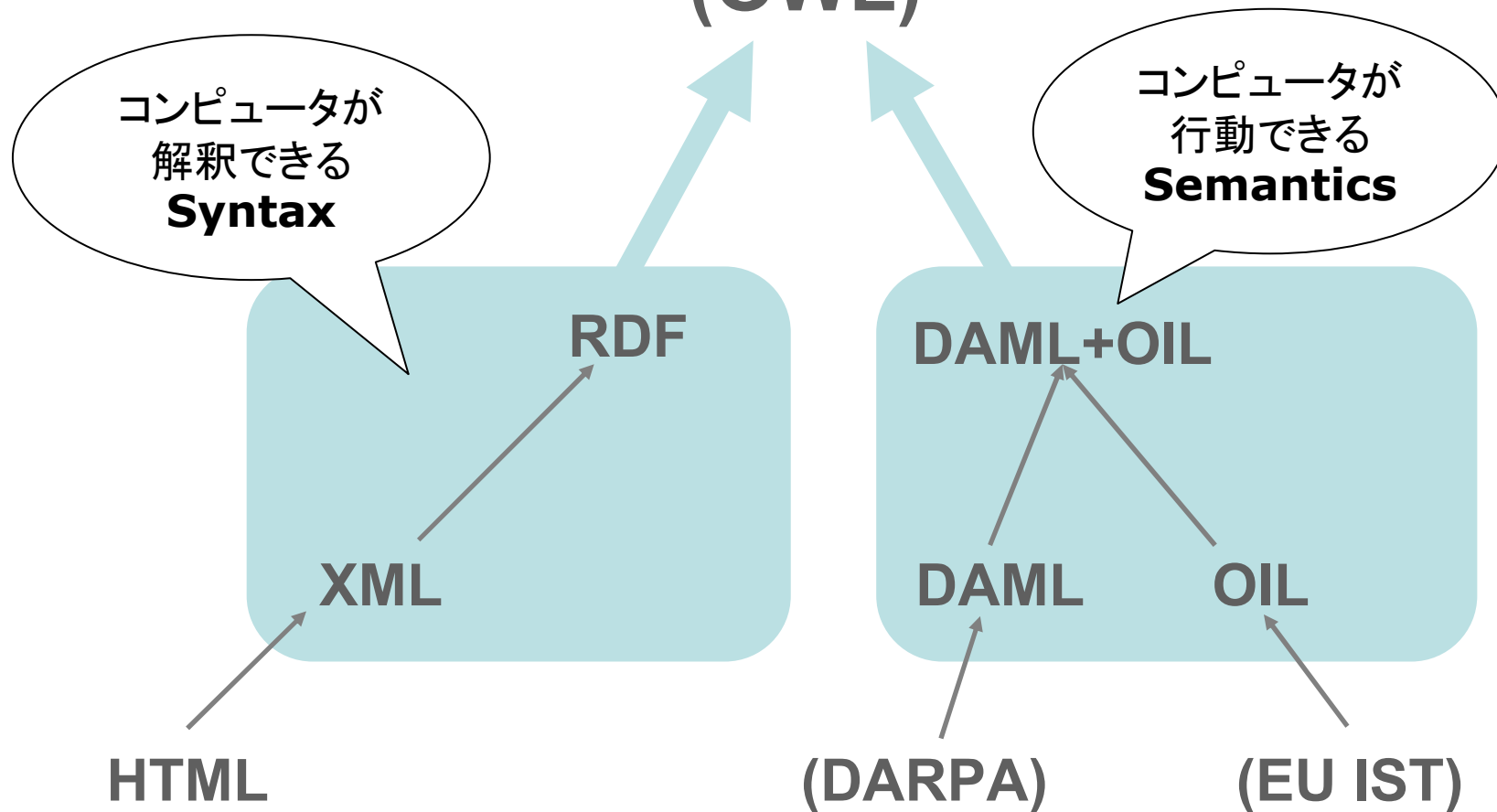


ウェブ・オントロジー言語 OWL

大森健児

OWLの歴史

Web Ontology Language (OWL)



