

OWL2(Web Ontology Language 2)のご紹介

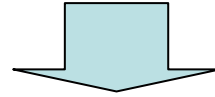
慶応義塾大学SFC研究所

清水 昇

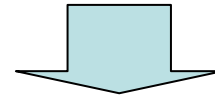
2011年3月4日(金)

OWL2とは

2004年に最初のオントロジ記述言語であるOWL1の仕様を発行



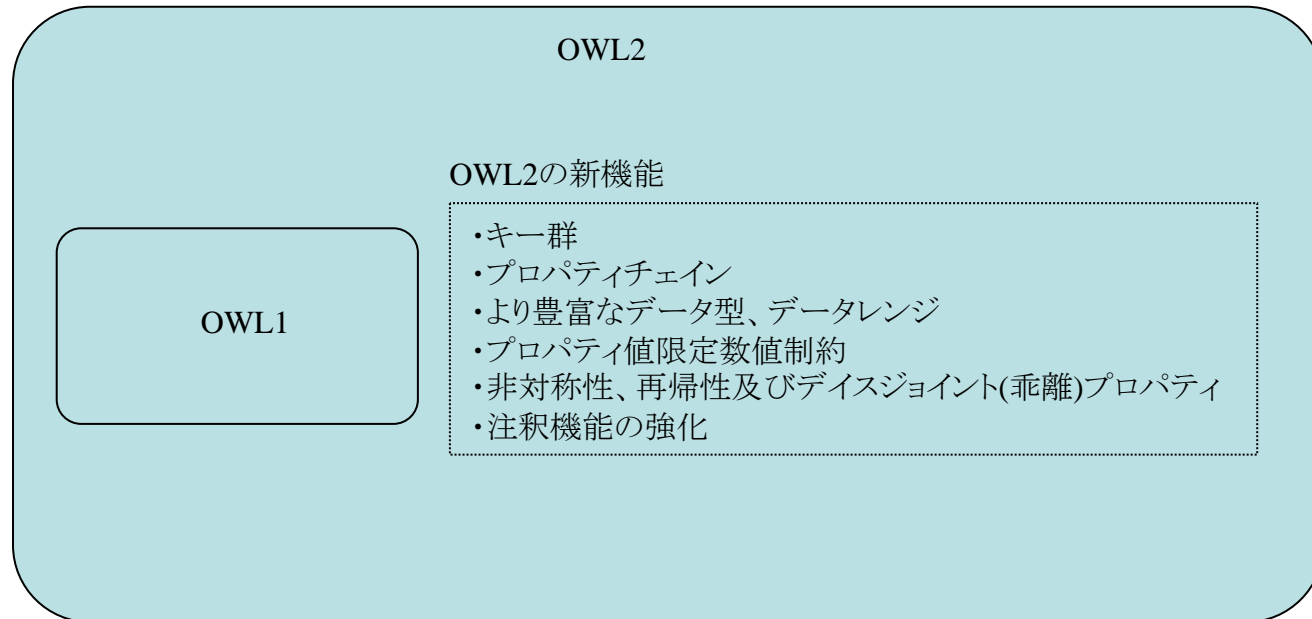
OWL1の欠点や限界の顕在化とDL (Description Logic)に関する研究の進展



