectives and Enumeration of Individuals

Language Feature	Functional Syntax	RDF Syntax
intersection	ObjectIntersectionOf(C_1 … C_n)	_:x rdf:type owl:Class. _:x owl:intersectionOf (C_1 … C_n).
union	ObjectUnionOf(C_1 … C_n)	_:x rdf:type owl:Class. _:x owl:unionOf (C_1 … C_n).
complement	ObjectComplementOf(C)	_:x rdf:type owl:Class. _:x owl:complementOf C.
enumeration	ObjectOneOf(a_1 … a_n)	_:x rdf:type owl:Class. _:x owl:oneOf (a_1 … a_n).

Object Property Restrictions

Language Feature	Functional Syntax	RDF Syntax
universal	ObjectAllValuesFrom(P C)	_:x rdf:type owl:Restriction. _:x owl:onProperty P. _:x owl:allValuesFrom C
existential	ObjectSomeValuesFrom(P C)	_:x rdf:type owl:Restriction. _:x owl:onProperty P. _:x owl:someValuesFrom C
individual value	ObjectHasValue(P a)	_:x rdf:type owl:Restriction. _:x owl:onProperty P. _:x owl:hasValue a.
local reflexivity	ObjectHasSelf(P)	_:x rdf:type owl:Restriction. _:x owl:onProperty P. _:x owl:hasSelf "true"^^xsd:boolean.
exact cardinality	ObjectExactCardinality(n P)	_:x rdf:type owl:Restriction. _:x owl:onProperty P. _:x owl:cardinality n.
qualified exact cardinality	ObjectExactCardinality(n P C)	_:x rdf:type owl:Restriction. _:x owl:onProperty P. _:x owl:qualifiedCardinality n. _:x owl:onClass C.
maximum cardinality	ObjectMaxCardinality(n P)	_:x rdf:type owl:Restriction. _:x owl:onProperty P. _:x owl:minCardinality n.
qualified maximum cardinality	ObjectMaxCardinality(n P C)	_:x rdf:type owl:Restriction. _:x owl:onProperty P. _:x

		owl:minQualifiedCardinality n. _:x owl:onClass C.
minimum cardinality	ObjectMinCardinality(n P)	_:x rdf:type owl:Restriction. _:x owl:onProperty P. _:x owl:maxCardinality n.
qualified minimum cardinality	ObjectMinCardinality(n P C)	_:x rdf:type owl:Restriction. _:x owl:onProperty P. _:x owl:maxQualifiedCardinality n. _:x owl:onClass C.

Data Property Restrictions

Language Feature	Functional Syntax	RDF Syntax
universal	DataAllValuesFrom(R D)	_:x rdf:type owl:Restriction. _:x owl:onProperty R. _:x owl:allValuesFrom D.
existential	DataSomeValuesFrom(R D)	_:x rdf:type owl:Restriction. _:x owl:onProperty R. _:x owl:someValuesFrom D.
literal value	DataHasValue(R v)	_:x rdf:type owl:Restriction. _:x owl:onProperty R. _:x owl:hasValue v.
exact cardinality	DataExactCardinality(n R)	_:x rdf:type owl:Restriction. _:x owl:onProperty R. _:x owl:cardinality n.
qualified exact cardinality	DataExactCardinality(n R D)	_:x rdf:type owl:Restriction. _:x owl:onProperty R. _:x owl:qualifiedCardinality n. _:x owl:onDataRange D.
maximum cardinality	DataMaxCardinality(n R)	_:x rdf:type owl:Restriction. _:x owl:onProperty R. _:x owl:maxCardinality n.
qualified maximum cardinality	DataMaxCardinality(n R D)	_:x rdf:type owl:Restriction. _:x owl:onProperty R. _:x owl:maxQualifiedCardinality n. _:x owl:onDataRange D.
minimum cardinality	DataMinCardinality(n R)	_:x rdf:type owl:Restriction. _:x owl:onProperty R. _:x owl:minCardinality n.
qualified minimum cardinality	DataMinCardinality(n R D)	_:x rdf:type owl:Restriction. _:x owl:onProperty R. _:x owl:minQualifiedCardinality n. _:x owl:onDataRange D.

