Semantic Web

Languages and Schemata

(1) Metadata and RDF

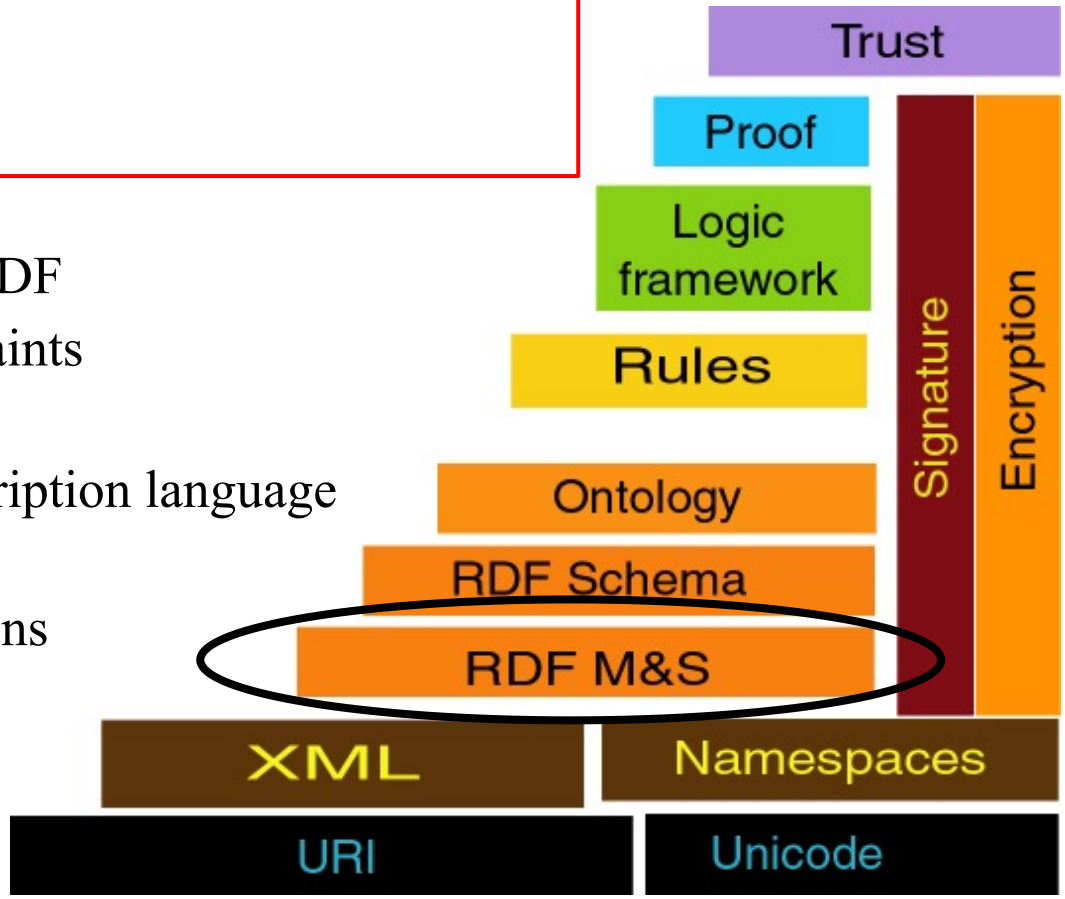
Hideaki Takeda
National Institute of Informatics
takeda@nii.ac.jp

Linked Data

- Languages for Semantic Web
  - Languages for RDF
    - Turtle
    - JSON-LD
    - RDFa
  - Languages for RDF query
    - SPARQL
- Schemata

A Layer model for Semantic Web

- RDF (Resource Description Framework)
  - The most primitive model for metadata description
    - SVO model
    - Entity-Relation Model
    - Semantic net
- RDF Schema
  - Addition of "concept" to RDF
    - class-subclass, constraints
- OWL
  - More general concept description language
    - Logical consistency
    - Various class expressions
    - Various constraints



Tim Berners-Lee http://www.w3.org/2002/Talks/09-lcs-sweb-tbl/

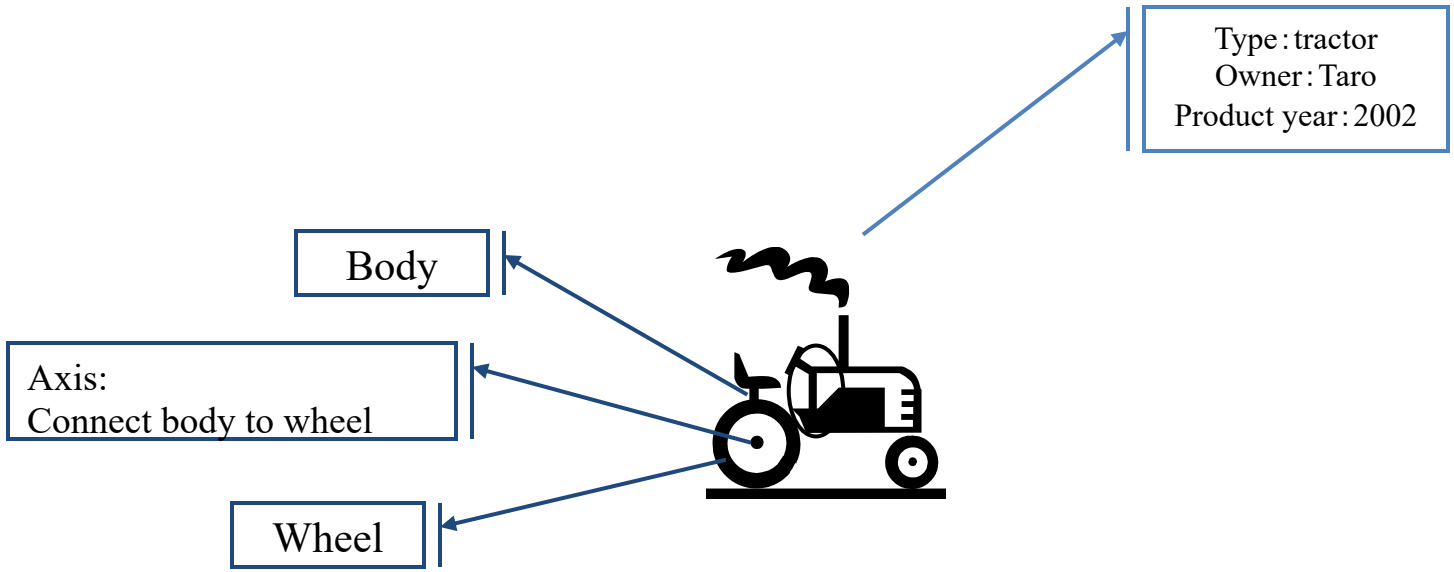
Metadata

- What is metadata?
  - Data about data
  - What one can say about any information object
- What is described as metadata?
  - Content relates to what the object contains or is about, and is *intrinsic* to an information object.
  - Context indicates the who, what, why, where, how aspects associated with the object's creation and is *extrinsic* to an information object.
  - Structure relates to the formal set of associations within or among individual information objects and can be *intrinsic* or *extrinsic*

Setting the State, Anne J.Gilliand-Swetland, Introduction to Metadata – Pathways to Digital Information, Murthsa Baca (ed.), Getty Information Institute.

Metadata

- Metadata to individual information objects
  - Bibliography, Dublin Core
- Metadata to part or structure of information objects
  - Drawings, RDF, RDFS, OWL