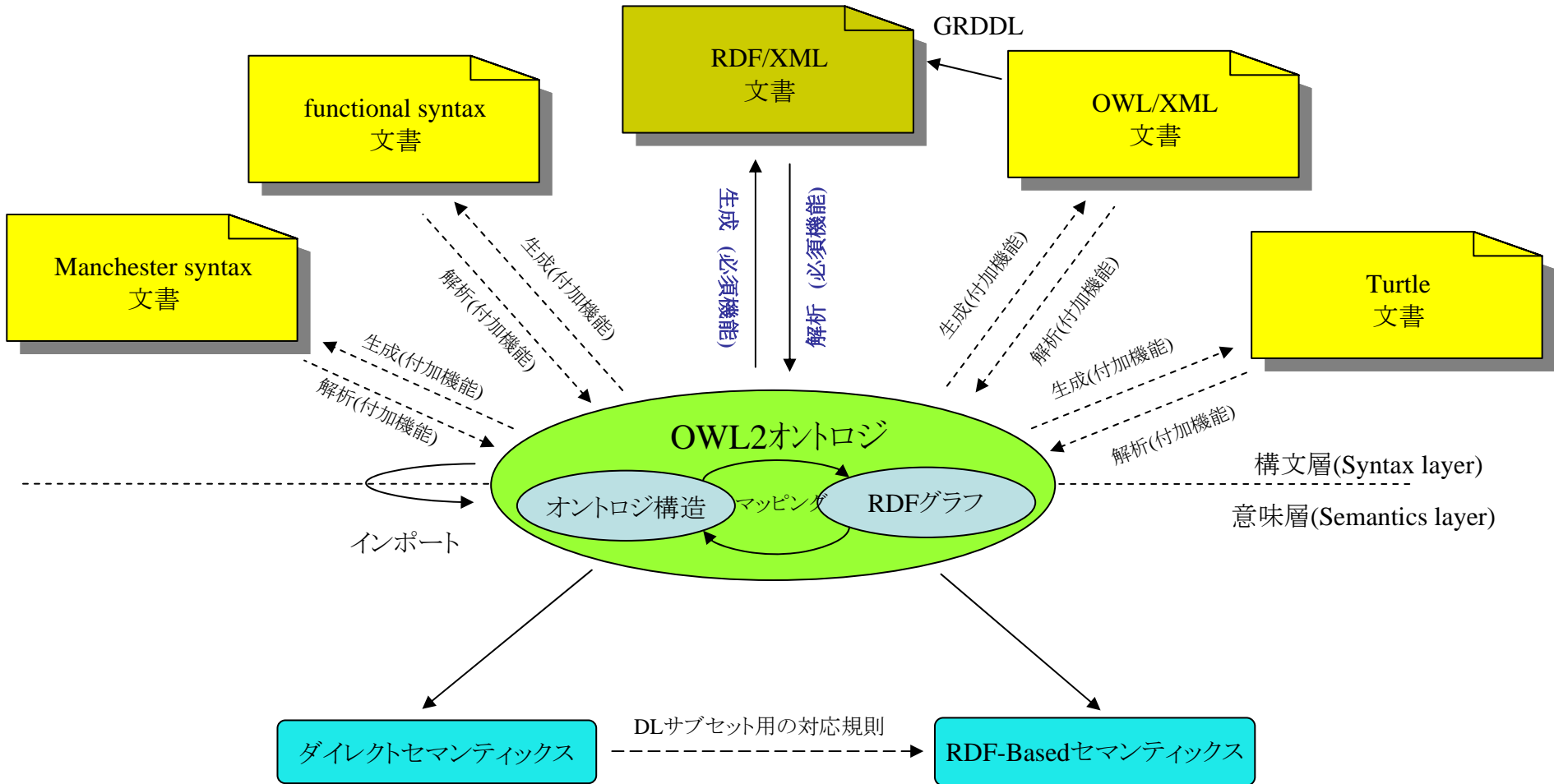
OWLの実装経験とDLに関する研究成果とを踏まえてOWLの仕様を改訂してOWL2を開発し、2009年12月にOWL2の仕様書を公開

