

「セマンティックWeb技術に触れてみよう！
RDF/SPARQLハンズオン勉強会」
～ オープンデータからLinkedDataまでを総ざらい～

LOD技術と オープンデータ

2013/12/21
コンテキスト・コンピューティング研究部会
サブリーダー 小林 茂

- オープンデータについて
 - オープンガバメント
- セマンティックWeb技術(RDF, SPARQL)
 - RDFとは
 - RDFの表現形式: タートル, RDFa, マイクロデータ
 - RDFグラフへの問い合わせ: SPARQL
 - 利用環境(SPARQL Timeliner, SparqlEPCU, LinkData)
- LODについて
 - Linked Data, LOD, LCD, LODクラウド図, ...
 - 事例

オープンデータとは

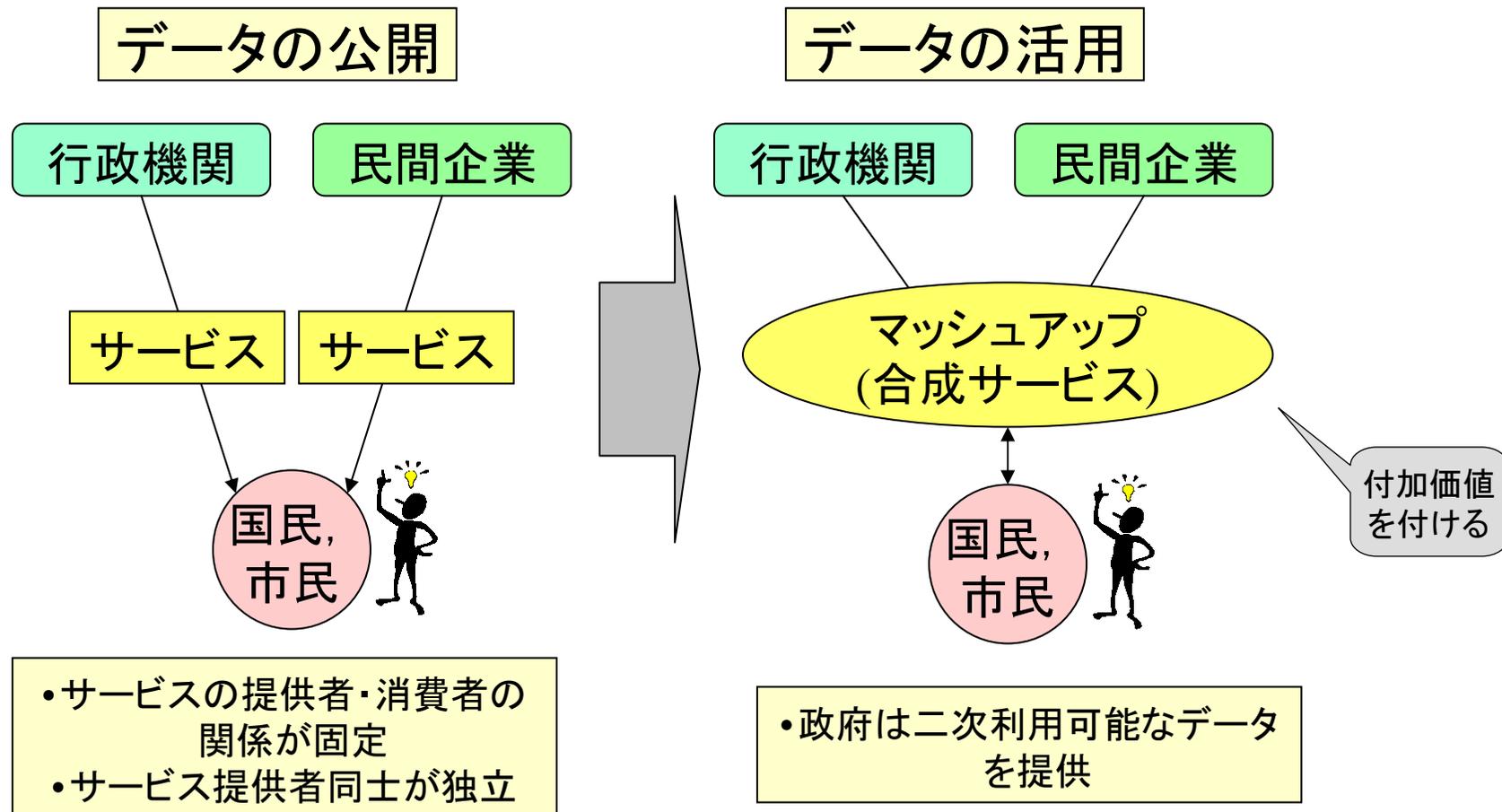
- オープン(ライセンス)であること
 - 誰でもが使えること
- データであること
 - 二次利用が可能なデータであること
- 今までは,
 - 公開されていなかった。
 - 公開されていてもWebページであるか, PDFであり, 使えなかった。(ひどい時は画像)
 - 人が見ると分かるが, データとして使えなかった。
 - 再利用するための権利が不明であった

