

CSE 6708 (Semantic Web)

RDFS and Ontology

Dr. Muhammad Masroor Ali

Professor

Department of Computer Science and Engineering
Bangladesh University of Engineering and Technology
Dhaka-1205, Bangladesh

October 2015

Version: 1.1, Last modified: December 18, 2015



Liyang Yu. A Developer's Guide to the Semantic Web.
2nd. Springer, 2014.



RDFS Overview

RDFS in Plain English: *A Developer's Guide to the Semantic Web*, p 121

- Why do we need RDFS, and
- What is RDFS?



RDFS Overview

RDFS in Plain English — *continued: A Developer's Guide to the Semantic Web, p 121*

```
1 <?xml version="1.0"?>
2 <!DOCTYPE rdf:RDF [<!ENTITY xsd "http://www.w3.org/2001/XMLSchema#">]>
3
4 <rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
5     xmlns:uom="http://www.example.org/units#"
6     xmlns:myCamera="http://www.liyangyu.com/camera#">
7
8     <rdf:Description rdf:about="http://www.liyangyu.com/camera#Nikon_D300">
9         <rdf:type rdf:resource="http://www.liyangyu.com/camera#DSLR"/>
10        <myCamera:manufactured_by
11            rdf:resource="http://www.dbpedia.org/resource/Nikon"/>
12        <myCamera:performance>
13            <rdf:Description
14                rdf:about="http://www.liyangyu.com/camera#PictureQuality">
15                <myCamera:evaluate>5 stars</myCamera:evaluate>
16            </rdf:Description>
17        </myCamera:performance>
18        <myCamera:model rdf:datatype="&xsd:string">D300</myCamera:model>
19        <myCamera:weight rdf:parseType="Resource">
20            <rdf:value rdf:datatype="&xsd:decimal">0.6</rdf:value>
21            <uom:units rdf:resource="http://www.example.org/units#kg"/>
22        </myCamera:weight>
23    </rdf:Description>
24
25 </rdf:RDF>
```

RDFS Overview

RDFS in Plain English — *continued: A Developer's Guide to the Semantic Web, p 121*

```
8 <rdf:Description rdf:about="http://www.liyangyu.com/camera#Nikon_D300">
9   <rdf:type rdf:resource="http://www.liyangyu.com/camera#DSLR"/>
10  <myCamera:manufactured_by
11    rdf:resource="http://www.dbpedia.org/resource/Nikon"/>
12  <myCamera:performance>
13    <rdf:Description
14      rdf:about="http://www.liyangyu.com/camera#PictureQuality">
15      <myCamera:evaluate>5 stars</myCamera:evaluate>
16    </rdf:Description>
17  </myCamera:performance>
18  <myCamera:model rdf:datatype="xsd:string">D300</myCamera:model>
19  <myCamera:weight rdf:parseType="Resource">
20    <rdf:value rdf:datatype="xsd:decimal">0.6</rdf:value>
21    <uom:units rdf:resource="http://www.example.org/units#kg"/>
22  </myCamera:weight>
23 </rdf:Description>
```

- Line 9 of List 2.25 says `myCamera:Nikon_D300` is an instance (by using predicate `rdf:type`) of the resource identified by URI `myCamera:DSLR`.



RDFS Overview

RDFS in Plain English — *continued: A Developer's Guide to the Semantic Web, p 121*

```
8 <rdf:Description rdf:about="http://www.liyangyu.com/camera#Nikon_D300">
9   <rdf:type rdf:resource="http://www.liyangyu.com/camera#DSLR"/>
10  <myCamera:manufactured_by
11    rdf:resource="http://www.dbpedia.org/resource/Nikon"/>
12  <myCamera:performance>
13    <rdf:Description
14      rdf:about="http://www.liyangyu.com/camera#PictureQuality">
15        <myCamera:evaluate>5 stars</myCamera:evaluate>
16      </rdf:Description>
17    </myCamera:performance>
18    <myCamera:model rdf:datatype="xsd:string">D300</myCamera:model>
19    <myCamera:weight rdf:parseType="Resource">
20      <rdf:value rdf:datatype="xsd:decimal">0.6</rdf:value>
21      <uom:units rdf:resource="http://www.example.org/units#kg"/>
22    </myCamera:weight>
23  </rdf:Description>
```

- But where is this `myCamera:DSLR` resource defined?
- What does it look like?