OWL2とOWL1との関係

OWL2は、2004年に発行されたOWL(即ち、OWL1)の拡張したものである。



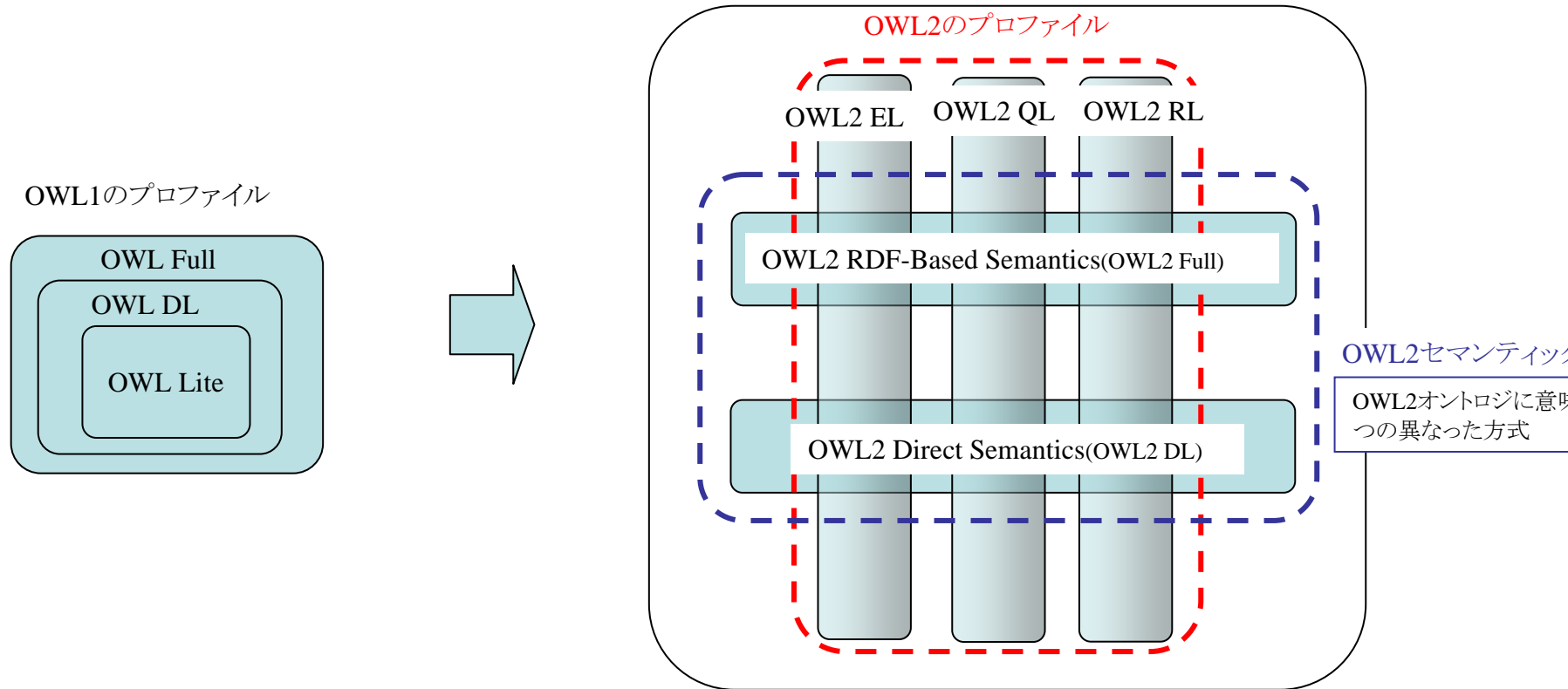
OWL2の構成



GRDDL: Gleaning Resource Descriptions from Dialects of Languages (方言言語からのリソース記述の収集)

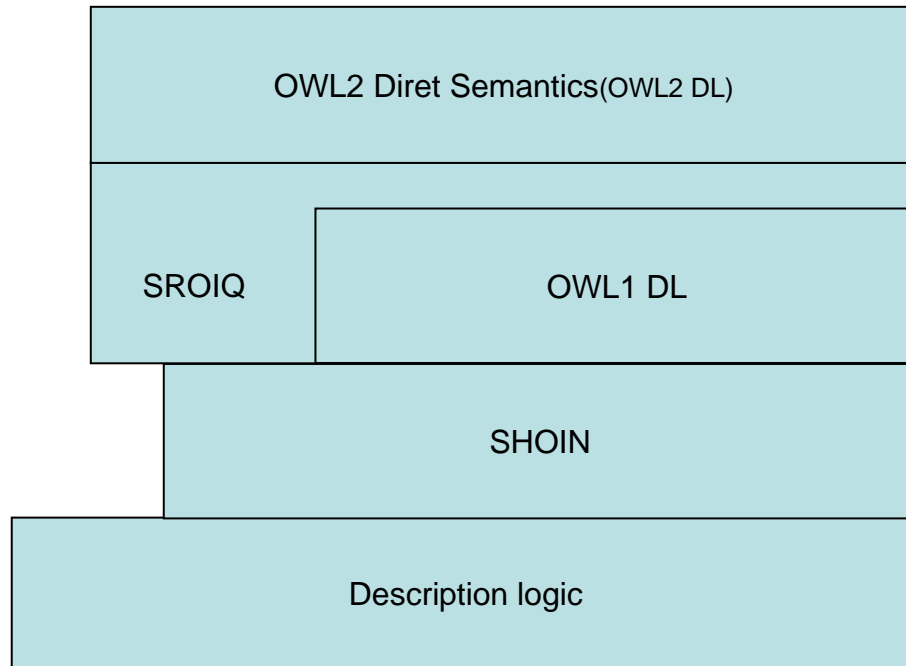
OWL2のセマンティクスとプロファイル(OWL2のサブ言語[構文サブセット])