オープンガバメントとは

- 開かれた政府とは
 - － インターネットの双方向性等を活用することで、積極的な政府情報の公開や行政への市民参加を促進する取組
(オープンガバメントラボによる)
- キーワード
 - － インターネット活用
 - － 官民協働の公共活動
 - － 政府への国民参加
 - － 政府の透明性向上
- オープンデータは、オープンガバメントのひとつの要素
- 新しいサービス、ビジネスの創出や産業活性化が促進されることを期待

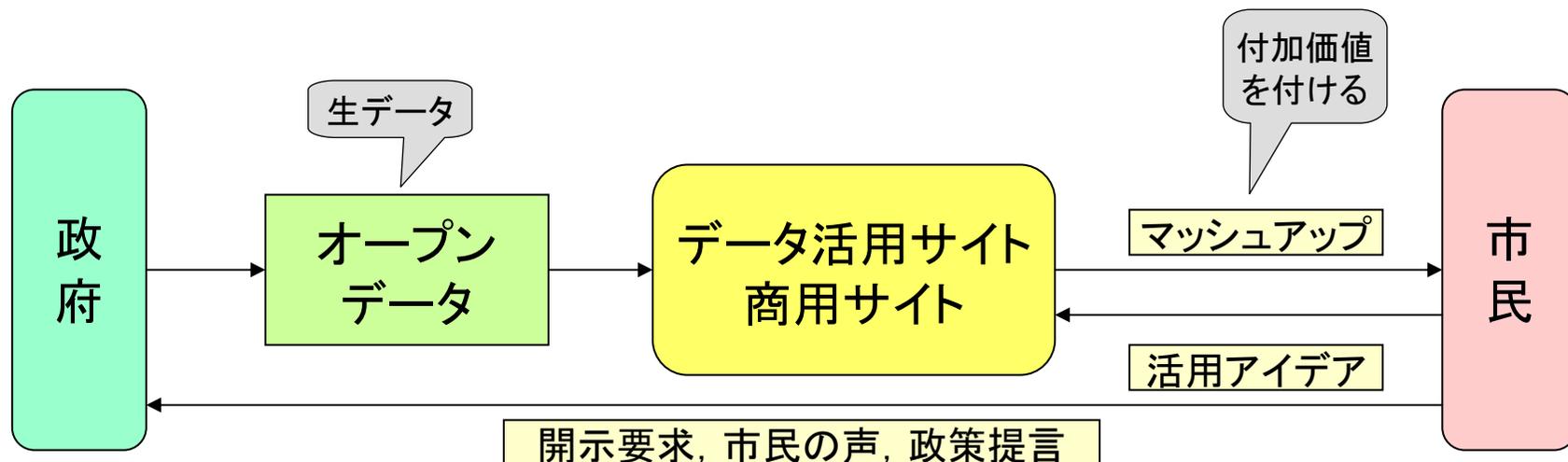
出典：<http://www.openlabs.go.jp/labs>

オープン化の流れ



何ができるか