RDFS Overview

RDFS in Plain English — *continued: A Developer's Guide to the Semantic Web, p 121*

```
8 <rdf:Description rdf:about="http://www.liyangyu.com/camera#Nikon_D300">
9   <rdf:type rdf:resource="http://www.liyangyu.com/camera#DSLR"/>
10  <myCamera:manufactured_by
11    rdf:resource="http://www.dbpedia.org/resource/Nikon"/>
12  <myCamera:performance>
13    <rdf:Description
14      rdf:about="http://www.liyangyu.com/camera#PictureQuality">
15        <myCamera:evaluate>5 stars</myCamera:evaluate>
16      </rdf:Description>
17    </myCamera:performance>
18  <myCamera:model rdf:datatype="xsd:string">D300</myCamera:model>
19  <myCamera:weight rdf:parseType="Resource">
20    <rdf:value rdf:datatype="xsd:decimal">0.6</rdf:value>
21    <uom:units rdf:resource="http://www.example.org/units#kg"/>
22  </myCamera:weight>
23 </rdf:Description>
```

- If we use object-oriented concepts, `myCamera:DSLR` can be understood as a class.



RDFS Overview

RDFS in Plain English — *continued: A Developer's Guide to the Semantic Web, p 121*

```
8 <rdf:Description rdf:about="http://www.liyangyu.com/camera#Nikon_D300">
9   <rdf:type rdf:resource="http://www.liyangyu.com/camera#DSLR"/>
10  <myCamera:manufactured_by
11    rdf:resource="http://www.dbpedia.org/resource/Nikon"/>
12  <myCamera:performance>
13    <rdf:Description
14      rdf:about="http://www.liyangyu.com/camera#PictureQuality">
15      <myCamera:evaluate>5 stars</myCamera:evaluate>
16    </rdf:Description>
17  </myCamera:performance>
18  <myCamera:model rdf:datatype="xsd:string">D300</myCamera:model>
19  <myCamera:weight rdf:parseType="Resource">
20    <rdf:value rdf:datatype="xsd:decimal">0.6</rdf:value>
21    <uom:units rdf:resource="http://www.example.org/units#kg"/>
22  </myCamera:weight>
23 </rdf:Description>
```

-
- Now, if `myCamera:DSLR` represents a class, are there any other classes that are defined as its super classes or subclasses?



RDFS Overview

RDFS in Plain English — *continued: A Developer's Guide to the Semantic Web, p 121*

```
8 <rdf:Description rdf:about="http://www.liyangyu.com/camera#Nikon_D300">
9   <rdf:type rdf:resource="http://www.liyangyu.com/camera#DSLR"/>
10  <myCamera:manufactured_by
11    rdf:resource="http://www.dbpedia.org/resource/Nikon"/>
12  <myCamera:performance>
13    <rdf:Description
14      rdf:about="http://www.liyangyu.com/camera#PictureQuality">
15      <myCamera:evaluate>5 stars</myCamera:evaluate>
16    </rdf:Description>
17  </myCamera:performance>
18  <myCamera:model rdf:datatype="xsd:string">D300</myCamera:model>
19  <myCamera:weight rdf:parseType="Resource">
20    <rdf:value rdf:datatype="xsd:decimal">0.6</rdf:value>
21    <uom:units rdf:resource="http://www.example.org/units#kg"/>
22  </myCamera:weight>
23 </rdf:Description>
```

- The rest of List 2.25 uses several properties (such as `myCamera:model` and `myCamera:weight`) to describe `myCamera:Nikon_D300`.