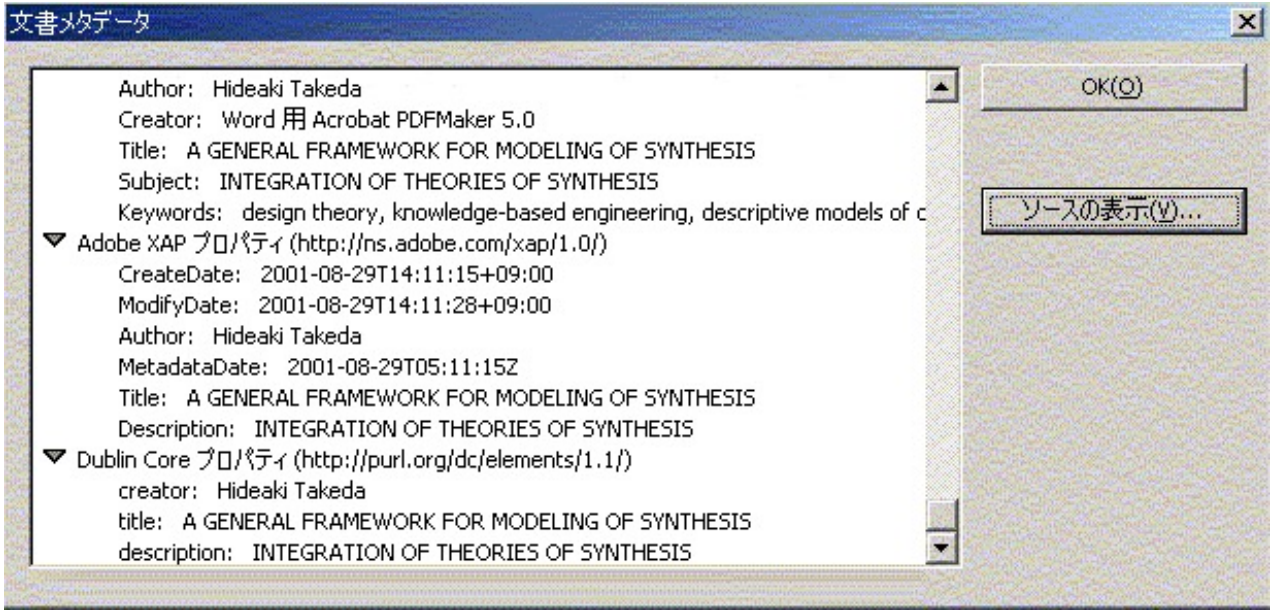


Dublin Core

- Framework for “bibliographic” style metadata
- Simplicity “Pidgins”
  - A few vocabulary
  - A simple structure
- 15 elements: Dublin Core Metadata Element Set
  - Creator, Title, Subject/Keywords, Description, Publisher, Contributor, Date, Resource Type, Format, Resource Identifier, Source, , Language, Relation, Coverage, Rights Management
  - All elements can be omitted and duplicated in any order

Dublin Core

- An example by PDF




```
<rdf:Description about=''
  xmlns='http://purl.org/dc/elements/1.1/'
  xmlns:dc='http://purl.org/dc/elements/1.1/'>
  <dc:creator>Hideaki Takeda</dc:creator>
  <dc:title>A GENERAL FRAMEWORK FOR MODELING
OF SYNTHESIS</dc:title>
  <dc:description>INTEGRATION OF THEORIES OF
SYNTHESIS</dc:description>
 </rdf:Description>
```

RDF (Resource Description Framework)

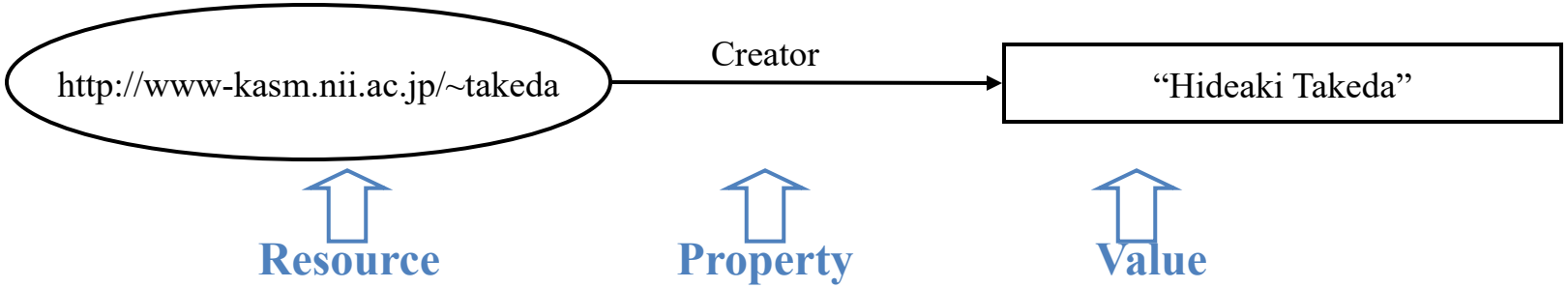
- A framework to describe metadata
- Separation of model and syntax
- W3C Recommendation (2004)

RDF Model

- Element
  - Resource:
    - URI(Universal Resource Identifier)
    - Literal(string)
      - No need to be specified by Web
  - Property:
    - Attribute when describing resources
    - URI or Literal just as Resource
  - Statement: triad of resource, property, and resource

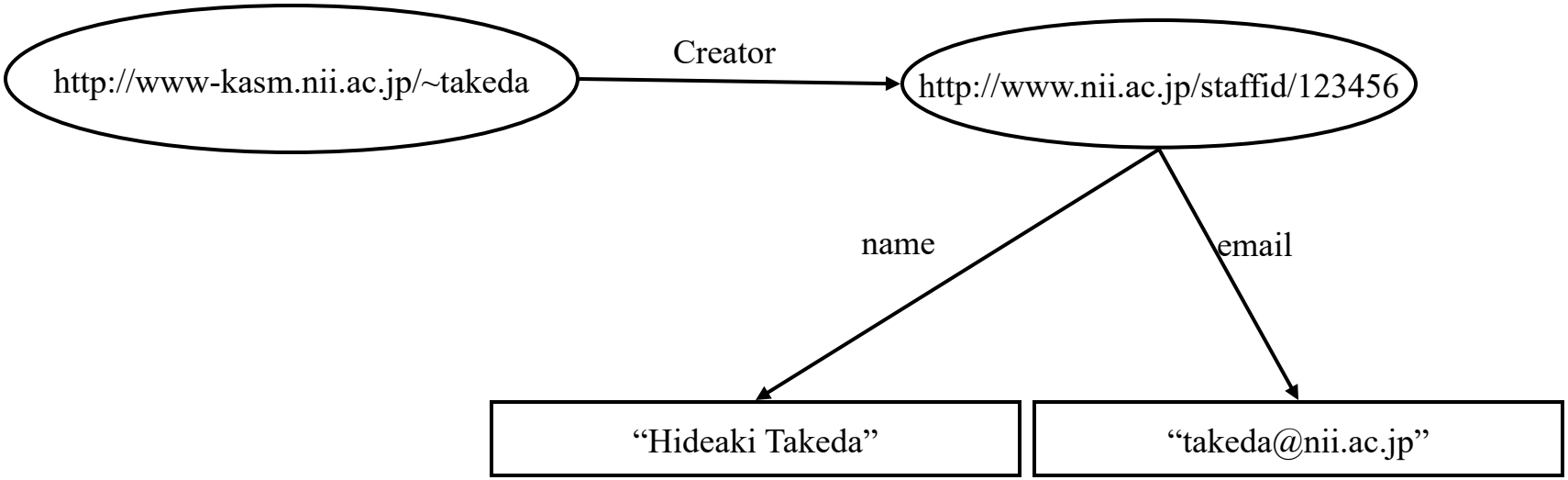
RDF model

- Statement
  - Creator of http://www-kasm.nii.ac.jp/~takeda is "Hideaki Takeda"
- Structure
  - Resource (subject): http://www-kasm.nii.ac.jp/~takeda
  - Property (predicate): Creator
  - Value (object): "Hideaki Takeda"



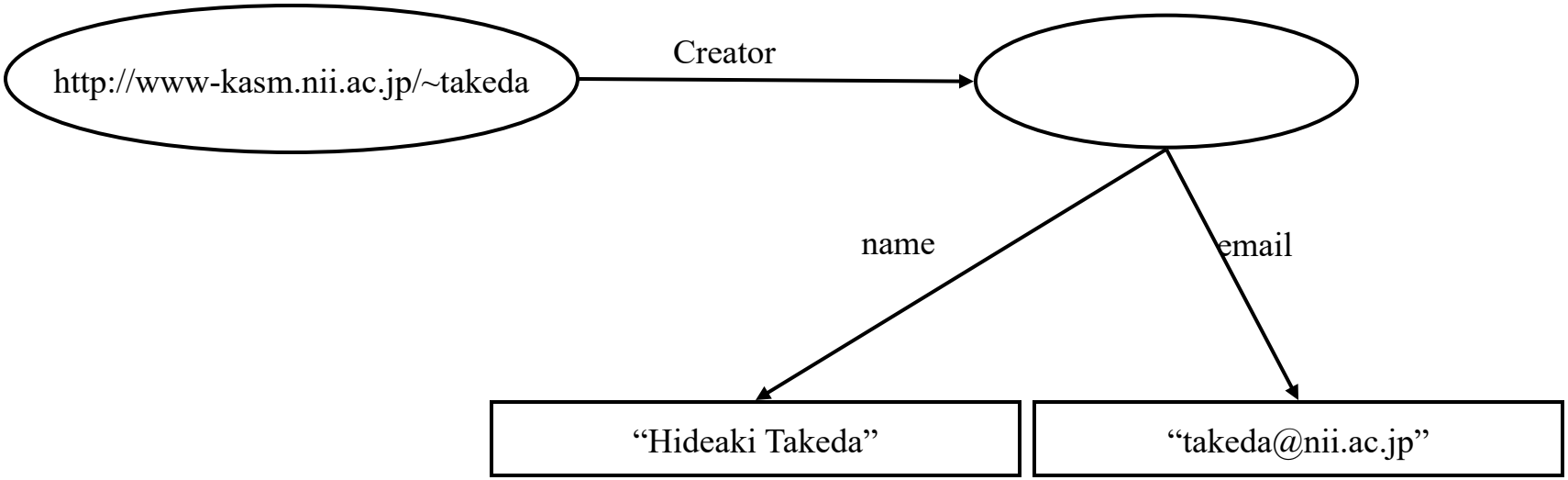
RDF model

- Creator of http://www-kasm.nii.ac.jp/~takeda is http://www.nii.ac.jp/staffid/123456 which has name “Hideaki Takeda” and email “takeda@nii.ac.jp” .