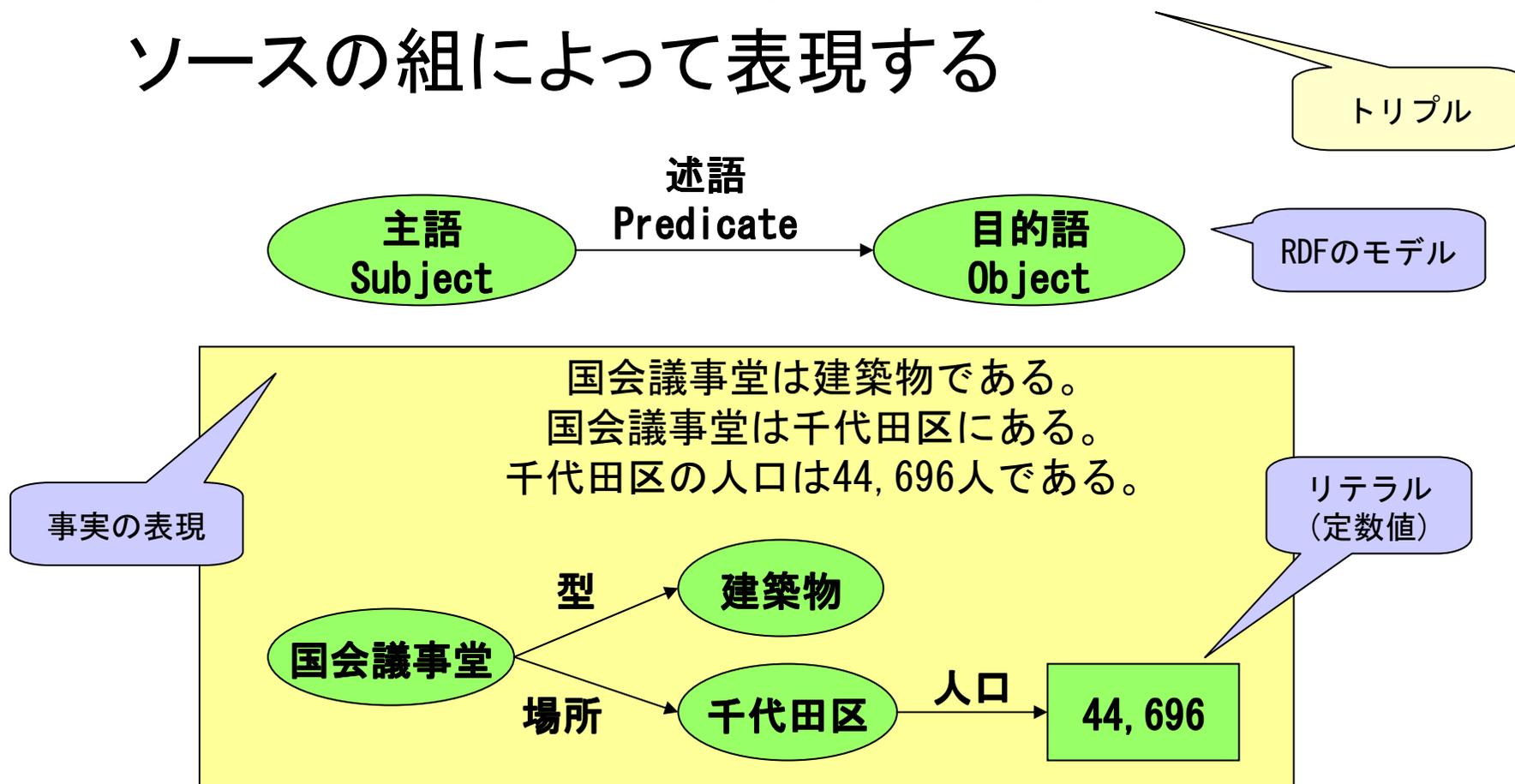
- 市民活動によるデータの加工・配信
 - 政府は生データを提供
- 民間による情報の活用・商用利用
 - マッシュアップにより付加価値を提供
- 行政へのフィードバック・官民協業



- オープンデータについて
 - オープンガバメント
- セマンティックWeb技術(RDF, SPARQL)
 - RDFとは
 - RDFの表現形式: タートル, RDFa, マイクロデータ
 - RDFグラフへの問い合わせ: SPARQL
 - 利用環境(SPARQL Timeliner, SparqlEPCU, LinkData)
- LODについて
 - Linked Data, LOD, LCD, LODクラウド図, ...
 - 事例