RDFS Overview

RDFS in Plain English — *continued: A Developer's Guide to the Semantic Web, p 121*

```
8 <rdf:Description rdf:about="http://www.liyangyu.com/camera#Nikon_D300">
9   <rdf:type rdf:resource="http://www.liyangyu.com/camera#DSLR"/>
10  <myCamera:manufactured_by
11    rdf:resource="http://www.dbpedia.org/resource/Nikon"/>
12  <myCamera:performance>
13    <rdf:Description
14      rdf:about="http://www.liyangyu.com/camera#PictureQuality">
15      <myCamera:evaluate>5 stars</myCamera:evaluate>
16    </rdf:Description>
17  </myCamera:performance>
18  <myCamera:model rdf:datatype="xsd:string">D300</myCamera:model>
19  <myCamera:weight rdf:parseType="Resource">
20    <rdf:value rdf:datatype="xsd:decimal">0.6</rdf:value>
21    <uom:units rdf:resource="http://www.example.org/units#kg"/>
22  </myCamera:weight>
23 </rdf:Description>
```

-
- Are there any other properties that we can use to describe myCamera:Nikon_D300?



RDFS Overview

RDFS in Plain English — *continued: A Developer's Guide to the Semantic Web, p 121*

```
8 <rdf:Description rdf:about="http://www.liyangyu.com/camera#Nikon_D300">
9   <rdf:type rdf:resource="http://www.liyangyu.com/camera#DSLR"/>
10  <myCamera:manufactured_by
11    rdf:resource="http://www.dbpedia.org/resource/Nikon"/>
12  <myCamera:performance>
13    <rdf:Description
14      rdf:about="http://www.liyangyu.com/camera#PictureQuality">
15        <myCamera:evaluate>5 stars</myCamera:evaluate>
16      </rdf:Description>
17    </myCamera:performance>
18    <myCamera:model rdf:datatype="xsd:string">D300</myCamera:model>
19    <myCamera:weight rdf:parseType="Resource">
20      <rdf:value rdf:datatype="xsd:decimal">0.6</rdf:value>
21      <uom:units rdf:resource="http://www.example.org/units#kg"/>
22    </myCamera:weight>
23  </rdf:Description>
```

- How do we know these properties exist for us to use in the first place?



RDFS Overview

RDFS in Plain English — *continued: A Developer's Guide to the Semantic Web, p 121*

```
8 <rdf:Description rdf:about="http://www.liyangyu.com/camera#Nikon_D300">
9   <rdf:type rdf:resource="http://www.liyangyu.com/camera#DSLR"/>
10  <myCamera:manufactured_by
11    rdf:resource="http://www.dbpedia.org/resource/Nikon"/>
12  <myCamera:performance>
13    <rdf:Description
14      rdf:about="http://www.liyangyu.com/camera#PictureQuality">
15        <myCamera:evaluate>5 stars</myCamera:evaluate>
16      </rdf:Description>
17    </myCamera:performance>
18  <myCamera:model rdf:datatype="xsd:string">D300</myCamera:model>
19  <myCamera:weight rdf:parseType="Resource">
20    <rdf:value rdf:datatype="xsd:decimal">0.6</rdf:value>
21    <uom:units rdf:resource="http://www.example.org/units#kg"/>
22  </myCamera:weight>
23 </rdf:Description>
```

■ You can ask more questions like these.



RDFS Overview

RDFS in Plain English — *continued: A Developer's Guide to the Semantic Web, p 121*

```
8 <rdf:Description rdf:about="http://www.liyangyu.com/camera#Nikon_D300">
9   <rdf:type rdf:resource="http://www.liyangyu.com/camera#DSLR"/>
10  <myCamera:manufactured_by
11    rdf:resource="http://www.dbpedia.org/resource/Nikon"/>
12  <myCamera:performance>
13    <rdf:Description
14      rdf:about="http://www.liyangyu.com/camera#PictureQuality">
15      <myCamera:evaluate>5 stars</myCamera:evaluate>
16    </rdf:Description>
17  </myCamera:performance>
18  <myCamera:model rdf:datatype="xsd:string">D300</myCamera:model>
19  <myCamera:weight rdf:parseType="Resource">
20    <rdf:value rdf:datatype="xsd:decimal">0.6</rdf:value>
21    <uom:units rdf:resource="http://www.example.org/units#kg"/>
22  </myCamera:weight>
23 </rdf:Description>
```

-
- The last question, in particular, raises an important issue.