Restrictions Using n-ary Data Range

In the following table 'D^n' is an n-ary data range.

Language Feature	Functional Syntax	RDF Syntax
n-ary universal	DataAllValuesFrom(R_1 … R_n D^n)	_:x rdf:type owl:Restriction. _:x owl:onProperties (R_1 … R_n). _:x owl:allValuesFrom D^n.
n-ary existential	DataSomeValuesFrom(R_1 … R_n D^n)	_:x rdf:type owl:Restriction. _:x owl:onProperties (R_1 … R_n). _:x owl:someValuesFrom D^n.

2.2 Properties

Object Property Expressions

Language Feature	Functional Syntax	RDF Syntax
named object property	PN	PN
universal object property	owl:topObjectProperty	owl:topObjectProperty
empty object property	owl:bottomObjectProperty	owl:bottomObjectProperty
inverse property	ObjectInverseOf(PN)	_:x owl:inverseOf PN

Data Property Expressions

Language Feature	Functional Syntax	RDF Syntax
named data property	R	R
universal data property	owl:topDataProperty	owl:topDataProperty
empty data property	owl:bottomDataProperty	owl:bottomDataProperty

2.3 Individuals & Literals

Language Feature	Functional Syntax	RDF Syntax
named individual	aN	aN
anonymous individual	_:a	_:a
literal (datatype value)	"abc"^^DN	"abc"^^DN

2.4 Data Ranges

Data Range Expressions

Language Feature	Functional Syntax	RDF Syntax
named datatype	DN	DN
data range complement	DataComplementOf(D)	_:x rdf:type rdfs:Datatype. _:x owl:datatypeComplementOf D.
data range intersection	DataIntersectionOf(D_1…D_n)	_:x rdf:type rdfs:Datatype. _:x owl:intersectionOf (D_1…D_n).

data range union	DataUnionOf(D_1…D_n)	_:x rdf:type rdfs:Datatype. _:x owl:unionOf (D_1…D_n).
literal enumeration	DataOneOf(v_1 … v_n)	_:x rdf:type rdfs:Datatype. _:x owl:oneOf (v_1 … v_n).
datatype restriction	DatatypeRestriction(DN f_1 v_1 … f_n v_n)	_:x rdf:type rdfs:Datatype. _:x owl:onDatatype DN. _:x owl:withRestrictions (_:x_1 … _:x_n). _:x_j f_j v_j. j=1…n

2.5 Axioms

Class Expression Axioms

Language Feature	Functional Syntax	RDF Syntax
subclass	SubClassOf(C_1 C_2)	C_1 rdfs:subClassOf C_2.
equivalent classes	EquivalentClasses(C_1 … C_n)	C_j owl:equivalentClass C_{j+1}. j=1…n-1
disjoint classes	DisjointClasses(C_1 C_2)	C_1 owl:disjointWith C_2.
pairwise disjoint classes	DisjointClasses(C_1 … C_n)	_:x rdf:type owl:AllDisjointClasses. _:x owl:members (C_1 … C_n).
disjoint union	DisjointUnionOf(CN C_1 … C_n)	CN owl:disjointUnionOf (C_1 … C_n).

Object Property Axioms

Language Feature	Functional Syntax	RDF Syntax
subproperty	SubObjectPropertyOf(P_1 P_2)	P_1 rdfs:subPropertyOf P_2.
property chain inclusion	SubObjectPropertyOf(ObjectPropertyChain(P_1 … P_n) P)	P owl:propertyChainAxiom (P_1 … P_n).
property domain	ObjectPropertyDomain(P C)	P rdfs:domain C.
property range	ObjectPropertyRange(P C)	P rdfs:range C.
equivalent properties	EquivalentObjectProperties(P_1 … P_n)	P_j owl:equivalentProperty P_{j+1}. j=1…n-1
disjoint properties	DisjointObjectProperties(P_1 P_2)	P_1 owl:propertyDisjointWith P_2.
pairwise disjoint properties	DisjointObjectProperties(P_1 … P_n)	_:x rdf:type owl:AllDisjointProperties. _:x owl:members (P_1 … P_n).
inverse properties	InverseObjectProperties(P_1 P_2)	P_1 owl:inverseOf P_2.
functional property	FunctionalObjectProperty(P)	P rdf:type owl:FunctionalProperty.
inverse functional property	InverseFunctionalObjectProperty(P)	P rdf:type owl:InverseFunctionalProperty.
reflexive property	ReflexiveObjectProperty(P)	P rdf:type owl:ReflexiveProperty.