各プロファイルは、OWLの記述力の違いと異なる計算能力及び/又は実装の利点との間でトレードオフ関係を有する。



- ①.OWL2 EL 大規模オントロジ用(例えば、SNOMED CT及びNCIシソーラス)、推論保証、記述力を犠牲。
- ②.OWL2 QL RDB上での推論器の実装用、比較的軽量なオントロジ(シソーラ/ER/UML等)の分散システム用、Conjunctive Queryが可能。
- ③.OWL2 RL RDFトリプルを直接操作するアプリケーション用、比較的軽量なオントロジの分散システム用、

OWL2 Direct SemanticsとOWL1 DLとの関係



- ①OWL DLの基礎となっているのは、description logic(DL) SHOINである。
- ②OWL DL,SHOINの基礎となっているのは、description logicである。
- ③SHOINを拡張したのがSROIQである。
- ④OWL2 Direct Semanticsの基礎となっているのが、SROIQである。

主な新機能1. キー群(keys)

複数のキーとなるプロパティの種類により、クラスのインディビジュアルをユニークに識別する事を可能にする。

例

HasKey(:Transplantation :donorId :recipientID :ofOrgan)
各々の臓器移植は、臓器提供候補者(ドナー)、移植候補者(レシピエント)及び臓器とにより識別される。

主な新機能2. プロパティチェーン(property chains)

プロパティの組合せを或るプロパティであると定義可能にする。
例えば、プロパティチェーンにより、指の負傷は、手に対する負傷として推論可能にしたい。

- ①指に負傷した。(has-location)
- ②指は手の一部である。(part-of)
- この場合、
- ③手に負傷した。(has-location)

例

SubPropertyOf (ObjectPropertyChain(:locatedIn :partOf) :locatedIn)

もし、xが、yの中に位置(:locatedIn)しており、且つ、yがzの部分(:partOf)ならば、それは、xは、zの中に位置(:locatedIn)する。
例えば、ある疾病がある部分の中に位置している事は、全体の中に位置している事を示す。

利用事例

SNOMED-CTでは、プロパティチェーンを使うことにより、クラスの数
54,380個 → 18,125個
に削減できた。

主な新機能3.

より豊富なデータ型、データレンジ(richer datatypes, data ranges)

データ型

- OWL1は、任意のクラスによって定義される新たなクラスを可能にしているが、新たなデータ型を明示的に定義する方法を提供していない。
- OWL2は、データ型を定義する為に新たな構成要素を提供

例

DatatypeDefinition(:adultAge DatatypeRestriction(xsd:integer minInclusive 18))
成人年齢は、XMLスキーマデータ型xsd:integerの18の下限を用いて定義される。

データレンジ

- OWL1は、複数のクラスを組合せて新たなクラスを定義する事を可能にしているが、複数のデータ型を組合せて新たなデータ型を定義する方法を提供していない。
- 即ち、データ型を組合わせて、データレンジとする事が可能
- データ型の組合わせには、積集合(DataIntersectionOf)、和集合(DataUnionOf)および補集合(DataComplementOf)を使用できる。

例

DatatypeDefinition(:adultAge DatatypeRestriction(xsd:integer minInclusive 18))
DataPropertyRange(:p DataComplementOf(:adultAge))
子供の年齢(:p)のレンジは、成人年齢でない。

主な新機能4.

プロパティ値限定数値制約(qualified cardinality restrictions)

- OWL1では、或るクラスに対して数値制約(`owl:cardinality`、`owl:maxCardinality`、`owl:minCardinality`)を設定して、そのクラスの種類を限定できるが、或るプロパティの値の種類数を指定することはできない。
- OWL2では、`ObjectExactCardinality`、`ObjectMinCardinality`、`ObjectMaxCardinality`、`DataExactCardinality`、`DataMinCardinality`及び`DataMaxCardinality`により、これが可能になっている。

例

```
ClassAssertion( ObjectMaxCardinality( 1 a:hasPet ) a:Taro )  
太郎には、少なくとも1匹のペットを飼っている。
```

主な新機能5. 非対称的、再帰的及びディスジョイントプロパティ (asymmetric, reflexive, and disjoint properties)