RDFS Overview

RDFS in Plain English — *continued: A Developer's Guide to the Semantic Web, p 121*

```
8 <rdf:Description rdf:about="http://www.liyangyu.com/camera#Nikon_D300">
9   <rdf:type rdf:resource="http://www.liyangyu.com/camera#DSLR"/>
10  <myCamera:manufactured_by
11    rdf:resource="http://www.dbpedia.org/resource/Nikon"/>
12  <myCamera:performance>
13    <rdf:Description
14      rdf:about="http://www.liyangyu.com/camera#PictureQuality">
15        <myCamera:evaluate>5 stars</myCamera:evaluate>
16      </rdf:Description>
17    </myCamera:performance>
18    <myCamera:model rdf:datatype="xsd:string">D300</myCamera:model>
19    <myCamera:weight rdf:parseType="Resource">
20      <rdf:value rdf:datatype="xsd:decimal">0.6</rdf:value>
21      <uom:units rdf:resource="http://www.example.org/units#kg"/>
22    </myCamera:weight>
23  </rdf:Description>
```

-
- When we describe a real-world resource such as `myCamera:Nikon_D300`, what are the things (predicates) we can use to describe it?



RDFS Overview

RDFS in Plain English — *continued: A Developer's Guide to the Semantic Web, p 121*

```
8 <rdf:Description rdf:about="http://www.liyangyu.com/camera#Nikon_D300">
9   <rdf:type rdf:resource="http://www.liyangyu.com/camera#DSLR"/>
10  <myCamera:manufactured_by
11    rdf:resource="http://www.dbpedia.org/resource/Nikon"/>
12  <myCamera:performance>
13    <rdf:Description
14      rdf:about="http://www.liyangyu.com/camera#PictureQuality">
15      <myCamera:evaluate>5 stars</myCamera:evaluate>
16    </rdf:Description>
17  </myCamera:performance>
18  <myCamera:model rdf:datatype="xsd:string">D300</myCamera:model>
19  <myCamera:weight rdf:parseType="Resource">
20    <rdf:value rdf:datatype="xsd:decimal">0.6</rdf:value>
21    <uom:units rdf:resource="http://www.example.org/units#kg"/>
22  </myCamera:weight>
23 </rdf:Description>
```

■ Let us say that we all say something about it.



RDFS Overview

RDFS in Plain English — *continued: A Developer's Guide to the Semantic Web, p 121*

```
8 <rdf:Description rdf:about="http://www.liyangyu.com/camera#Nikon_D300">
9   <rdf:type rdf:resource="http://www.liyangyu.com/camera#DSLR"/>
10  <myCamera:manufactured_by
11    rdf:resource="http://www.dbpedia.org/resource/Nikon"/>
12  <myCamera:performance>
13    <rdf:Description
14      rdf:about="http://www.liyangyu.com/camera#PictureQuality">
15      <myCamera:evaluate>5 stars</myCamera:evaluate>
16    </rdf:Description>
17  </myCamera:performance>
18  <myCamera:model rdf:datatype="xsd:string">D300</myCamera:model>
19  <myCamera:weight rdf:parseType="Resource">
20    <rdf:value rdf:datatype="xsd:decimal">0.6</rdf:value>
21    <uom:units rdf:resource="http://www.example.org/units#kg"/>
22  </myCamera:weight>
23 </rdf:Description>
```

- Furthermore, let us say that, we all go on to invent our own things to say about it.