RDFとは

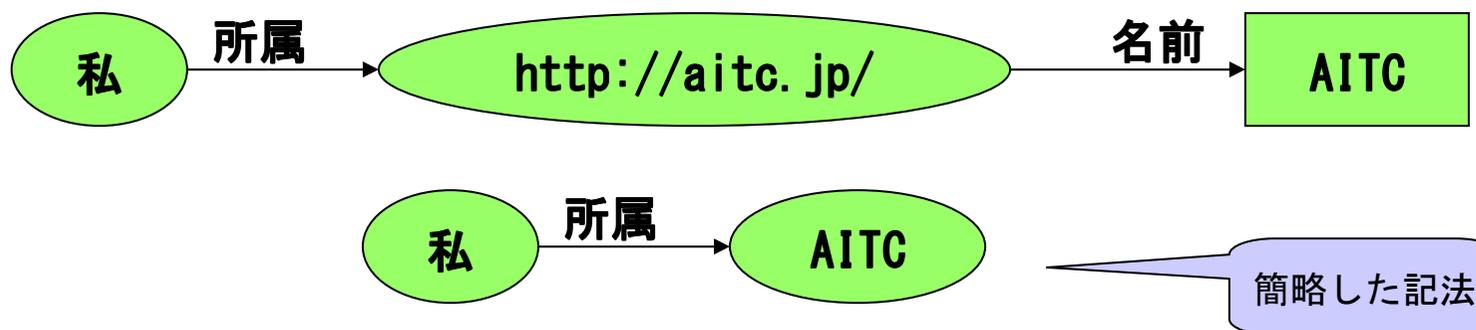
- Resource Description **Framework**
- メタデータを主語, 述語, 目的語の3つのリソースの組によって表現する



URI:リソースを特定

- リソースは, Web上で表現されるあらゆる物
 - 人物, 文書, 建築物, イベント, 属性名, ...
- RDFでは, リソースを特定するために, URI(Universal Resource Identifier)を適用
 - URIで識別されるものがリソース
 - 私の所属 → `http://aitc.jp/`