- OWL1は、オブジェクトプロパティが対称的又は遷移的であるとの記述を可能にしているが、当該プロパティが再帰的、非再帰的若しくは非対称的であるとの記述はできない。
- OWL2は、或オブジェクトプロパティ表現が、当該プロパティが有する総てのインデビデュアルに於いて、広い範囲で再帰的であると記述する為のReflexiveObjectPropertと言う構成要素を有している。

例

ReflexiveObjectProperty(a:knows)

知る(a:knows)は、反射的オブジェクトプロパティである。

ClassAssertion(a:Person a:Taro)

太郎(a:Taro)は人間(a:Person)である。

太郎が、知る(a:knows)により自分と関係付けられているならば、次のEntailmentが成り立つ。

ObjectPropertyAssertion(a:knows a:Taro a:Taro)

主な新機能6. 注釈機能の強化(enhanced annotation capabilities)

- OWL1は、各オントロジエンティティに付与するラベル又はコメントの様な注釈を可能にしているが、公理に対する注釈は可能ではない。
- 例えば、或る公理を誰が何時主張したかに付いての情報の注釈は、可能でない。
- OWL2は、オントロジ、エンティティ、匿名のインディビデュアル、公理及び注釈自身に対して注釈を付与する事が可能である。

例

```
SubClassOf( Annotation( rdfs:comment "左葉は中葉(:MiddleLobe)を有しないので、肺の中葉(:MiddleLobe)は必然的に右葉(:RightLobe)に属する。" ) :MiddleLobe :RightLobe )
```

コメントは、何故、中葉(:MiddleLobe)が右葉(:RightLobe)のサブクラスか説明する為のサブクラス公理に対する注釈である。

OWL2とOWL1とのクラス語彙の比較表

OWL2		OWL1		
OWL2語彙名	type	語彙名	type	
owl:	owl:Ontology	owl:	owl:Ontology	
owl:AllDifferent	rdfs:Class	AllDifferent		owl1では、単体定義だが、owl2ではrdfs:Resourceのサブクラスとして定義
owl:AllDisjointClasses	rdfs:Class	-		rdfs:Resourceのサブクラスとして定義
owl:AllDisjointProperties	rdfs:Class	-		rdfs:Resourceのサブクラスとして定義
owl:Annotation	rdfs:Class	-		rdfs:Resourceのサブクラスとして定義
owl:AnnotationProperty	rdfs:Class	AnnotationProperty	rdfs:Class	owl1では、rdf:Propertyのサブクラスだが、owl2では、rdfs:Resourceのサブクラスとして定義
owl:AsymmetricProperty	rdfs:Class	-		owl:ObjectPropertyのサブクラスとして定義
owl:Axiom	rdfs:Class	-		rdfs:Resourceのサブクラスとして定義
owl:Class	rdfs:Class	owl:Class	rdfs:Class	どちらもrdfs:Classのサブクラス
owl:DataRange	rdfs:Class	DataRange	rdfs:Class	owl1では、単体定義だが、owl2ではrdfs:Datatypeのサブクラスとして定義
owl:DatatypeProperty	rdfs:Class	DatatypeProperty	rdfs:Class	どちらもrdf:Propertyのサブクラス
owl:DeprecatedClass	rdfs:Class	DeprecatedClass	rdfs:Class	どちらもrdfs:Classのサブクラス
owl:DeprecatedProperty	rdfs:Class	DeprecatedProperty	rdfs:Class	どちらもrdf:Propertyのサブクラス
owl:FunctionalProperty	rdfs:Class	FunctionalProperty	rdfs:Class	どちらもrdf:Propertyのサブクラス
owl:InverseFunctionalProperty	rdfs:Class	InverseFunctionalProperty	rdfs:Class	どちらもowl:ObjectPropertyのサブクラス
owl:IrreflexiveProperty	rdfs:Class	-		owl:ObjectPropertyのサブクラス
owl:NamedIndividual	rdfs:Class	-		owl:Thingのサブクラス
owl:NegativePropertyAssertion	rdfs:Class	-		rdfs:Resourceのサブクラス
owl:Nothing	rdfs:Class	Nothing	owl:Class	owl1では、owl:Thingの補集合と定義、owl2では、owl:Thingのサブクラスとして定義
owl:ObjectProperty	rdfs:Class	ObjectProperty	rdfs:Class	どちらもrdf:Propertyのサブクラス
owl:Ontology	rdfs:Class	Ontology	rdfs:Class	owl1では、単体定義だが、owl2ではrdfs:Resourceのサブクラスとして定義
owl:OntologyProperty	rdfs:Class	OntologyProperty	rdfs:Class	どちらもrdf:Propertyのサブクラス
owl:ReflexiveProperty	rdfs:Class	-		owl:ObjectPropertyのサブクラス
owl:Restriction	rdfs:Class	Restriction	rdfs:Class	どちらもowl:Classのサブクラス
owl:SymmetricProperty	rdfs:Class	SymmetricProperty	rdfs:Class	どちらもowl:ObjectPropertyのサブクラス
owl:Thing	owl:Class	owl:Thing	owl:Class	owl1では、owl:Nothingとowl:Nothingの補集合との和、owl2では単体定義
owl:TransitiveProperty	rdfs:Class	TransitiveProperty	rdfs:Class	どちらもowl:ObjectPropertyのサブクラス