irreflexive property	IrreflexiveObjectProperty(P)	P rdf:type owl:IrreflexiveProperty.
symmetric property	SymmetricObjectProperty(P)	P rdf:type owl:SymmetricProperty.
asymmetric property	AsymmetricObjectProperty(P)	P rdf:type owl:AsymmetricProperty.
transitive property	TransitiveObjectProperty(P)	P rdf:type owl:TransitiveProperty.

Data Property Axioms

Language Feature	Functional Syntax	RDF Syntax
subproperty	SubDataPropertyOf(R_1 R_2)	R_1 rdfs:subPropertyOf R_2.
property domain	DataPropertyDomain(R C)	R rdfs:domain C.
property range	DataPropertyRange(R D)	R rdfs:range D.
equivalent properties	EquivalentDataProperties(R_1 … R_n)	R_j owl:equivalentProperty R_{j+1}. j=1…n-1
disjoint properties	DisjointDataProperties(R_1 R_2)	R_1 owl:propertyDisjointWith R_2.
pairwise disjoint properties	DisjointDataProperties(R_1 … R_n)	_:x rdf:type owl:AllDisjointProperties. _:x owl:members (R_1 … R_n).
functional property	FunctionalDataProperty(R)	R rdf:type owl:FunctionalProperty.

Datatype Definitions

Language Feature	Functional Syntax	RDF Syntax
datatype definition	DatatypeDefinition(DN D)	DN owl:equivalentClass D.

Assertions

Language Feature	Functional Syntax	RDF Syntax
individual equality	SameIndividual(a_1 … a_n)	a_j owl:sameAs a_{j+1}. j=1…n-1
individual inequality	DifferentIndividuals(a_1 a_2)	a_1 owl:differentFrom a_2.
pairwise individual inequality	DifferentIndividuals(a_1 … a_n)	_:x rdf:type owl:AllDifferent. _:x owl:members (a_1 … a_n).
class assertion	ClassAssertion(C a)	a rdf:type C.
positive object property assertion	ObjectPropertyAssertion(PN a_1 a_2)	a_1 PN a_2.
positive data property assertion	DataPropertyAssertion(R a v)	a R v.
negative object	NegativeObjectPropertyAssertion(P a_1 a_2)	_:x rdf:type owl:NegativePropertyAssertion. _:x owl:sourceIndividual a_1.

property assertion		_:x owl:assertionProperty P. _:x owl:targetIndividual a_2.
negative data property assertion	NegativeDataPropertyAssertion(R a v)	_:x rdf:type owl:NegativePropertyAssertion. _:x owl:sourceIndividual a. _:x owl:assertionProperty R. _:x owl:targetValue v.

Keys

Language Feature	Functional Syntax	RDF Syntax
Key	HasKey(C (P_1 … P_m) (R_1 … R_n))	C owl:hasKey (P_1 … P_m R_1 … R_n).

2.6 Declarations

Language Feature	Functional Syntax	RDF Syntax
class	Declaration(Class(CN))	CN rdf:type owl:Class.
datatype	Declaration(Datatype(DN))	DN rdf:type rdfs:Datatype.
object property	Declaration(ObjectProperty(PN))	PN rdf:type owl:ObjectProperty.
data property	Declaration(DataProperty(R))	R rdf:type owl:DatatypeProperty.
annotation property	Declaration(AnnotationProperty(A))	A rdf:type owl:AnnotationProperty.
named individual	Declaration(NamedIndividual(aN))	aN rdf:type owl:NamedIndividual.