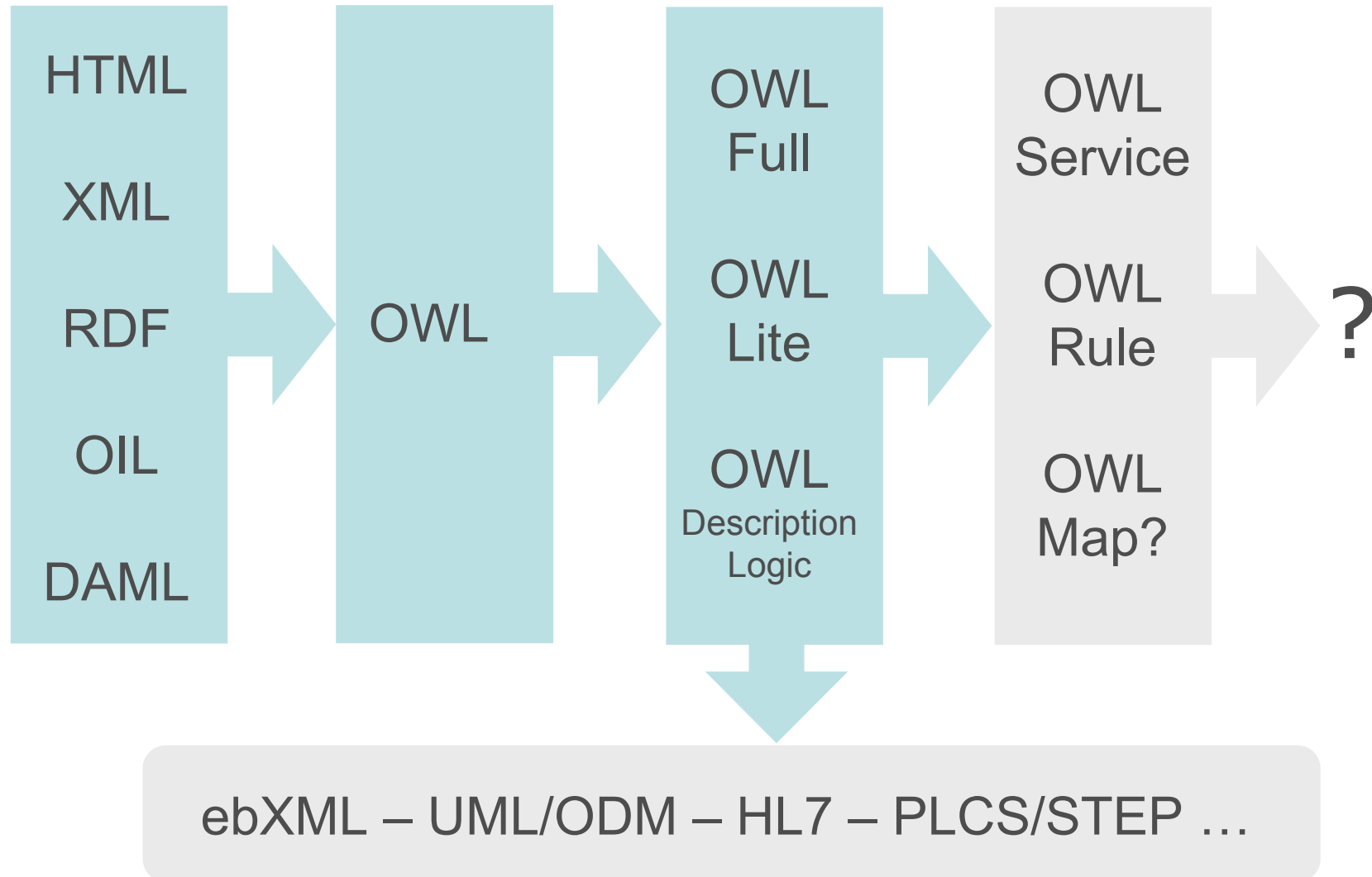
OWLとは何か

- World Wide Web Consortium のセマンティックウェブの活動の中心
- ウェブに親和性の高い知識モデリング言語を目指したこれまでの活動の継承
- W3CのWeb Ontology Working Groupは知識表現分野での”who's who”