RDFS Overview

RDFS in Plain English — *continued: A Developer's Guide to the Semantic Web, p 121*

```
8 <rdf:Description rdf:about="http://www.liyangyu.com/camera#Nikon_D300">
9   <rdf:type rdf:resource="http://www.liyangyu.com/camera#DSLR"/>
10  <myCamera:manufactured_by
11    rdf:resource="http://www.dbpedia.org/resource/Nikon"/>
12  <myCamera:performance>
13    <rdf:Description
14      rdf:about="http://www.liyangyu.com/camera#PictureQuality">
15        <myCamera:evaluate>5 stars</myCamera:evaluate>
16      </rdf:Description>
17    </myCamera:performance>
18  <myCamera:model rdf:datatype="xsd:string">D300</myCamera:model>
19  <myCamera:weight rdf:parseType="Resource">
20    <rdf:value rdf:datatype="xsd:decimal">0.6</rdf:value>
21    <uom:units rdf:resource="http://www.example.org/units#kg"/>
22  </myCamera:weight>
23 </rdf:Description>
```

-
- There will be no common language shared among us.



RDFS Overview

RDFS in Plain English — *continued: A Developer's Guide to the Semantic Web, p 121*

```
8 <rdf:Description rdf:about="http://www.liyangyu.com/camera#Nikon_D300">
9   <rdf:type rdf:resource="http://www.liyangyu.com/camera#DSLR"/>
10  <myCamera:manufactured_by
11    rdf:resource="http://www.dbpedia.org/resource/Nikon"/>
12  <myCamera:performance>
13    <rdf:Description
14      rdf:about="http://www.liyangyu.com/camera#PictureQuality">
15      <myCamera:evaluate>5 stars</myCamera:evaluate>
16    </rdf:Description>
17  </myCamera:performance>
18  <myCamera:model rdf:datatype="xsd:string">D300</myCamera:model>
19  <myCamera:weight rdf:parseType="Resource">
20    <rdf:value rdf:datatype="xsd:decimal">0.6</rdf:value>
21    <uom:units rdf:resource="http://www.example.org/units#kg"/>
22  </myCamera:weight>
23 </rdf:Description>
```

- And in that case, any given application cannot go too much further beyond simply aggregating the distributed RDF models.



RDFS Overview

RDFS in Plain English — *continued: A Developer's Guide to the Semantic Web, p 121*

- A common language or shared vocabulary seems to be the key there.
- More specifically, if properties such as `myCamera:model` and `myCamera:weight` are used to describe a camera, that is because somewhere, in some document, someone has defined that these are indeed the predicates we can use to describe it.



RDFS Overview

RDFS in Plain English — *continued: A Developer's Guide to the Semantic Web, p 121*

- There are possibly more terms defined for us to use.
- It is our choice which predicates to use when publishing our own descriptions.
- Therefore, this common language can make sure of one important thing for us.
- Everything we say about a given resource, we have a reason to say it.



RDFS Overview

RDFS in Plain English — *continued: A Developer's Guide to the Semantic Web, p 121*

- Clearly at this stage, what seems to be missing for our RDF documents is such a common language, or a vocabulary.
- Here, classes, subclasses, properties and relations between these classes and properties are defined.



RDFS Overview

RDFS in Plain English — *continued: A Developer's Guide to the Semantic Web, p 121*

- RDFS is such a language that we can use to define a vocabulary.
- Which can then be used to structure the RDF documents we create.



RDFS Overview

RDFS in Plain English — *continued: A Developer's Guide to the Semantic Web, p 121*

- Therefore, in plain English, we can define RDFS as follows:

- RDFS is a language one can use to create a vocabulary.
- Often the created vocabulary is domain-specific.
- When distributed RDF documents are created in this domain, terms from this vocabulary can be used.
- Therefore, everything we say, we have a reason to say it.



RDFS + RDF: One...

A Common Language to Share: A Developer's Guide to the Semantic Web, p 123

- The first important fact about RDFS is that RDFS can be used to define a vocabulary.
- A vocabulary will be a common language everyone can use.