IRI 也可

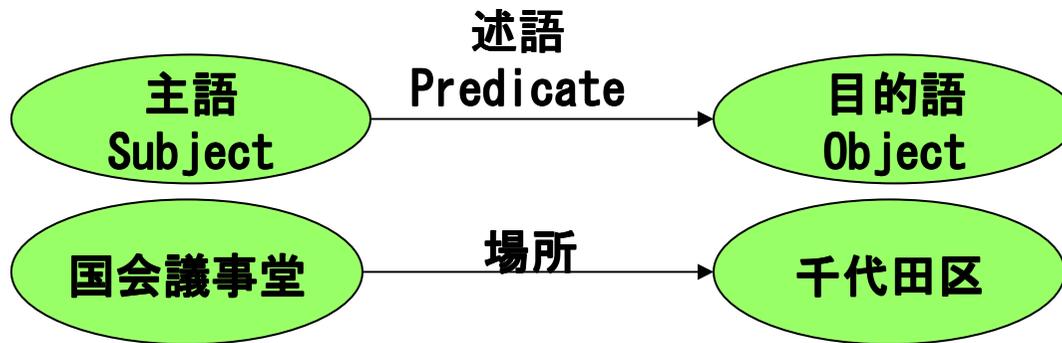


RDFのXML表現

リソース
国会議事堂

メタデータ
場所:千代田区

「場所」,
「千代田区」
もリソース



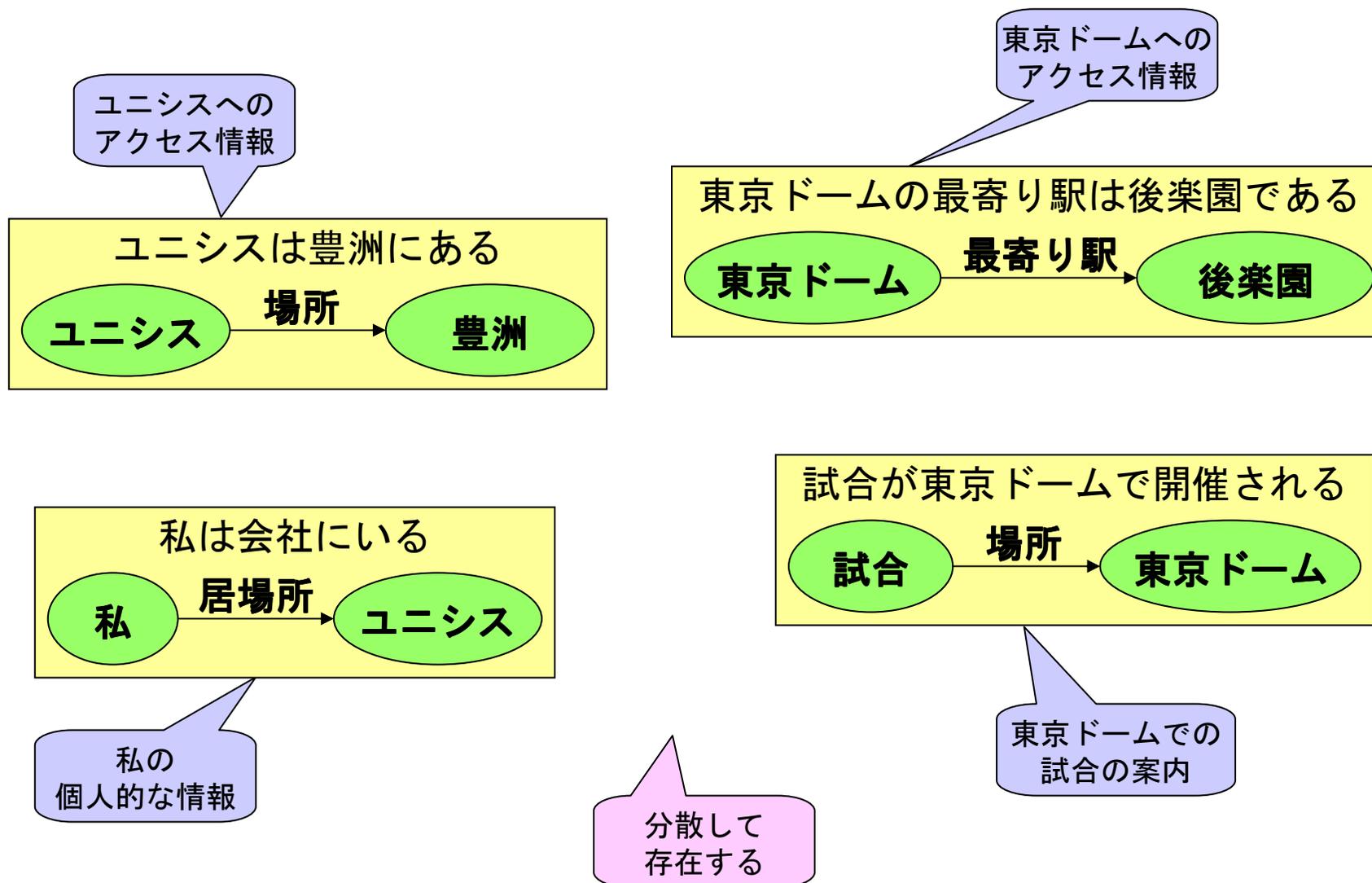
RDFであることを示すコンテナ

```
<rdf:RDF xmlns:rdf=http://www.w3.org/1999/02/22-rdf-syntax-ns# >  
<rdf:Description rdf:about="国会議事堂を示すURI">  
  <場所>千代田区</場所>  
</rdf:Description>  
</rdf:RDF>
```

```
<rdf:RDF xmlns:rdf=http://www.w3.org/1999/02/22-rdf-syntax-ns# >  
<rdf:Description rdf:about="国会議事堂を示すURI" 場所="千代田区">  
</rdf:Description>  
</rdf:RDF>
```

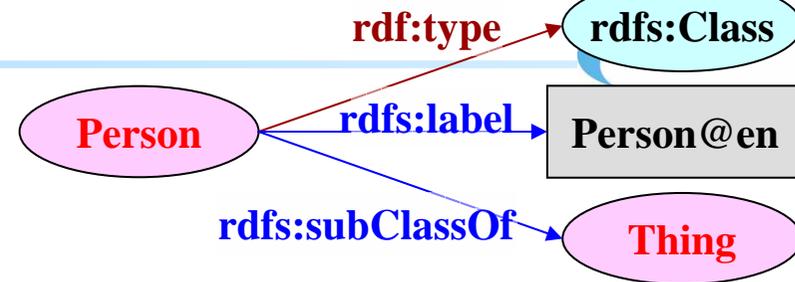
属性で記述

RDF: 事実を表現する



RDFの表現形式

schema.org で定義する “**Person**” の定義



nTriple

```
<http://schema.org/Person> <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>  
  <http://www.w3.org/2000/01/rdf-schema#Class> .  
<http://schema.org/Person> <http://www.w3.org/2000/01/rdf-schema#label> "Person"@en  
<http://schema.org/Person> <http://www.w3.org/2000/01/rdf-schema#subClassOf>  
  <http://schema.org/Thing> .
```