なぜOWLか

- セマンティックウェブの応用分野
 - ポータルウェブサイト(情報アーキテクチャ)
 - マルティメディアデジタル図書館(豊富なメタデータ)
 - ウェブサービス(相互接続、自動化)
 - 設計ドキュメント(複雑、相互にリンク)
- 可能性
 - オントロジーの共有、進化、相互接続
 - 矛盾の発見
 - 表現力 VS 拡張性
 - 標準規格準拠

OWLのロードマップ



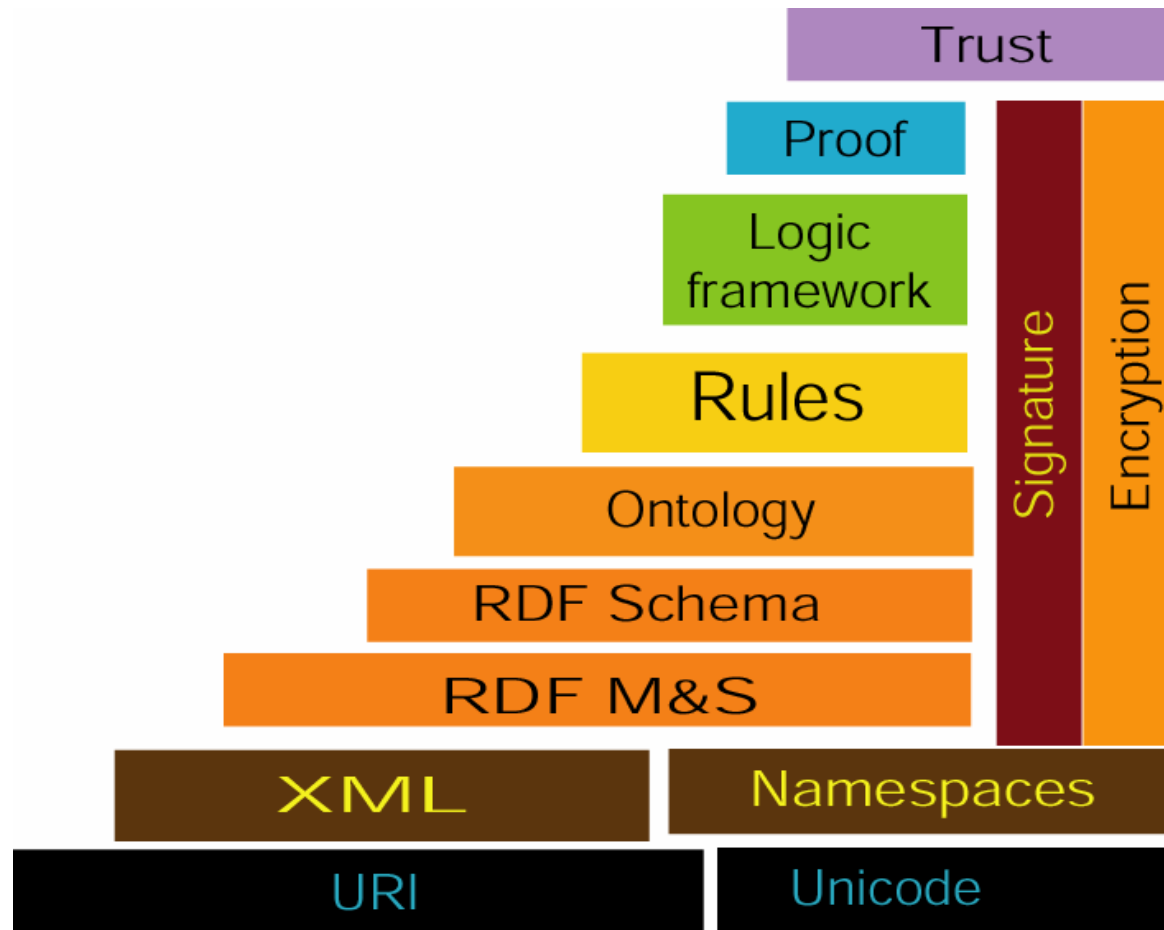
XML, RDF & OWL

- XML: 汎用シンタックス
- XML Schema: XML文章の構造定義
- RDF: リソースオブジェクトのデータモデル
- RDF Schema: RDFのクラス、プロパティを定義するための基本語彙とそれぞれの階層
- OWL: 拡張言語
 - Cardinality
 - Equality
 - Relationships between classes
 - Characteristics of properties

OWL sublanguages

- OWL Lite
 - RDFに機能拡張
 - クラス階層と簡単な制約
- OWL DL
 - Description logic theoretical properties
 - 記述論理をベースにタイプの区別を厳密にして決定可能性を確保
- OWL Full
 - 計算の可能性は保障せず
 - クラスを個体とみなすことができるなど実用的なオントロジー構築を念頭