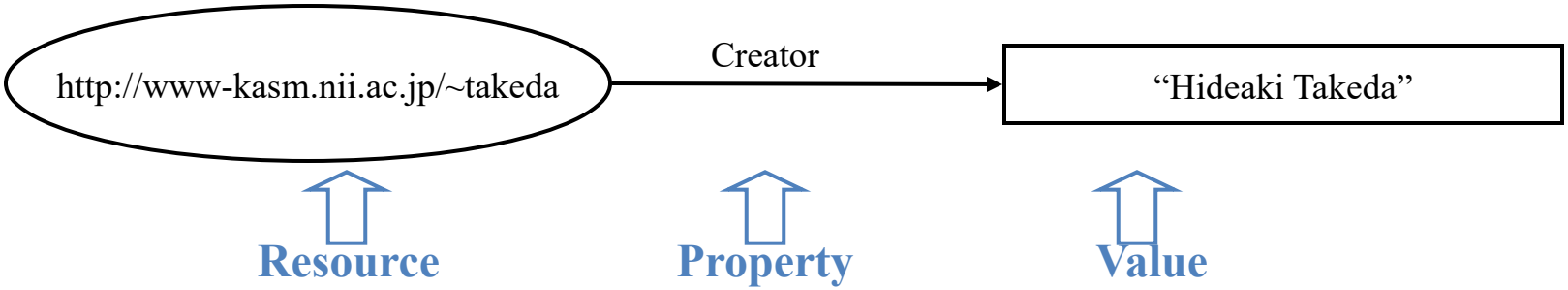
RDF model

- Creator of http://www-kasm.nii.ac.jp/~takeda has name “Hideaki Takeda” email “takeda@nii.ac.jp” .



RDF/Turtle

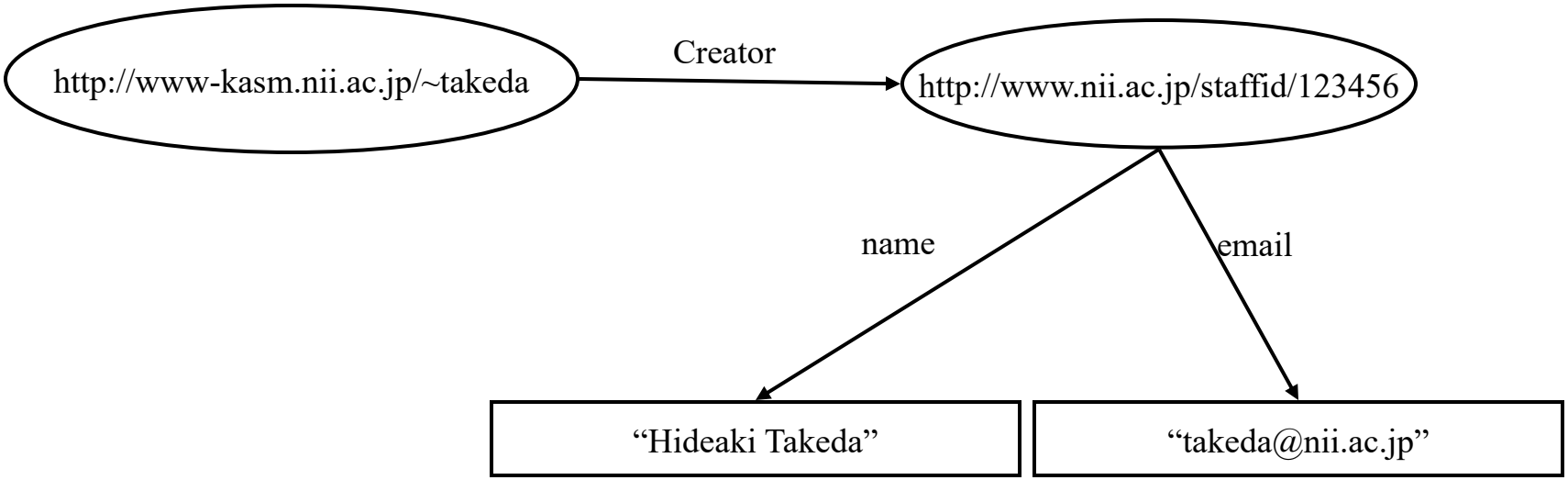
- Creator of http://www-kasm.nii.ac.jp/~takeda is “Hideaki Takeda”




```
@prefix dc: <http://dublincore.org/2001/08/14/dces#> .
<http://www-kasm.nii.ac.jp/~takeda> dc:Creator "Hideaki Takeda" .
```

RDF/Turtle

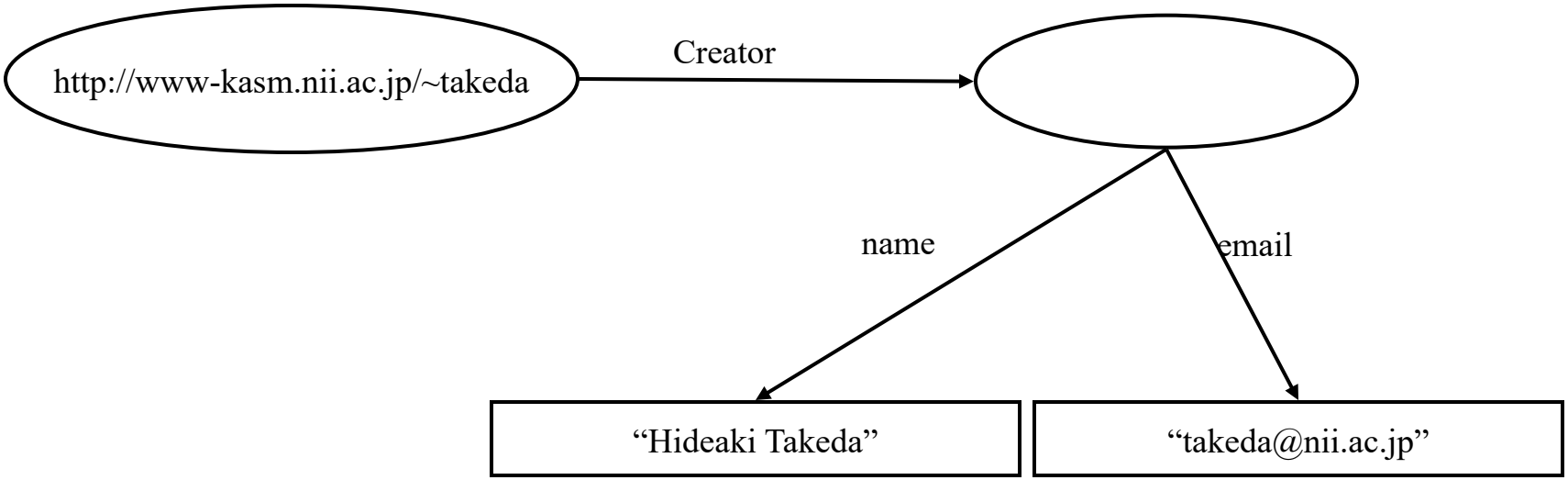
- Creator of http://www-kasm.nii.ac.jp/~takeda is http://www.nii.ac.jp/staffid/123456 which has name “Hideaki Takeda” and email “takeda@nii.ac.jp” .




```
@prefix dc: <http://dublincore.org/2001/08/14/dces#> .
@prefix ex: <http://example.org/schema> .
<http://www-kasm.nii.ac.jp/~takeda> dc:Creator
<http://www.nii.ac.jp/staffid/123456> .
<http://www.nii.ac.jp/staffid/123456> ex:name "Hideaki Takeda" .
<http://www.nii.ac.jp/staffid/123456> ex:email "takeda@nii.ac.jp" .
```

RDF syntax

Creator of http://www-kasm.nii.ac.jp/~takeda has name “Hideaki Takeda” email “takeda@nii.ac.jp” .