RDFS + RDF: One...

A Common... — continued: *A Developer's Guide to the Semantic Web*, p 123

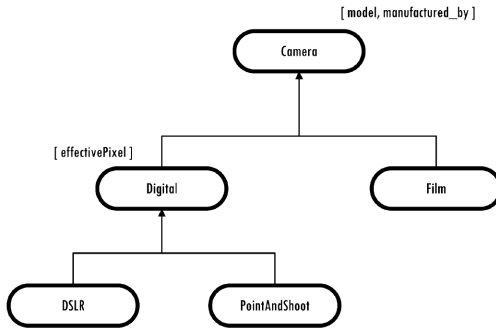


Fig. 4.1 A small vocabulary for the domain of photography

- An oval box is used to represent a specific resource type.



RDFS + RDF: One...

A Common... — continued: *A Developer's Guide to the Semantic Web*, p 123

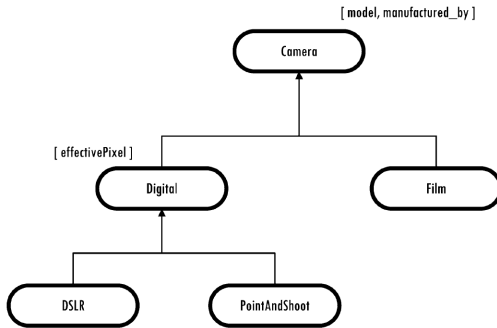


Fig. 4.1 A small vocabulary for the domain of photography

- The arrow from one oval box to another oval box means that the first oval box is a subtype of the second oval box.



RDFS + RDF: One...

A Common... — continued: *A Developer's Guide to the Semantic Web*, p 123

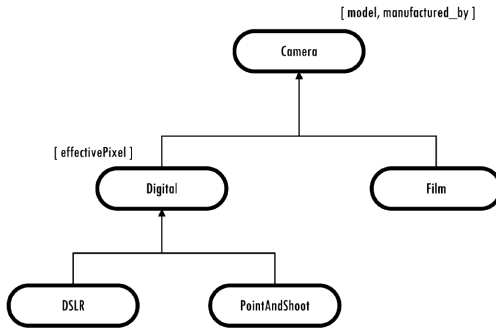


Fig. 4.1 A small vocabulary for the domain of photography

- The properties that one can use to describe a given resource type are included in `[]`, and are placed beside that specific oval box.

RDFS + RDF: One...

A Common... — continued: *A Developer's Guide to the Semantic Web*, p 123

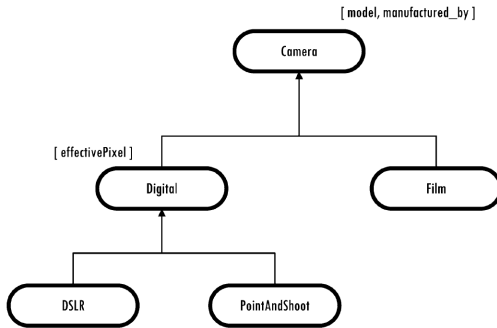


Fig. 4.1 A small vocabulary for the domain of photography

- This simple vocabulary tells us the following fact.



RDFS + RDF: One...

A Common... — continued: A Developer's Guide to the Semantic Web, p 123

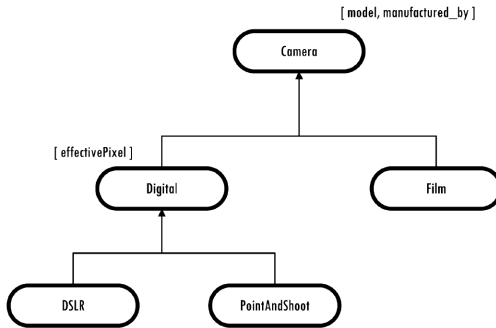


Fig. 4.1 A small vocabulary for the domain of photography

- We have a resource call **Camera**.
- **Digital** and **Film** are its two subresources.