セマンティックウェブのアーキテクチャ

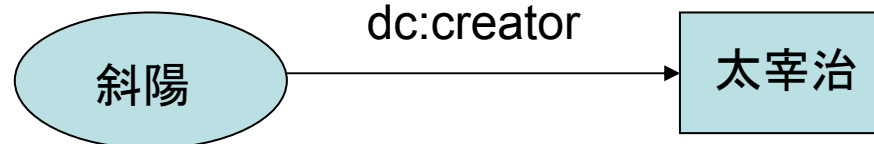


RDFのトリプル表現

- RDFはリソースの関係を主語、述語、目的語という3つの要素で表現
 - 例: 斜陽は太宰治が作者です。

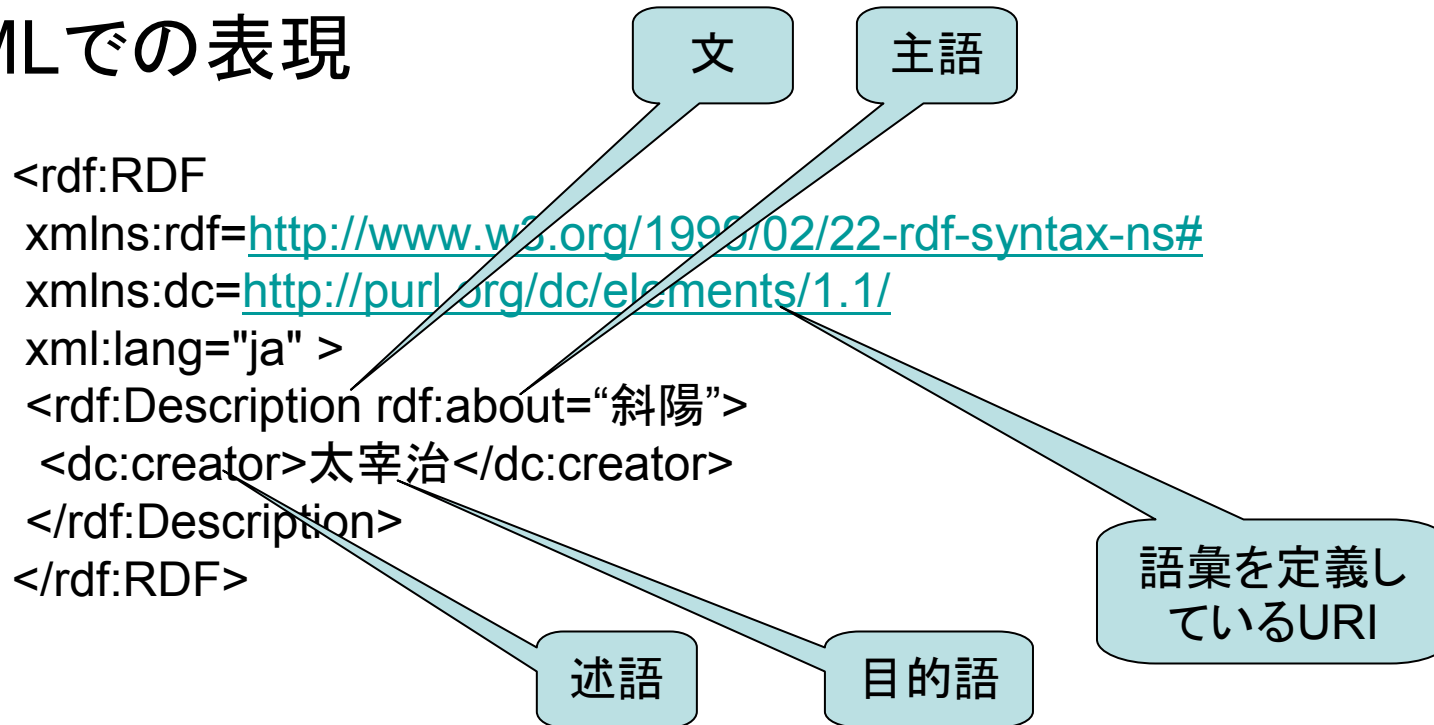
主語	リソース	斜陽
述語	プロパティ	作者
目的語	プロパティの値	太宰治

- プロパティはDublin Coreの語彙を用いてdc:creatorで表すと主語から目的語に向かう有向グラフは次のようになる。