2.7 Annotations

Annotations

Language Feature	Functional Syntax	RDF Syntax
annotation assertion	AnnotationAssertion(A s t)	s A t.
annotation of an axiom where the axiom in RDF is one or more triples with the same predicate s_i U t_i	AXIOM(Annotation(A t) …)	_:x_i A t. s_i U t_i. … _:x_i rdf:type owl:Axiom. _:x_i owl:annotatedSource s_i. _:x_i owl:annotatedProperty U. _:x_i owl:annotatedTarget t_i.

annotation of an axiom where the axiom in RDF starts with _:x	AXIOM(Annotation(A t) …)	_:x A t. _:x …. …
annotation of another annotation (the other annotation in RDF starts with s_1)	Annotation(Annotation(A t) … A_1 t_1)	_:x A t. s_1 A_1 t_1. _:x rdf:type owl:Annotation. _:x owl:annotatedSource s_1. _:x owl:annotatedProperty A_1. _:x owl:annotatedTarget t_1.

Annotation Properties

Language Feature	Functional Syntax	RDF Syntax
named annotation property	A	A
human-readable name	rdfs:label	rdfs:label
human-readable comment	rdfs:comment	rdfs:comment
additional information	rdfs:seeAlso	rdfs:seeAlso
defining agent	rdfs:isDefinedBy	rdfs:isDefinedBy
version information	owl:versionInfo	owl:versionInfo
deprecation	owl:deprecated	owl:deprecated
backwards compatibility	owl:backwardCompatibleWith	owl:backwardCompatibleWith
incompatibility	owl:incompatibleWith	owl:incompatibleWith
prior version	owl:priorVersion	owl:priorVersion

Annotation Axioms

Language Feature	Functional Syntax	RDF Syntax
annotation subproperties	SubAnnotationPropertyOf(A_1 A_2)	A_1 rdfs:subPropertyOf A_2.
annotation property domain	AnnotationPropertyDomain(A U)	A rdfs:domain U.
annotation property range	AnnotationPropertyRange(A U)	A rdfs:range U.

2.8 Ontologies

Ontologies

Language Feature	Functional Syntax	RDF Syntax
OWL ontology (importing)*	Ontology([ON [U]] Import(ON_1)...	ON rdf:type owl:Ontology. [ON owl:versionIRI U.] ON owl:imports ON_1. ...

	Annotation(A t) ...)	ON A t. ...
prefix declaration**	Prefix(p=U)	@prefix p U.

Note *: in the RDF syntax _:x is used in place of ON if there is no ontology name. ** RDF syntax is in Turtle, other RDF serializations may vary.

3 Built-in Datatypes and Facets

3.1 Built-in Datatypes

Universal Datatype	rdfs:Literal			
Numbers	owl:rational		owl:real	
	xsd:double	xsd:float	xsd:decimal	xsd:integer
	xsd:long	xsd:int	xsd:short	xsd:byte
	xsd:nonNegativeInteger		xsd:nonPositiveInteger	
	xsd:positiveInteger		xsd:negativeInteger	
	xsd:unsignedLong		xsd:unsignedInt	
	xsd:unsignedShort		xsd:unsignedByte	
Strings	rdf:PlainLiteral (RDF plain literals)			
	xsd:string	xsd:NCName	xsd:Name	xsd:NMTOKEN
	xsd:token	xsd:language	xsd:normalizedString	
Boolean Values	xsd:boolean (value space: *true* and *false*)			
Binary Data	xsd:base64Binary		xsd:hexBinary	
IRIs	xsd:anyURI			
Time Instants	xsd:dateTime (optional time zone offset)			
	xsd:dateTimeStamp (required time zone offset)			
XML Literals	rdf:XMLLiteral			

3.2 **Facets**

Facet	Value	Applicable Datatypes	Explanation
xsd:minInclusive xsd:maxInclusive xsd:minExclusive xsd:maxExclusive	literal in the corresponding datatype	Numbers, Time Instants	Restricts the value-space to greater than (equal to) or lesser than (equal to) a value
xsd:minLength xsd:maxLength xsd:length	Non-negative integer	Strings, Binary Data, IRIs	Restricts the value-space based on the lengths of the literals
xsd:pattern	xsd:string literal as a regular expression	Strings, IRIs	Restricts the value space to literals that> match the regular expression
rdf:langRange	xsd:string literal as a regular expression	rdf:PlainLiteral	Restricts the value space to literals with language tags that match the regular expression