RDFS + RDF: One...

A Common... — continued: *A Developer's Guide to the Semantic Web*, p 123

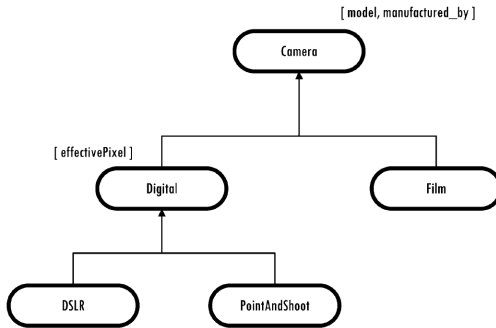


Fig. 4.1 A small vocabulary for the domain of photography

- Resource **Digital** has two subresources, **DSLR** and **PointAndShoot**.



RDFS + RDF: One...

A Common... — continued: A Developer's Guide to the Semantic Web, p 123

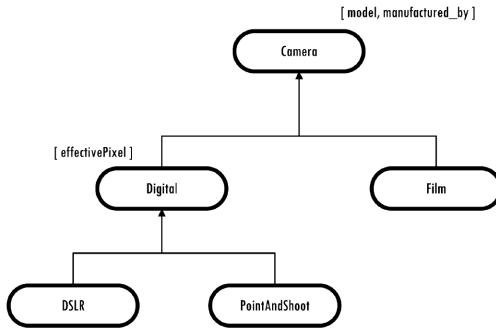


Fig. 4.1 A small vocabulary for the domain of photography

- Resource **Camera** can be described by properties called **model** and **manufactured_by**.



RDFS + RDF: One...

A Common... — continued: *A Developer's Guide to the Semantic Web*, p 123

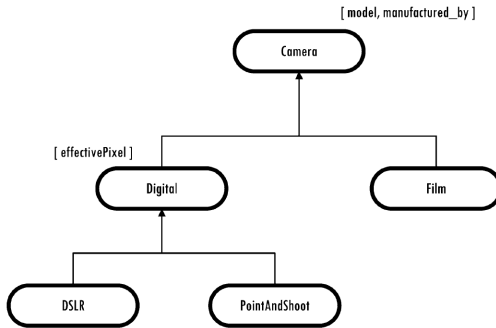


Fig. 4.1 A small vocabulary for the domain of photography

- Resource **Digital** can be described by a property called **effectivePixel**.



RDFS + RDF: One...

A Common... — *continued: A Developer's Guide to the Semantic Web, p 123*

- DSLR is short for Digital Single Lens Reflex.
- This is a type of camera that is more often used by professional photographers.
- It tends to be expensive as well.



RDFS + RDF: One...

A Common... — *continued: A Developer's Guide to the Semantic Web, p 123*

- On the other hand, a Point And Shoot camera is a compact camera.
- It is often used by non-professionals.
- It normally offers functionalities such as autofocus and autoexposure setting.



RDFS + RDF: One...

List 4.1: *A Developer's Guide to the Semantic Web*, p 123

```
1 <?xml version="1.0"?>
2 <rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
3     xmlns:myCamera="http://www.liyangyu.com/camera#">
4
5     <rdf:Description
6         rdf:about="http://www.liyangyu.com/camera#Nikon_D300">
7         <rdf:type
8             rdf:resource="http://www.liyangyu.com/camera#DSLR"/>
9         <myCamera:model>Nikon D300</myCamera:model>
10        <myCamera:manufactured_by
11            rdf:resource="http://www.dbpedia.org/resource/Nikon"/>
12        <myCamera:effectivePixel>12.3</myCamera:effectivePixel>
13    </rdf:Description>
14
15 </rdf:RDF>
```



RDFS + RDF: One...