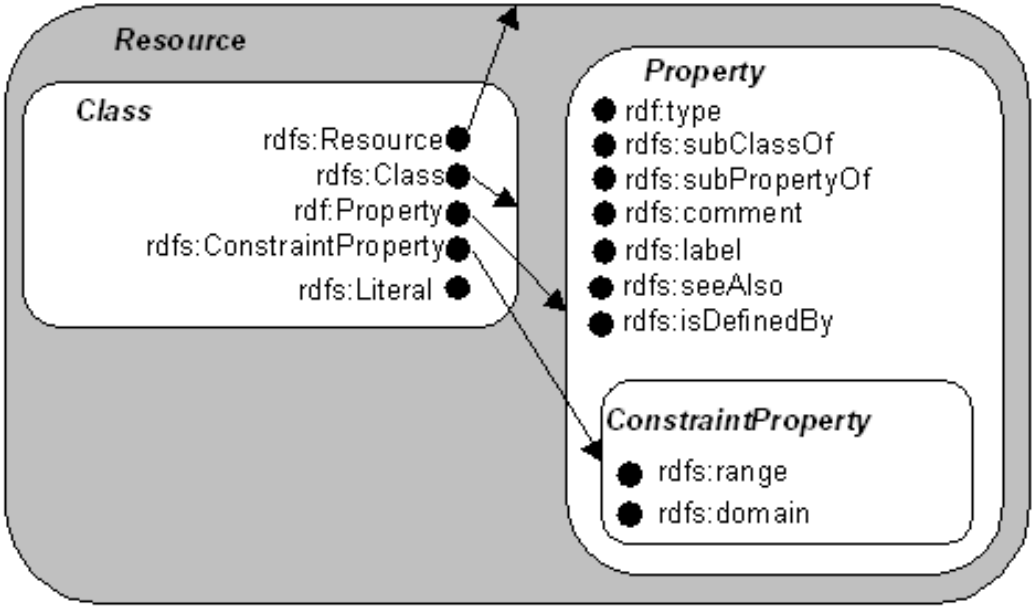

```
@prefix dc: <http://dublincore.org/2001/08/14/dces#> .
@prefix ex: <http://example.org/schema> .
<http://www-kasm.nii.ac.jp/~takeda> dc:Creator [
        ex:name "Hideaki Takeda" ;
        ex:email "takeda@nii.ac.jp"
        ] .
```

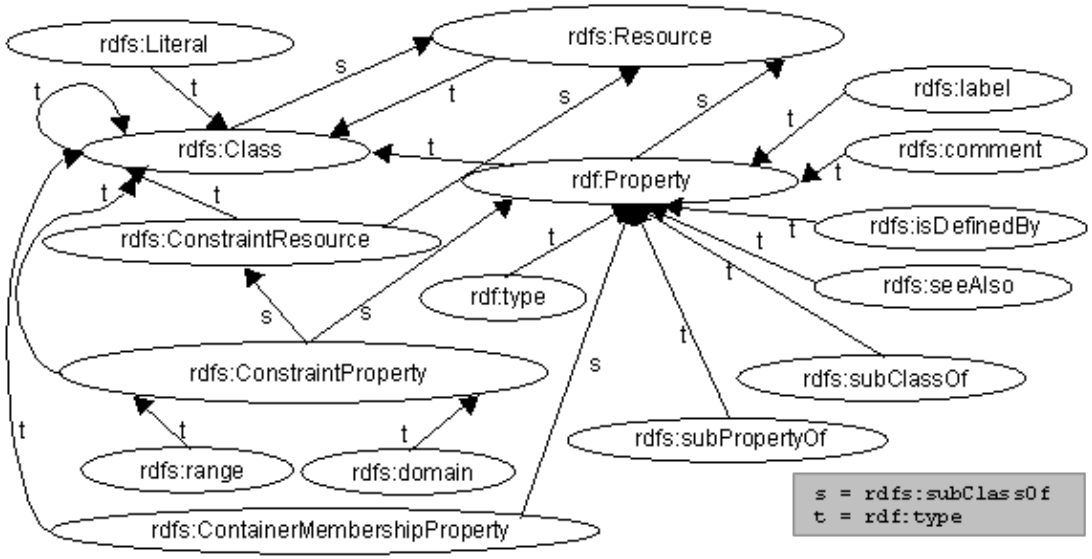
RDFS (RDF Schema)

- Stronger knowledge representation model
  - RDF: ER model, semantic net
  - RDF Schema: Frame model, object-oriented paradigm
    - Minimal definition
    - Property-centered approach
- RDFS is defined as extension of RDF
- RDFS gives definitions of RDF descriptions

RDFS

- Class Definition
  - rdfs:Resource
  - rdfs:Class
  - rdf:Property
  - rdfs:ConstraintProperty
  - rdfs:Literal
- Property Definition
  - rdf:type
  - rdfs:subClassOf
  - rdfs:subPropertyOf
  - rdfs:comment
  - rdfs:label
  - rdfs:seeAlso
  - rdfs:isDefinedBy
- ConstraintProperty Definition
  - rdfs:range
  - rdfs:domain





RDFSのClass階層

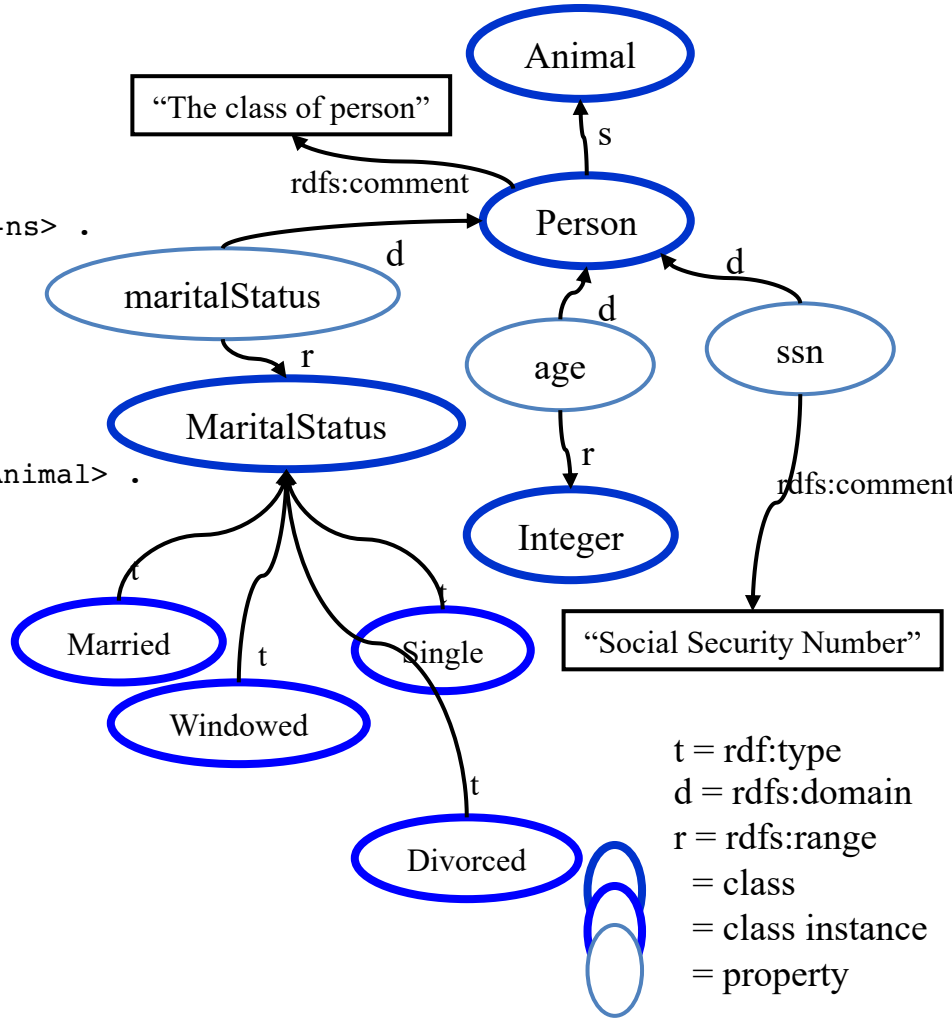
Resource Description Framework(RDF) Schema Specification 1.0
http://www.w3.org/TR/2000/CR-rdf-schema-20000327/