OWL2とOWL1とのプロパティ語彙の比較表(1/2)

OWL2		OWL1		
OWL2語彙名	type	語彙名	type	
owl:allValuesFrom	rdf:Property	allValuesFrom	rdf:Property	どちらもドメインがowl:Restrictionで、レンジがrdfs:Class
owl:annotatedProperty	rdf:Property	-		ドメイン、レンジ共にrdfs:Resource
owl:annotatedSource	rdf:Property	-		ドメイン、レンジ共にrdfs:Resource
owl:annotatedTarget	rdf:Property	-		ドメイン、レンジ共にrdfs:Resource
owl:assertionProperty	rdf:Property	-		ドメインがowl:NegativePropertyAssertionで、レンジはrdf:Property
owl:backwardCompatibleWith	owl:AnnotationProperty owl:OntologyProperty	backwardCompatibleWith	rdf:Property owl:OntologyProperty	どちらもドメイン、レンジ共にowl:Ontology
owl:bottomDataProperty	owl:DatatypeProperty	-		ドメインがowl:Thingで、レンジはrdfs:Literal
owl:bottomObjectProperty	owl:ObjectProperty	-		ドメイン、レンジ共にowl:Thing
owl:cardinality	rdf:Property	cardinality	rdf:Property	どちらもドメインがowl:Restrictionで、レンジはxsd:nonNegativeInteger
owl:complementOf	rdf:Property	complementOf	rdf:Property	どちらもドメイン、レンジ共にowl:Class
owl:datatypeComplementOf	rdf:Property	-		ドメイン、レンジ共にrdfs:Datatype
owl:deprecated	owl:AnnotationProperty	-		ドメイン、レンジ共にrdfs:Resource
owl:differentFrom	rdf:Property	differentFrom	rdf:Property	どちらもドメイン、レンジ共にowl:Thing
owl:disjointUnionOf	rdf:Property	-		ドメインがowl:Classで、レンジはrdf:List、owl1では導入されなかったが、Damiから導入
owl:disjointWith	rdf:Property	disjointWith	rdf:Property	どちらもドメイン、レンジ共にowl:Class
owl:distinctMembers	rdf:Property	distinctMembers	rdf:Property	どちらもドメインがowl:AllDifferentで、レンジはrdf:List
owl:equivalentClass	rdf:Property	equivalentClass	rdf:Property	owl1では、rdfs:subClassOfのサブクラスとしての定義がある。 ? どちらもドメイン、レンジ共にowl:Class
owl:equivalentProperty	rdf:Property	equivalentProperty	rdf:Property	owl1では、rdfs:subPropertyOfのサブプロパティであるとの定義があるが、ドメインレンジの定義は無く、 owl2では、サブプロパティ定義は無くドメイン、レンジ共にrdf:Propertyであるとの定義がある
owl:hasKey	rdf:Property	-		ドメインがowl:Classで、レンジはrdf:List
owl:hasSelf	rdf:Property	-		ドメインがowl:Restrictionで、レンジはrdf:Resource
owl:hasValue	rdf:Property	hasValue	rdf:Property	owl1では、ドメインがowl:Restrictionの定義のみで、レンジの定義は無く、 owl2では、ドメインがowl:Restrictionで、レンジは、rdf:Resource
owl:imports	owl:OntologyProperty	imports	owl:OntologyProperty	どちらもドメイン、レンジ共にowl:Ontology
owl:incompatibleWith	owl:AnnotationProperty owl:OntologyProperty	incompatibleWith	rdf:Property owl:OntologyProperty	どちらもドメイン、レンジ共にowl:Ontology