Turtle

```
schema:Person a rdfs:Class;  
  rdfs:label "Person"@en;  
  rdfs:subClassOf schema:Thing  
  .
```

簡略形URIを3つ並べる。
同一な主語を省略可。
ピリオドで終了

<URI>を3つ並べ
ピリオドで終了

RDF/XML

```
<rdf:Description rdf:about="http://schema.org/Person">  
  <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Class"/>  
  <rdfs:label xml:lang="en">Person</rdfs:label>  
  <rdfs:subClassOf rdf:resource="http://schema.org/Thing"/>  
</rdf:Description>
```

XML形式。
少し複雑

RDFa (第2版)

- W3Cで標準化作業中。作業案 (2013-09-13)
 - HTML Microdata
<http://www.w3.org/TR/microdata/>
- XML, HTML文書の中にメタデータを埋め込むための仕様
- 属性として, 13種類を定義
 - about 主語を指定する
 - typeof 型を参照する
 - property 述語(プロパティ)を指定する
 - resource 他のリソースを指定する
 - content リテラルを指定する
 - datatype データ型を参照する
 - prefix プレフィックスを指定する
 - vocab 語彙(URI)を参照する
 - inlist RDFのリストを指定する
 - rel, rev, href, src HTMLと同様
- 属性値は, コンテンツそのものを利用可能

```
<span  
  about="主語のURI"  
  typeof="型のURI"  
  property="プロパティのURI"  
>内容そのもの</span>
```

```
<span  
  about="urn:ISBN:978-4-7649-0427-9"  
  typeof="http://purl.org/ontology/bibo/Book"  
  property="http://purl.org/dc/terms/description"  
>Linked Data</span>
```