RDF Schema

- rdfs:Class
- rdfs:SubclassOf
  - Detailed class
  - Multiple
  - Transivity
- rdf:type
  - Indicate an instance of a class
- rdf:property
  - Attribute
- rdfs:subPropertyOf
  - Detailed property
  - Transivity
- Range
  - Only one
    - No cardinality
- Domain
  - Multiple (or)

RDF Schema

```
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns> .
@prefix ex: <http://example.org/> .

ex:Person
  a rdfs:Class ;
  rdfs:comment "The class of people." ;
  rdfs:subClassOf
          <http://www.w3.org/2000/03/example/classesAnimal> .

ex:MaritalStatus a rdfs:Class .
ex:Married a ex:MaritalStatus .
ex:Divorced a ex:MaritalStatus .
ex:Single a ex:MaritalStatus .
ex:Widowed a ex:MaritalStatus .

ex:maritalStatus
  a rdf:Property ;
  rdfs:range ex:MaritalStatus ;
  rdfs:domain ex:Person .

ex:ssn
  a rdf:Property ;
  rdfs:comment "Social Security Number" ;
  rdfs:range <http://www.w3.org/2000/03/example/classesInteger> ;
  rdfs:domain ex:Person .

ex:age
  a rdf:Property ;
  rdfs:range <http://www.w3.org/2000/03/example/classesInteger> ;
  rdfs:domain ex:Person .
```



Resource Description Framework(RDF) Schema Specification 1.0
http://www.w3.org/TR/2000/CR-rdf-schema-20000327/

Serialization of RDF

- RDF is an abstract model represented as a graph
- RDF serialization gives the format to describe RDF statements as a textual expression.
- Different serialization formats
  - RDF/XML
  - Turtle
  - N-Triples
  - JSON-LD
  - RDFa

Turtle

- Syntax
  - `[Subject] [Predicate] [Object] .`

```
<http://example.org/#spiderman>
<http://www.perceive.net/schemas/relationship/enemyOf>
<http://example.org/#green-goblin> .
```

  - `[Subject] [Predicate1] [Object1] ;`
    `[Predicate2] [Object2] .`

```
<http://example.org/#spiderman>
<http://www.perceive.net/schemas/relationship/enemyOf>
        <http://example.org/#green-goblin> ;
        <http://xmlns.com/foaf/0.1/name> "Spiderman" .
```

  - `[Subject] [Predicate] [Object1] ,`
    `[Object2] .`

```
<http://example.org/#spiderman> <http://xmlns.com/foaf/0.1/name>
        “Spiderman”@en,
        “スパイダーマン"@ja .
```

Turtle

- IRI
  - Absolute IRI: ex. `<http://example.org/#green-goblin>`
  - Relative IRI: ex.

```
@base <http://example.org/> .
#green-goblin <http://www.perceive.net/schemas/relationship/enemyOf>
#spiderman .

<http://example.org/#green-goblin>
 <http://www.perceive.net/schemas/relationship/enemyOf>
 <http://example.org/#spiderman> .
```

  - Prefixed name: a prefix label + local part

```
@base <http://example.org/> .
@prefix rel: <http://www.perceive.net/schemas/relationship/> .
#green-goblin rel:enemyOf #spiderman .

<http://example.org/#green-goblin>
 <http://www.perceive.net/schemas/relationship/enemyOf>
 <http://example.org/#spiderman> .
```

Turtle

- RDF literals

```
@prefix foaf: <http://xmlns.com/foaf/0.1/> .

<http://example.org/#green-goblin> foaf:name "Green Goblin" .
<http://example.org/#spiderman> foaf:name "Spiderman" .
```

Turtle

- Numbers

Data Type	Abbreviated	Lexical	Description
xsd:integer	-5	"-5"^^xsd:integer	Integer values may be written as an optional sign and a series of digits. Integers match the regular expression "[+-]?[0-9]+".
xsd:decimal	-5.0	"-5.0"^^xsd:decimal	Arbitrary-precision decimals may be written as an optional sign, zero or more digits, a decimal point and one or more digits. Decimals match the regular expression "[+-]?[0-9]*¥.[0-9]+".
xsd:double	4.2E9	"4.2E9"^^xsd:double	Double-precision floating point values may be written as an optionally signed mantissa with an optional decimal point, the letter "e" or "E", and an optionally signed integer exponent. The exponent matches the regular expression "[+-]?[0-9]+" and the mantissa one of these regular expressions: "[+-]?[0-9]+¥.[0-9]+", "[+-]?¥.[0-9]+" or "[+-]?[0-9]".

  - Ex.

```
@prefix : <http://example.org/elements> .
<http://en.wikipedia.org/wiki/Helium>
     :atomicNumber 2 ; # xsd:integer
     :atomicMass 4.002602 ; # xsd:decimal
     :specificGravity 1.663E-4 . # xsd:double
```

- Booleans

```
@prefix : <http://example.org/stats> .
<http://somecountry.example/census2007> :isLandlocked false . # xsd:boolean
```

Turtle

- Blank nodes

```
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
_:alice foaf:knows _:bob .
_:bob foaf:knows _:alice .
```

- Nesting Unlabeled Blank Nodes in Turtle

```
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
    # Someone knows someone else, who has the name "Bob".
    [] foaf:knows [ foaf:name "Bob" ] .

@prefix foaf: <http://xmlns.com/foaf/0.1/> .
[ foaf:name "Alice" ] foaf:knows [
         foaf:name "Bob" ;
         foaf:knows [ foaf:name "Eve" ] ;
         foaf:mbox <bob@example.com> ] .

 _:a foaf:knows "Alice" .
 _:a foaf:knows _:b .
 _:b foaf:name "Bob" .
 _:b foaf:knows _:c .
 _:c foaf:name "Eve" .
 _:b foaf:mbox <bob@example.com> .
```

N-Triples

- Line-based RDF Graph
- Pros and cons in comparison with Turtle
  - Pros
    - Can process line by line
  - Cons
    - Bigger data

N-Triples

- IRI Representations

	Turtle	N-Triples	example
absolute IRI	Yes	yes	<http://a.example/some/path/>
relative IRI	Yes	no	</some/path/>
prefixed name	yes	no	rdfs:label
a for the predicate rdf:type	yes	no	a

- Literal Representations

	Turtle	N-Triples	example
single-quoted single-line lexical representation	yes	no	'some literal'
double-quoted single-line lexical representation	yes	yes	"some literal"
single-quoted multi-line lexical representation	yes	no	'''some literal'''
double-quoted multi-line lexical representation	yes	no	"""some literal"""
abbreviated numeric	yes	no	13
abbreviated boolean	yes	no	true

- Blank Node Representations in N-Triples and Turtle

	Turtle	N-Triples	example
labeled blank node	yes	yes	<http://a.example/who#Alice> <http://xmlns.com/foaf/0.1/knows> _:bob .
anonymous node	yes	no	<http://a.example/who#Alice> foaf:knows [] .
blank node property list	yes	no	<http://a.example/who#Alice> foaf:knows [foaf:name "Bob"] .

JSON-LD

- An extension of JSON for semantic data
  - But still within JSON syntax
- Introduction of reserved names starting with '@'
  - @context
  - @type
- Use for schema.org

JSON-LD

- Example 1

```
{
    "name": "Manu Sporny",
    "homepage": "http://manu.sporny.org/",
    "image": http://manu.sporny.org/images/manu.png
}
```

JSON-LD

- Example 2

```
{
  "http://schema.org/name": "Manu Sporny",
  "http://schema.org/url":
        {
              "@id": http://manu.sporny.org/
        }
   "http://schema.org/image":
        {
              "@id": "http://manu.sporny.org/images/manu.png"
        }
}
```

The '@id' keyword means 'This value is an identifier that is an IRI'

JSON-LD

- Example 3

```
{
 "@context": {
      "name": "http://schema.org/name",
      "image": {
          "@id": "http://schema.org/image",
          "@type": "@id"
      },
      "homepage": {
          "@id": "http://schema.org/url",
          "@type": "@id"
      }
  }
}
```

This means that 'name' is shorthand for 'http://schema.org/name'

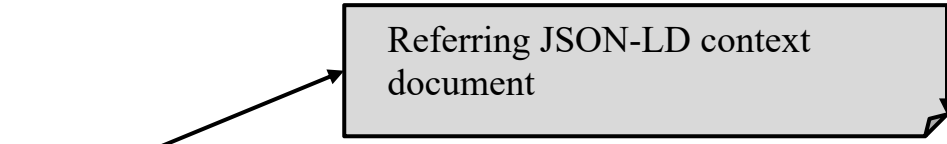
This means that 'image' is shorthand for 'http://schema.org/image

This means that a string value associated with 'image' should be interpreted as an identifier that is an IRI