4 Appendix

4.1 New Features in OWL 2

Class Expressions	• local reflexivity (self restriction) • object and data qualified exact/maximum/minimal cardinality restriction • universal and existential restriction on n-ary data range
Class Axioms	• pairwise disjoint classes • class disjoint union
Property Expressions	• universal and empty object property • universal and empty data property • inverse object property expression
Property Axioms	• property chain inclusion • disjoint object properties • disjoint data properties • reflexive, irreflexive, and asymmetric object property.
Data Ranges	• datatype definition • data range complement, intersection and union • datatype restriction and facets • hook for n-ary datatype
Assertions	• negative object property assertion • negative data property assertion
Annotation	• annotation assertion • annotation of an axiom or an annotation • annotation subproperties • annotation property domain and range • owl:deprecated annotation property
Extra Built-in Datatypes	• owl:rational, owl:real, xsd:dateTimeStamp, rdf:PlainLiteral
Others	• key • declaration • metamodeling capabilities (Punning) • anonymous individual

4.2 Additional Vocabulary in OWL 2 RDF Syntax

Feature	Vocabulary	Note
data range	owl:DataRange	deprecated in OWL 2, replaced by rdfs:Datatype
membership of a set of pairwise different individuals	owl:distinctMembers	can alternatively use owl:members
ontology property	owl:OntologyProperty	

deprecation	owl:DeprecatedClass,
	owl:DeprecatedProperty
		alternative RDF syntax:
		`s rdf:type owl:DeprecatedClass .`
		or
		`s rdf:type owl:DeprecatedProperty`

		. can be replaced by `s owl:deprecated "true"^^xsd:boolean .`

5 Appendix: Change Log (Informative)

5.1 Changes Since Last Call

This section summarizes the changes to this document since the Candidate Recommendation of 11 June, 2009.

- The "Features At Risk" note w.r.t. the owl:rational and rdf:XMLLiteral datatypes was removed: implementation support has been adequately demonstrated, and the features are no longer considered at risk (see Resolution 5 and Resolution 6, 05 August 2009).
- Some minor editorial changes were made.

6 Acknowledgments

The starting point for the development of OWL 2 was the OWL1.1 member submission, itself a result of user and developer feedback, and in particular of information gathered during the OWL Experiences and Directions (OWLED) Workshop series. The working group also considered postponed issues from the WebOnt Working Group.

This document has been produced by the OWL Working Group (see below), and its contents reflect extensive discussions within the Working Group as a whole. The editors extend special thanks to Bernardo Cuenca Grau (Oxford University), Christine Golbreich (Université de Versailles St-Quentin and LIRMM), Ivan Herman (W3C/ERCIM), and Bijan Parsia (University of Manchester) for their thorough reviews.

The regular attendees at meetings of the OWL Working Group at the time of publication of this document were: Jie Bao (RPI), Diego Calvanese (Free University of Bozen-Bolzano), Bernardo Cuenca Grau (Oxford University Computing Laboratory), Martin Dzbor (Open University), Achille Fokoue (IBM Corporation), Christine Golbreich (Université de Versailles St-Quentin and LIRMM), Sandro Hawke (W3C/MIT), Ivan Herman (W3C/ERCIM), Rinke Hoekstra (University of Amsterdam), Ian Horrocks (Oxford University Computing Laboratory), Elisa Kendall (Sandpiper Software), Markus Krötzsch (FZI), Carsten Lutz (Universität Bremen), Deborah L. McGuinness (RPI), Boris Motik (Oxford University Computing Laboratory), Jeff Pan (University of Aberdeen), Bijan Parsia (University of Manchester), Peter F. Patel-Schneider (Bell Labs Research, Alcatel-Lucent), Sebastian Rudolph (FZI), Alan Ruttenberg (Science Commons), Uli Sattler (University of Manchester), Michael Schneider (FZI), Mike Smith (Clark & Parsia), Evan Wallace (NIST), Zhe Wu (Oracle Corporation), and Antoine Zimmermann (DERI Galway). We would also like to thank past members of the working group: Jeremy Carroll, Jim Hendler, Vipul Kashyap.