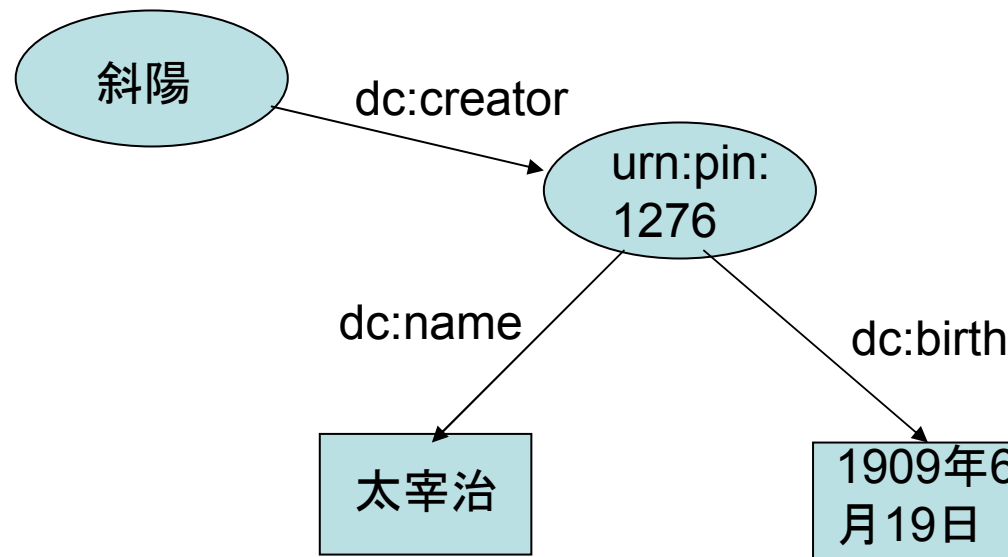
RDFのトリプル表現(続き)

- XMLでの表現



構造化モデル

- RDFの文の目的語にはリソースを指定できる。
- リソースはそれを主語とした文とすることができる。
 - 斜陽の作者はurn:pin:1276というURIで参照される人物で、その人物の名前は太宰治で、生年月日は1909年6月19日です。



構造化モデル(続き)