This means that 'homepage' is shorthand for 'http://schema.org/url"

This means that a string value associated with 'homepage' should be interpreted as an identifier that is an IRI

- Example 4




```
{
  "@context": "https://json-ld.org/contexts/person.jsonld",
  "name": "Manu Sporny",
  "homepage": "http://manu.sporny.org/",
  "image": "http://manu.sporny.org/images/manu.png"
}
```

JSON-LD

- Example 5

```
{
 "@context": {
      "name": "http://schema.org/name",
      "image": {
          "@id": "http://schema.org/image",
          "@type": "@id"
      },
      "homepage": {
          "@id": "http://schema.org/url",
          "@type": "@id"
      }
  }
 "name": "Manu Sporny",
 "homepage": "http://manu.sporny.org/",
 "image": "http://manu.sporny.org/images/manu.png"
}
```

JSON-LD @type

- Example 6

```
{
 "@context": {
   ...
   "givenName": "http://schema.org/givenName",
   "familyName": "http://schema.org/familyName"
 },
 "@id": "http://me.markus-lanthaler.com/",
 "@type": "http://schema.org/Person",
 "givenName": "Markus",
 "familyName": "Lanthaler",
 ...
}
```

- Example 7

```
{
 "@context": {
   ...
   "Person": "http://schema.org/Person"
 },
 "@id": "http://example.org/places#BrewEats",
 "@type": "Person",
```

The value of a @type key may also be a term defined in the active context:

JSON-LD Referring objects

- Example 8

```
{
 "@context": {
   "@vocab": "http://xmlns.com/foaf/0.1/",
   "knows": {"@type": "@id"}
 },
 "@id": "http://manu.sporny.org/about#manu",
 "@type": "Person",
 "name": "Manu Sporny",
 "knows": "https://greggkellogg.net/foaf#me"
}
```

declare a *default* vocabulary from which all terms derive, without having to declare specific mappings for each term (as you normally do in a @context).

JSON-LD embedding

- Example 9

```
{
 "@context": {
  "@vocab": "http://xmlns.com/foaf/0.1/"
 },
 "@id": "http://manu.sporny.org/about#manu",
 "@type": "Person",
 "name": "Manu Sporny",
 "knows": {
  "@id": "https://greggkellogg.net/foaf#me",
  "@type": "Person",
  "name": "Gregg Kellogg"
 }
}
```

Use of JSON-LD with schema.org

```
<script type="application/ld+json">
{
  "@context": "https://schema.org",
  "@type": "Organization",
  "url": "http://www.example.com",
  "name": "Unlimited Ball Bearings Corp.",
  "contactPoint": {
    "@type": "ContactPoint",
    "telephone": "+1-401-555-1212",
    "contactType": "Customer service"
  }
}
</script>
```

```
<script type="application/ld+json">
{
  "@context": "https://schema.org/",
  "@type": "Recipe",
  "name": "Grandma's Holiday Apple Pie",
  "author": "Elaine Smith",
  "image": "http://images.edge-generalmills.com/56459281-6fe6-4d9d-984f-385c9488d824.jpg",
  "description": "A classic apple pie.",
  "aggregateRating": {
    "@type": "AggregateRating",
    "ratingValue": "4",
    "reviewCount": "276",
    "bestRating": "5",
    "worstRating": "1"
  },
 "prepTime": "PT30M",
  "totalTime": "PT1H",
  "recipeYield": "8",
  "nutrition": {
    "@type": "NutritionInformation",
    "servingSize": "1 medium slice",
    "calories": "230 calories",
    "fatContent": "1 g",
    "carbohydrateContent": "43 g",
  },
  "recipeIngredient": [
    "1 box refrigerated pie crusts, softened as directed on box",
    "6 cups thinly sliced, peeled apples (6 medium)",
    "..."
  ],
  "recipeInstructions": [
    "1...",
    "2..."
   ]
}
</script>
```

Google Structured data testing tool

Google 構造化データ テストツール https://search.google.com/structured-data/testing-tool/u/0/

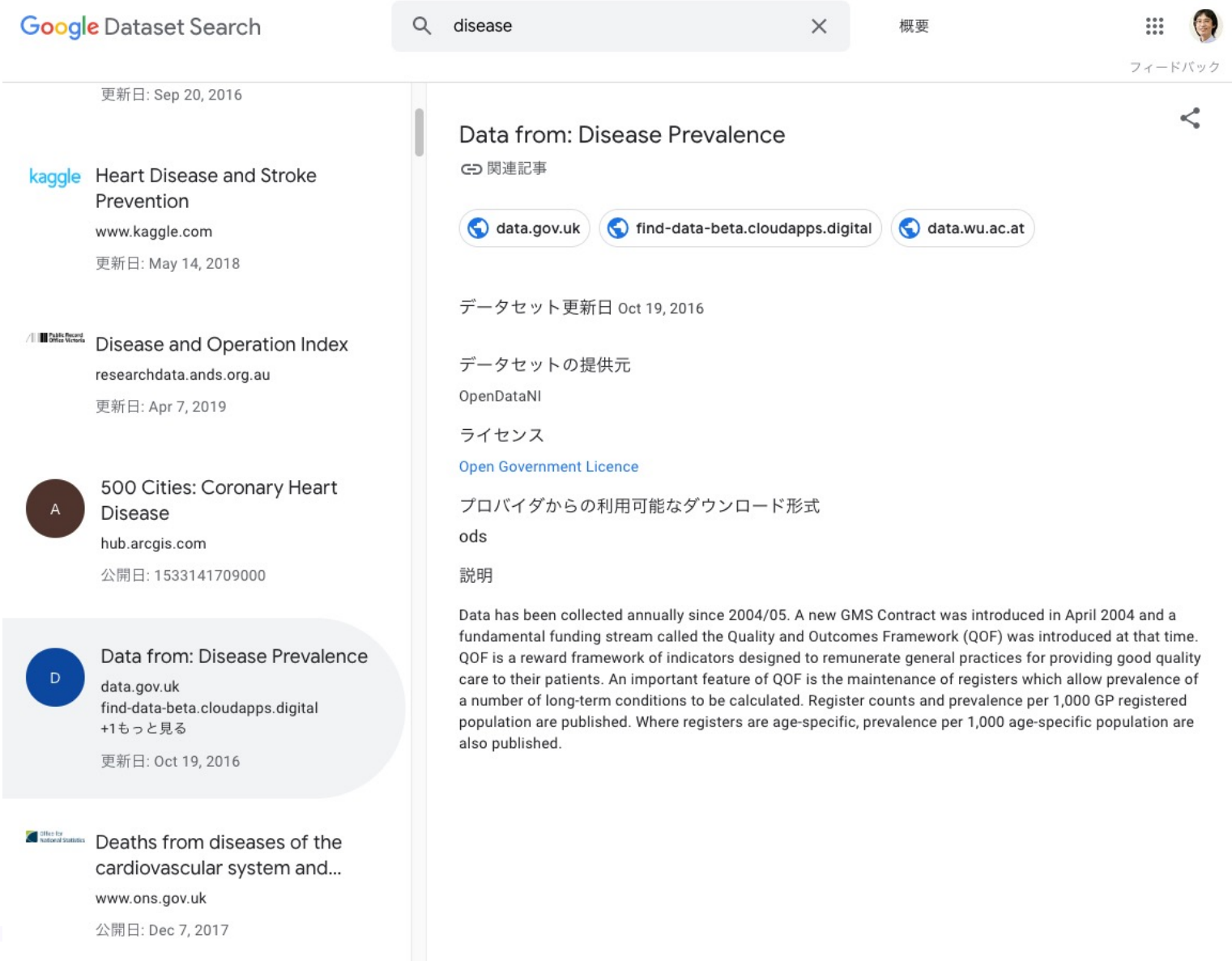
新しいテスト

```
<html>
<body>
<script type="application/ld+json">
{
  "@context": "https://schema.org/",
  "@type": "Recipe",
  "name": "Grandma's Holiday Apple Pie",
  "author": "Elaine Smith",
  "image": "http://images.edge-generalmills.com/56459281-6fe6-4d9d-984f-385c9488d824.jpg",
  "description": "A classic apple pie.",
  "aggregateRating": {
    "@type": "AggregateRating",
    "ratingValue": "4",
    "reviewCount": "276",
    "bestRating": "5",
    "worstRating": "1"
  },
 "prepTime": "PT30M",
  "totalTime": "PT1H",
  "recipeYield": "8",
"nutrition": {
    "@type": "NutritionInformation",
    "servingSize": "1 medium slice",
    "calories": "230 calories",
    "fatContent": "1 g",
    "carbohydrateContent": "43 g"
  },
  "recipeIngredient": [
    "1 box refrigerated pie crusts, softened as directed on box",
    "6 cups thinly sliced, peeled apples (6 medium)",
    "..."
  ],
  "recipeInstructions": [
    "1...",
    "2..."
   ]
}
</script>
</script>
</body>
</html>
```