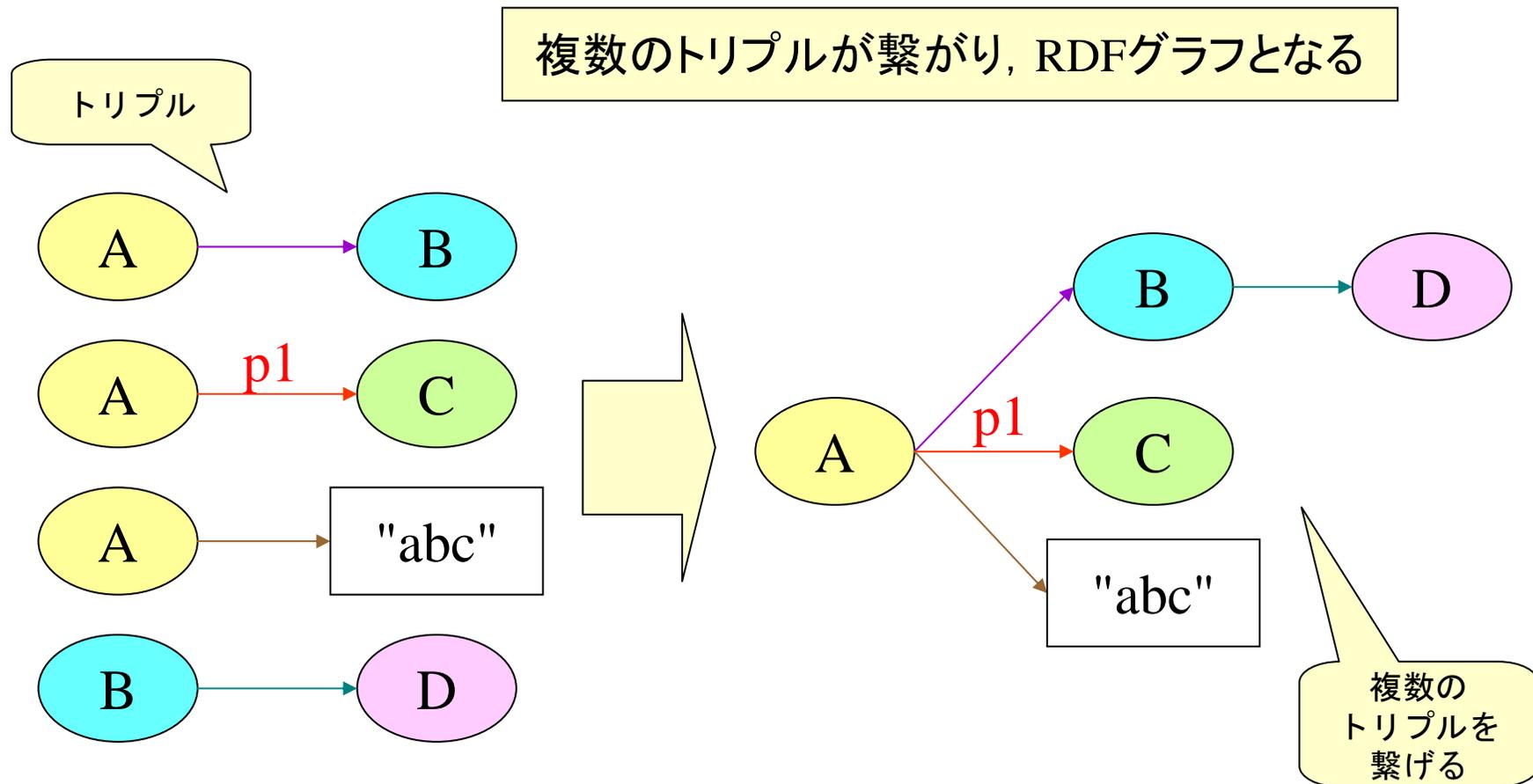
マイクロデータ

- W3Cで標準化作業中。作業案 (2012-10-25)
 - HTML Microdata
<http://www.w3.org/TR/microdata/>
- HTML文書の中にメタデータを埋め込むための仕様
- HTML5とは独立した仕様だがHTML5の一部と言えよう
- 属性として、5種類を定義
 - itemscope アイテムの範囲を設定する
 - itemprop プロパティ名を指定する
 - itemtype アイテムの型を指定する
 - itemref 他のアイテムを参照する
 - itemid アイテムに識別子を設定する
- 属性値は、コンテンツそのものや既存のHTMLタグを利用
- アイテムを得るためのDOM APIが定義されている
 - `document.getItems([types])` アイテムを得る
 - `element.properties` プロパティ名を得る
 - `element.itemValue [= value]` プロパティ値を得る