OWL2とOWL1とのプロパティ語彙の比較表(2/2)

owl:intersectionOf	rdf:Property	intersectionOf	rdf:Property	ドメインがowl:Classで、レンジはrdf:List
owl:inverseOf	rdf:Property	inverseOf	rdf:Property	どちらもドメイン、レンジ共にowl:ObjectProperty
owl:maxCardinality	rdf:Property	maxCardinality	rdf:Property	どちらもドメインがowl:Restrictionで、レンジはxsd:nonNegativeInteger
owl:maxQualifiedCardinality	rdf:Property	-	-	ドメインがowl:Restrictionで、レンジはxsd:nonNegativeInteger
owl:members	rdf:Property	-	-	ドメインがrdfs:Resourceで、レンジはrdf:List
owl:minCardinality	rdf:Property	minCardinality	rdf:Property	どちらもドメインがowl:Restrictionで、レンジはxsd:nonNegativeInteger
owl:minQualifiedCardinality	rdf:Property	-	-	ドメインがowl:Restrictionで、レンジはxsd:nonNegativeInteger
owl:onClass	rdf:Property	-	-	ドメインがowl:Restrictionで、レンジはowl:Class
owl:onDataRange	rdf:Property	-	-	ドメインがowl:Restrictionで、レンジはrdfs:Datatype
owl:onDatatype	rdf:Property	-	-	ドメイン、レンジ共にrdfs:Datatype
owl:oneOf	rdf:Property	oneOf	rdf:Property	どちらもドメインがrdfs:Classで、レンジがrdf:List
owl:onProperties	rdf:Property	-	-	ドメインがowl:Restrictionで、レンジはrdf:List
owl:onProperty	rdf:Property	onProperty	rdf:Property	どちらもドメインがowl:Restrictionで、レンジはrdf:Property
owl:priorVersion	owl:AnnotationProperty owl:OntologyProperty	priorVersion	rdf:Property owl:OntologyProperty	どちらもドメイン、レンジ共にowl:Ontology
owl:propertyChainAxiom	rdf:Property	-	-	ドメインがowl:ObjectPropertyで、レンジはrdf:List
owl:propertyDisjointWith	rdf:Property	-	-	ドメイン、レンジ共にrdf:Property
owl:qualifiedCardinality	rdf:Property	-	-	ドメインがowl:Restrictionで、レンジはxsd:nonNegativeInteger
owl:sameAs	rdf:Property	sameAs	rdf:Property	どちらもドメイン、レンジ共にowl:Thing
owl:someValuesFrom	rdf:Property	someValuesFrom	rdf:Property	どちらもドメインがowl:Restrictionで、レンジがrdfs:Class
owl:sourceIndividual	rdf:Property	-	-	ドメインがowl:NegativePropertyAssertionで、レンジはowl:Thing
owl:targetIndividual	rdf:Property	-	-	ドメインがowl:NegativePropertyAssertionで、レンジはowl:Thing
owl:targetValue	rdf:Property	-	-	ドメインがowl:NegativePropertyAssertionで、レンジはrdfs:Literal
owl:topDataProperty	owl:DatatypeProperty	-	-	ドメインがowl:Thingで、レンジはrdfs:Literal
owl:topObjectProperty	owl:ObjectProperty	-	-	ドメイン、レンジ共にowl:Thing
owl:unionOf	rdf:Property	unionOf	rdf:Property	owl1では、ドメインがowl:Classで、レンジは、rdf:List owl2では、ドメインがrdfs:Classで、レンジは、rdf:List
owl:versionInfo	owl:AnnotationProperty	versionInfo	rdf:Property owl:AnnotationProperty	owl1では、ドメインおよびレンジ定義は無く、 owl2では、ドメイン、レンジ共にrdfs:Resource
owl:versionIRI	owl:OntologyProperty	-	-	ドメイン、レンジ共にowl:Ontology
owl:withRestrictions	rdf:Property	-	-	ドメインがrdfs:Datatypeで、レンジはrdf:List