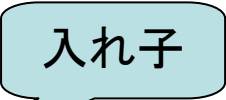
XMLでの表現

```
<rdf:RDF
  xmlns:rdf=http://www.w3.org/1999/02/22-rdf-syntax-ns#
  xmlns:dc=http://purl.org/dc/elements/1.1/
  xml:lang="ja" >
  <rdf:Description rdf:about="斜陽">
    <dc:creator rdf:resource="urn:pin:1276"/>
  </rdf:Description>
  <rdf:Description rdf:about="urn:pin:1276">
    <dc:name>太宰治</dc:name>
    <dc:birth>1909年6月16日</dc:birth>
  </rdf:Description>
</rdf:RDF>
```

構造化モデル(続き)

- XMLでの入れ子表現

```
<rdf:RDF
  xmlns:rdf=http://www.w3.org/1999/02/22-rdf-syntax-ns#
  xmlns:dc=http://purl.org/dc/elements/1.1/
  xml:lang="ja" >
  <rdf:Description rdf:about="斜陽">
    <dc:creator>
      </rdf:Description>
    <rdf:Description rdf:about="urn:pin:1276">
      <dc:name>太宰治</dc:name>
      <dc:birth>1909年6月16日</dc:birth>
    </rdf:Description>
  </dc:creator>
</rdf:RDF>
```



入れ子

構造化モデル(続き)

- リソースの無名化

- 斜陽の作者は名前は太宰治で、生年月日は1909年6月19日です。

```
<rdf:RDF
  xmlns:rdf=http://www.w3.org/1999/02/22-rdf-syntax-ns#
  xmlns:dc=http://purl.org/dc/elements/1.1/
  xml:lang="ja" >
  <rdf:Description rdf:about="斜陽">
    <dc:creator>
      </rdf:Description>
    <rdf:Alt>
      <dc:name>太宰治</dc:name>
      <dc:birth>1909年6月16日</dc:birth>
    </rdf:Alt>
  </dc:creator>
</rdf:RDF>
```

無名化

OWLの基本構成

- オントロジー・ヘッダー

ヘッダー

```
<owl:Ontology rdf:about="">  
<owl:versionInfo>webont.html, v.0.9; 2002-08-25 Exp</owl:versionInfo>  
<owl:imports rdf:resource="http://www.k.hosei.ac.jp/~ohmori"/owl>  
<dc:creator>Kenji Ohmori</dc:creator>  
</owl:Ontology>  
...  
<owl:AnnotationProperty rdf:about="http://purl.org/dc/elements/1.1/creator"/>  
...
```

メタデータの
埋め込み

目的語として示されたオ
ントロジーのグラフを取り
込んで、主語オントロ
ジーのグラフに加える

クラス公理

構成要素	意味
<code>rdfs:subClassOf</code>	参照クラスのサブクラス。必要条件(部分公理)を形成
<code>owl:disjointWith</code>	参照クラスとは分離(共通インスタンスがない)。必要条件(部分公理)を形成
<code>owl:equivalentClasses</code>	参照クラスと同じインスタンスを持つ。必要十分条件(完全公理)を形成
<code>owl:oneOf</code>	列挙されたクラス表現のうちひとつのインスタンスを持つ(OWL Liteでは使えない)。必要十分条件(完全公理)を形成
クラス式の組み合わせ	クラス名、クラスの列挙、プロパティの制約条件、もしくはこれらの論理的組み合わせと同等である。匿名クラス(<code>owl:Restriction</code>)をつくり、上記のプロパティによって <code>owl:Class</code> に結びつけて公理を形成

クラス公理(例)

- 「雄」というクラスは「動物」のサブクラスで「雌」とは分離している

```
<owl:Class rdf:ID="Male">  
  <rdfs:subClassOf rdf:resource="#Animal"/>  
  <owl:disjointWith rdf:resource="#Female"/>  
</owl:Class>
```