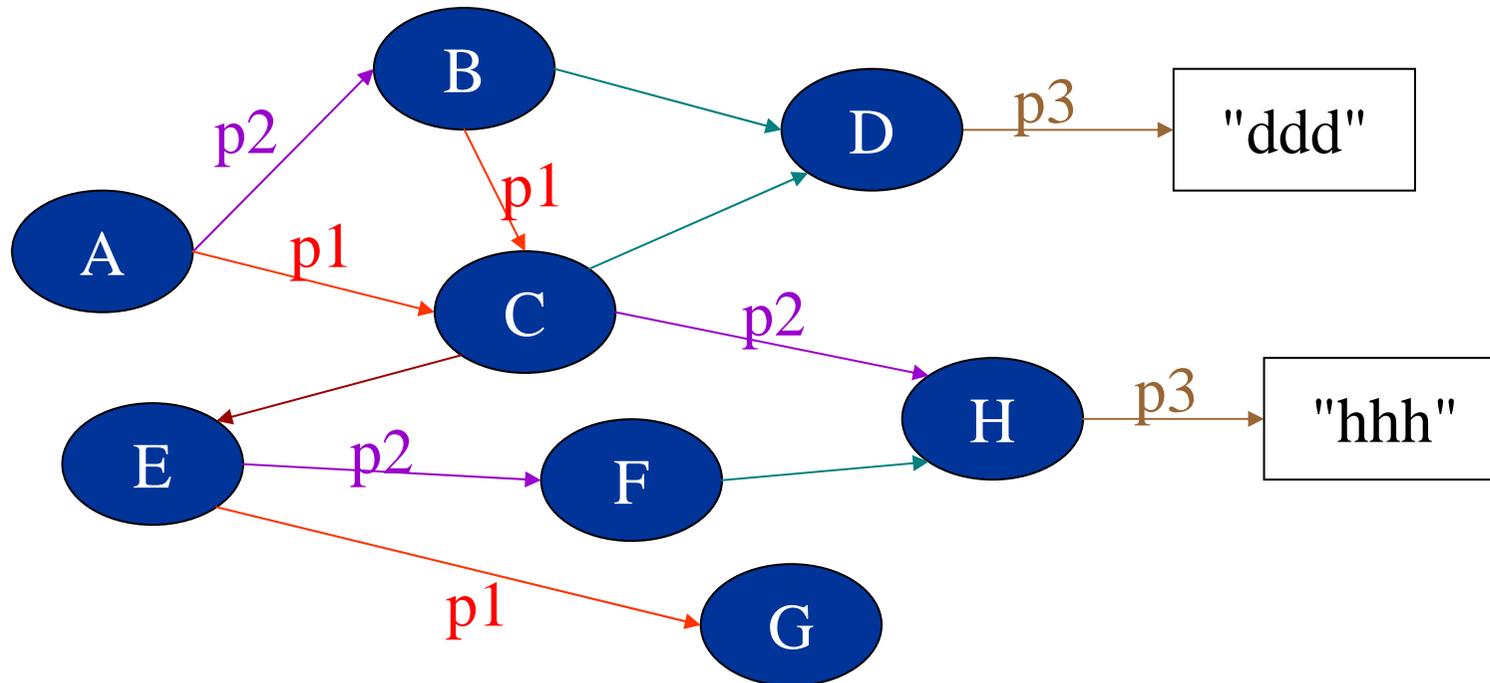
```
<figure itemscope  
        itemtype="型URIxxx" >  
  
    
  
  <figcaption itemprop="タイトル"  
  >私の家</figcaption>  
  
</figure>
```

型: 型URIxxx
作品: "images/house.jpeg"
タイトル: 私の家

RDFグラフ



RDFグラフに対する問合せ

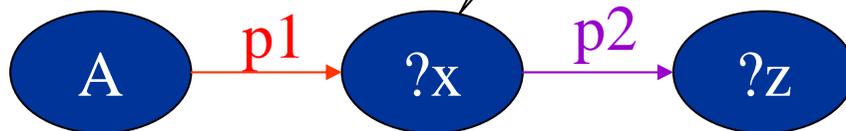


問1 $\xrightarrow{p1}$ $\xrightarrow{p2}$ 結果は

問2 $\xrightarrow{p1}$ $\xrightarrow{?p}$ 結果は

SPARQLでの問合せ

グラフパターン



変数を置く

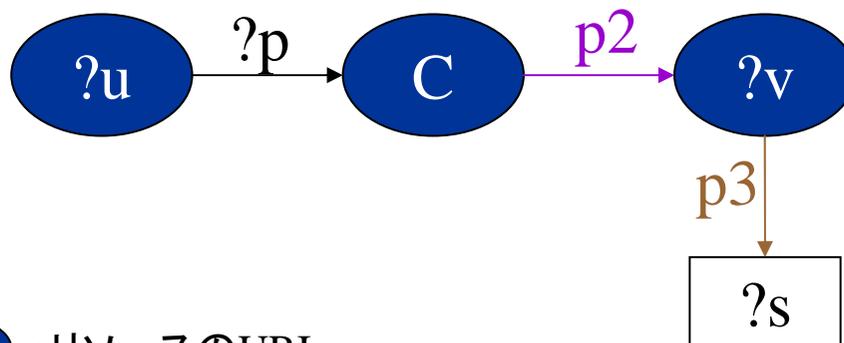
SPARQL問合せ

```
SELECT ?z
WHERE {
  A p1 ?x .
  ?x p2 ?z .
}
```

求める変数

結果

?z → H



```
SELECT ?u
WHERE {
  ?u ?p C .
  C p2 ?v .
  ?v p3 ?s .
}
```

?u → AとB

X : リソースのURI
 pn : プロパティのURI
 ? : 変数を表す