Recipe プレビュー エラーなし 4 件の警告

@type	Recipe
name	Grandma's Holiday Apple Pie
image	http://images.edge-generalmills.com/56459281-6fe6-4d9d-984f-385c9488d824.jpg
description	A classic apple pie.
prepTime	PT30M
totalTime	PT1H
recipeYield	8
recipeIngredient	1 box refrigerated pie crusts, softened as directed on box
recipeIngredient	6 cups thinly sliced, peeled apples (6 medium)
recipeIngredient	...
recipeInstructions	1...
recipeInstructions	2...
author	
@type	Thing
name	Elaine Smith
aggregateRating	
@type	AggregateRating
ratingValue	4
reviewCount	276
bestRating	5
worstRating	1
nutrition	
@type	NutritionInformation
servingSize	1 medium slice
calories	230 calories
fatContent	1 g
carbohydrateContent	43 g
keywords	「keywords」は推奨フィールドです。

Google Dataset Search



https://data.gov.uk/dataset/baa48aa1-63d1-45d7-a44b-608c71c8f46d/disease-prevalence

data.gov.uk | Find open data

Publish your data Support

BETA This is a new service – your feedback will help us to improve it

Home > OpenDataNI > Disease Prevalence

Disease Prevalence

Published by: OpenDataNI
Last updated: 19 October 2016
Topic: Health
Licence: Open Government Licence

Summary

Data has been collected annually since 2004/05. A new GMS Contract was introduced in April 2004 and a fundamental funding stream called the Quality and Outcomes Framework (QOF) was introduced at that time. QOF is a reward framework of indicators designed to remunerate general practices for providing good quality care to their patients. An important feature of QOF is the maintenance of registers which allow prevalence of a number of

View full summary

More from this publisher
All datasets from OpenDataNI

Related datasets
Quality and Outcomes Framework (QOF): Disease prevalence, achievement and exceptions data

Northern Ireland GMS Quality and Outcomes Framework Prevalence Bulletin

Quality & Outcomes Framework (QOF) of the new GMS Contract

Deaths from Liver Disease

Search

Data links

Link to the data	Format	File added	Data preview
Disease Prevalence Trends 2004-05 to 2015-16	ODS	19 October 2016	Not available

<script type="application/ld+json">
{
"@context":"http://schema.org",
"@type":"Dataset",
"name":"Disease Prevalence",
"url":"https://data.gov.uk/dataset/baa48aa1-63d1-45d7-a44b-608c71c8f46d/disease-prevalence",
"includedInDataCatalog":
{
"@type":"DataCatalog",
"url":https://data.gov.uk
},
"creator":
{
"@type":"Organization",
"name":"OpenDataNI"
},
"description":"Data has been collected annually since 2004/05. A new GMS Contract was introduced in April 2004 and a fundamental funding stream called the Quality and Outcomes Framework (QOF) was introduced at that time. QOF is a reward framework of indicators designed to remunerate general practices for providing good quality care to their patients. An important feature of QOF is the maintenance of registers which allow prevalence of a number of long-term conditions to be calculated. Register counts and prevalence per 1,000 GP registered population are published. Where registers are age-specific, prevalence per 1,000 age-specific population are also published.",
"license":
{
"@type":"CreativeWork",
"name":"Open Government Licence",
"text":null,
"url":http://www.nationalarchives.gov.uk/doc/open-government-licence/version/3/
},
"dateModified":"2016-10-19T10:00:20.345Z",
"keywords":"Health",
"distribution":[
{
"@type":"DataDownload",
"contentUrl":"https://www.opendatani.gov.uk/dataset/baa48aa1-63d1-45d7-a44b-608c71c8f46d/resource/b06c5b0f-fb6e-40aa-83dd-e5e05435bbf3/download/disease-prevalence-trends2004-05-to-2015-16.ods",
"fileFormat":"ODS",
"name":"Disease Prevalence Trends 2004-05 to 2015-16"
}
]
}
</script>

RDFa

- Add extra structured content to the (X)HTML pages
  - adds new (X)HTML/XML attributes
    - "*RDF in attributes*"
  - Programs can extract those and turn into RDF
  - Flexibility for using Literals and URI resources

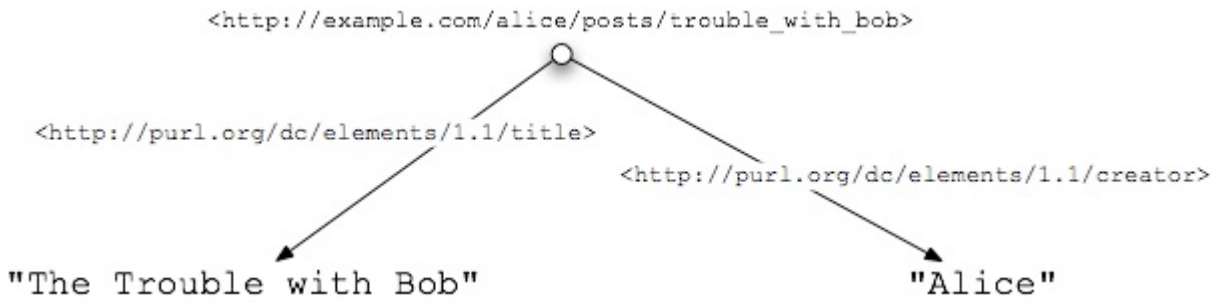
Principles of RDFa

- RDF contents are defined through XML attributes (no elements)
- XML/HTML *tree structure* is used
- Various attributes are *defined* by RDFa
  - Some attributes (@href, @rel) are also reused
- The text content can be also reused

Examples

http://example.com/alice/posts/trouble_with_bob

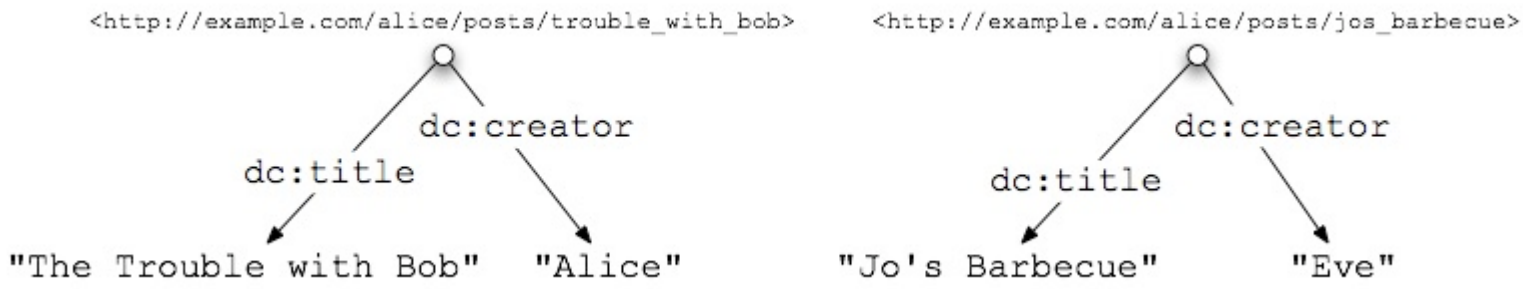
```
<div xmlns:dc="http://purl.org/dc/elements/1.1/">
  <h2 property="dc:title">The trouble with Bob</h2>
  <h3 property="dc:creator">Alice</h3>
   ...
</div>
```



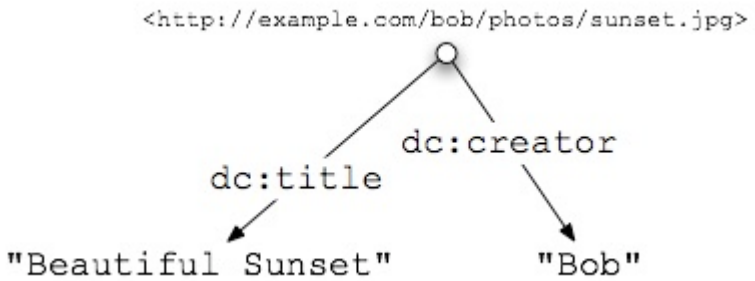
In N3

```
<http://www.example.com/alice/posts/trouble_with_bob>
  <http://purl.org/dc/elements/1.1/title> "The Trouble with Bob";
  <http://purl.org/dc/elements/1.1/creator> "Alice" .
```