#は文章内の他の
場所でrdf:IDで定
義されているリ
ソースへの参照

プロパティの制約条件

	制約	意味
値の範囲	owl:allValuesFrom	全ての値は、参照クラスあるいはデータタイプ値のインスタンスである
	owl:someValuesFrom	ひとつの値は、参照クラスあるいはデータタイプ値のインスタンスである
	owl:hasValue	主語クラスの全てのインスタンスについて、Pの値のうち、少なくともひとつは参照クラスあるいはデータタイプ値に等しい
出現回数の制約	owl:minCardinality	最小出現回数
	owl:maxCardinality	最大出現回数
	owl:cardinality	指定された回数だけ必ず出現する。

プロパティの制約条件(例)

- 「人」というクラスは「動物」のサブクラスで、「本名」というプロパティの値は1つだけである

```
<owl:Class rdf:ID="Person">  
<rdfs:subClassOf rdf:resource="#Animal"/>  
<rdfs:subClassOf> <owl:Restriction>  
<owl:onProperty  
rdf:resource="#hasRealName"/>  
<owl:cardinality>1</owl:cardinality>  
</owl:Restriction> </rdfs:subClassOf>  
</owl:Class>
```

クラスの論理的組合せ

プロパティ	意味
owl:intersectionOf	一連のクラスのインスタンスの共通部分に一致
owl:unionOf	一連のクラスのインスタンスの和集合に一致
owl:complementOf	一連のクラスインスタンス集合に属さない全てのオブジェクトに一致する

クラスの論理的組合せ(例)

- 「好物」というクラスを「にんじんとピーマン以外」

```
<owl:Class rdf:ID="FavoriteFoods">  
  <owl:complementOf>  
    <owl:Class>  
      <owl:unionOf rdf:parseType="Collection">  
        <owl:Class rdf:about="#Carrot"/>  
        <owl:Class rdf:about="#GreenPepper"/>  
      </owl:unionOf>  
    </owl:Class>  
  </owl:complementOf>  
</owl:Class>
```

プロパティ公理

構成要素	意味
rdfs:subPropertyOf	参照プロパティのサブプロパティ
rdfs:range	目的語は、参照クラスのインスタンス
rdfs:domain	主語は、参照クラスのインスタンスである
owl:equivalentProperty	参照プロパティと同じインスタンス(主語、目的語リソースの組み合わせ)を持つ
owl:inverseOf	参照プロパティと反対の関係を表現

クラス公理(例)

- 「子供がいる」というプロパティは「親がいる」の反対の関係を示しますから、次のように定義

```
<owl:ObjectProperty rdf:ID="hasChild">  
<owl:inverseOf rdf:resource="#hasParent"/>  
</owl:ObjectProperty>
```

プロパティの論理性質

クラス名URI	意味
owl:TransitiveProperty	「子孫」プロパティのように、 $P(x,y)$ と $P(y,z)$ が真なら $P(x,z)$ も真であるという具合に関係が推移していくプロパティ
owl:SymmetricProperty	「夫婦」プロパティのように、 $P(x,y) \Leftrightarrow P(y,x)$ が成り立つプロパティ
owl:FunctionalProperty	「本名」のように値が唯一に定まるプロパティ
owl:InverseFunctionalProperty	「ISBN」のように、その値から主語が特定できるようなプロパティ

プロパティの論理性質(例)

- 「子孫がいる」というプロパティは推移プロパティでかつ「祖先がいる」の反対の関係

```
<owl:TransitiveProperty rdf:ID="hasOffspring">  
  <owl:inverseOf rdf:resource="#hasAncestor"/>  
</owl:TransitiveProperty>
```

個体による事実の記述

構成要素	意味
owl:sameAs	2つの個体が同一
owl:differentFrom	2つの個体が別物
owl:AllDifferent	列挙した一連の個体が互いに別物
rdf:type	個体は参照クラスのインスタンス
その他のプロパティ要素	個体のプロパティを示す

クラス公理(例)

- 個体を記述する例

```
<ex:Novelist rdf:ID="Lewis_Carroll">  
  <owl:sameAs rdf:resource="#Charles_Lutwidge_Dodgson"/>  
  <ex:isAuthorOf rdf:resource="#Alice_in_Wonderland"/>  
</ex:Novelist>
```