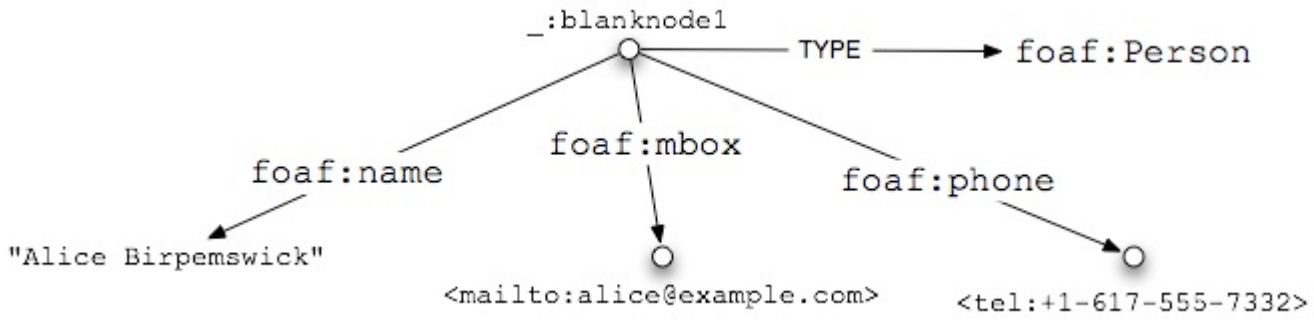
```
<div xmlns:dc="http://purl.org/dc/elements/1.1/">
  <div about="/alice/posts/trouble_with_bob">
    <h2 property="dc:title">The trouble with Bob</h2>
      <h3 property="dc:creator">Alice</h3>
   ...
  </div>
  <div about="/alice/posts/jos_barbecue">
     <h2 property="dc:title">Jo's Barbecue</h2>
       <h3 property="dc:creator">Eve</h3>
     ...
  </div>
   ...
</div>
```



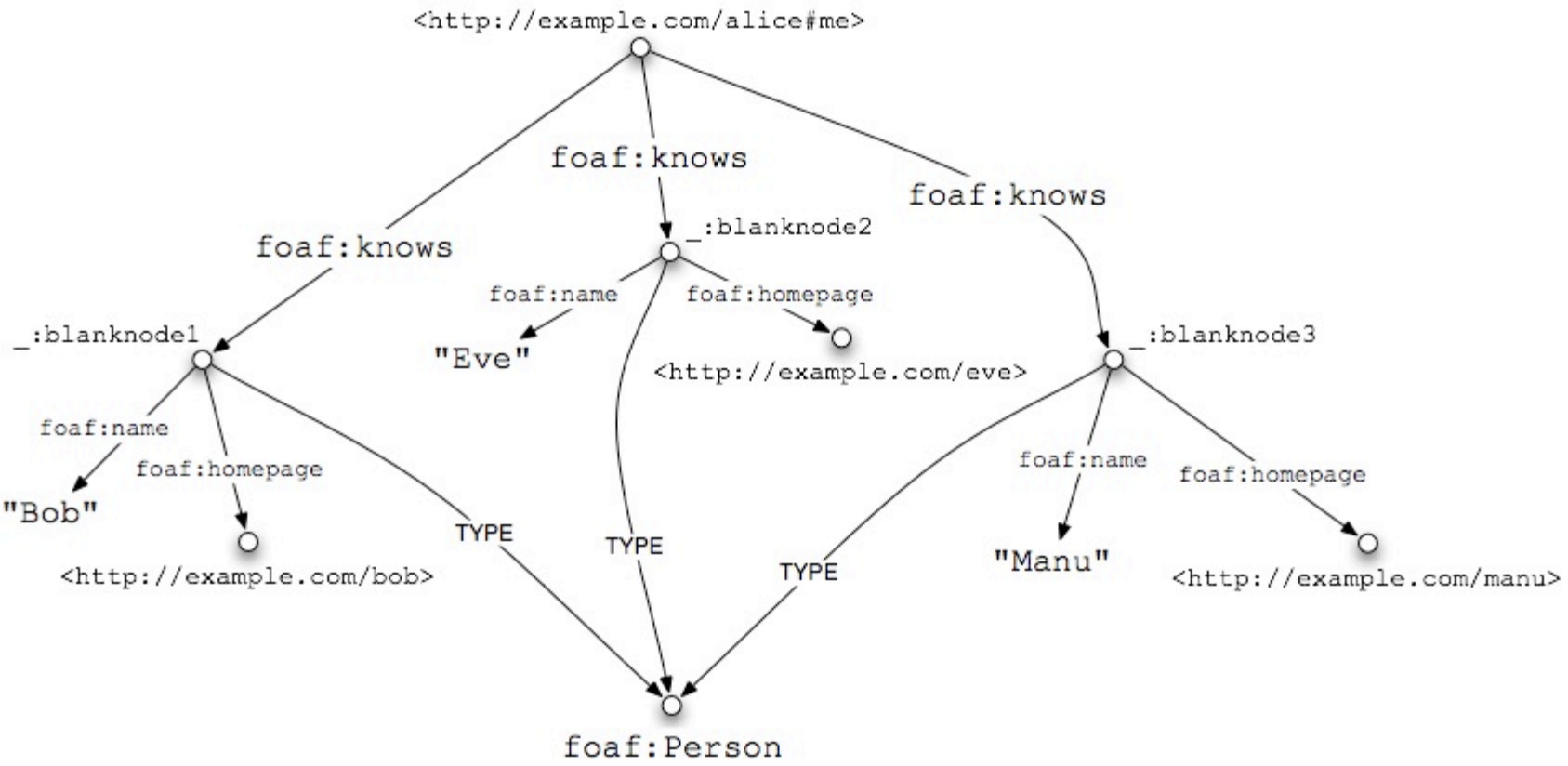
```
<div about="/alice/posts/trouble_with_bob">
  <h2 property="dc:title">The trouble with Bob</h2>
    The trouble with Bob is that he takes much better photos than I do:
  <div about="http://example.com/bob/photos/sunset.jpg">
    <img src="http://example.com/bob/photos/sunset.jpg" />
    <span property="dc:title">Beautiful Sunset</span>
     by
     <span property="dc:creator">Bob</span>.
   </div>
</div>
```



```
<div typeof="foaf:Person" xmlns:foaf="http://xmlns.com/foaf/0.1/">
  <p property="foaf:name"> Alice Birpemswick </p>
  <p> Email: <a rel="foaf:mbox" href="mailto:alice@example.com">alice@example.com</a></p>
  <p> Phone: <a rel="foaf:phone" href="tel:+1-617-555-7332">+1 617.555.7332</a> </p>
</div>
```



```
<div xmlns:foaf="http://xmlns.com/foaf/0.1/" about="#me" rel="foaf:knows">
  <ul>
    <li typeof="foaf:Person">
      <a property="foaf:name" rel="foaf:homepage" href="http://example.com/bob">Bob</a>
    </li>
    <li typeof="foaf:Person">
      <a property="foaf:name" rel="foaf:homepage" href="http://example.com/eve">Eve</a>
    </li>
    <li typeof="foaf:Person">
      <a property="foaf:name" rel="foaf:homepage" href="http://example.com/manu">Manu</a>
    </li>
  </ul>
</div>
```



Using RDFa

- RDF Validator
  - http://validator.w3.org/
    - Also graph visualization for small dataset
    - Only RDF/XML
- RDF Distiller
  - http://www.w3.org/2007/08/pyRdfa/

- RDF Validator & Converter
  - http://rdfvalidator.mybluemix.net/

- RDF Converter & Visualizer
  - http://www.easyrdf.org/converter

Which is the best serialization?

- RDF/XML
  - Pros: Compatible with existing XML-processing systems
  - Cons: Complicated and arbitrary, Not good for human reading
  - Recommended: Adding RDF data to the existing web services
- Turtle
  - Pros
    - Natural expression for RDF, text-based
    - Good for human reading
    - Relatively good for machine processing
  - Cons: Require specific processing
  - Recommended: Dumped files
- N-Triples
  - Pros: Line-based
  - Cons: Bigger data
  - Recommended: internal data
- JSON-LD
  - Pros:
    - well-used language (JSON)
    - Simple syntax
  - Cons:
    - Not suitable for complex structure
- RDFa
  - Pros: mixed with html
  - Cons: Difficult to edit, read, process
  - Recommend
    - Providing new web services with RDF

RDF validator

- Checking RDF statements whether they are valid or not
  - http://www.w3.org/RDF/Validator/
    - RDF/XML
  - http://www.rdfabout.com/demo/validator/
    - RDF/XML
    - Notation 3 (N-Triples, Turtle)