List 4.1: *A Developer's Guide to the Semantic Web*, p 123

```
5 <rdf:Description
6   rdf:about="http://www.liyangyu.com/camera#Nikon_D300">
7   <rdf:type
8     rdf:resource="http://www.liyangyu.com/camera#DSLR"/>
9   <myCamera:model>Nikon D300</myCamera:model>
10  <myCamera:manufactured_by
11    rdf:resource="http://www.dbpedia.org/resource/Nikon"/>
12  <myCamera:effectivePixel>12.3</myCamera:effectivePixel>
13 </rdf:Description>
```

-
- Resource Camera can be described by properties named manufactured_by and model.
 - Why can we use them to describe Nikon D300, an instance of DSLR, not Camera (lines 9 and 10)?



RDFS + RDF: One...

List 4.1: *A Developer's Guide to the Semantic Web*, p 123

```
5 <rdf:Description
6   rdf:about="http://www.liyangyu.com/camera#Nikon_D300">
7   <rdf:type
8     rdf:resource="http://www.liyangyu.com/camera#DSLR"/>
9   <myCamera:model>Nikon D300</myCamera:model>
10  <myCamera:manufactured_by
11    rdf:resource="http://www.dbpedia.org/resource/Nikon"/>
12  <myCamera:effectivePixel>12.3</myCamera:effectivePixel>
13 </rdf:Description>
```

-
- The reason is really simple.
 - Any property that can be used to describe the base type can also be used to describe any subtype of this base type.



RDFS + RDF: One...

List 4.1: *A Developer's Guide to the Semantic Web*, p 123

```
5 <rdf:Description
6   rdf:about="http://www.liyangyu.com/camera#Nikon_D300">
7   <rdf:type
8     rdf:resource="http://www.liyangyu.com/camera#DSLR"/>
9   <myCamera:model>Nikon D300</myCamera:model>
10  <myCamera:manufactured_by
11    rdf:resource="http://www.dbpedia.org/resource/Nikon"/>
12  <myCamera:effectivePixel>12.3</myCamera:effectivePixel>
13 </rdf:Description>
```

-
- On the other hand, we will not be able to use a term if that term is not defined in the vocabulary.
 - If we have to do so, we will then need to grow the vocabulary accordingly.



RDFS + RDF: One...

A Common... — *continued: A Developer's Guide to the Semantic Web, p 123*

- Now, imagine someone else from the same application domain has come up with another RDF document describing the same camera (or another camera).
- Whatever the resource being described might be, all these documents now share the same terms.
- Notice that when we say the same terms are shared, it is not that all the documents will use exactly the same terms to describe resource.



RDFS + RDF: One...

A Common... — *continued: A Developer's Guide to the Semantic Web, p 123*

- One document might use different properties compared to the other document.
- But all the properties available to use are included in the given vocabulary.



RDFS + RDF: One...

A Common... — *continued: A Developer's Guide to the Semantic Web, p 123*

- The result is that any application that “knows” this vocabulary will be able to process these documents with ease.
- This is an obvious benefit of having a common vocabulary.
- Another important benefit of having a vocabulary defined is to facilitate machine understanding.



RDFS + RDF: One...

Machine Inferencing Based on RDFS: *A Developer's Guide to the Semantic Web*, p 125

- A vocabulary created by using RDFS can facilitate inferencing on the RDF documents that make use of this vocabulary.



RDFS + RDF: One...

List 4.1: *A Developer's Guide to the Semantic Web*, p 125

```
5 <rdf:Description
6   rdf:about="http://www.liyangyu.com/camera#Nikon_D300">
7   <rdf:type
8     rdf:resource="http://www.liyangyu.com/camera#DSLR"/>
9   <myCamera:model>Nikon D300</myCamera:model>
10  <myCamera:manufactured_by
11    rdf:resource="http://www.dbpedia.org/resource/Nikon"/>
12  <myCamera:effectivePixel>12.3</myCamera:effectivePixel>
13 </rdf:Description>
```

- The inferencing for this case is based on line 8, which says the resource identified by `http://www.liyangyu.com/camera#Nikon_D300` is a DSLR.



RDFS + RDF: One...

List 4.1: *A Developer's Guide to the Semantic Web*, p 125

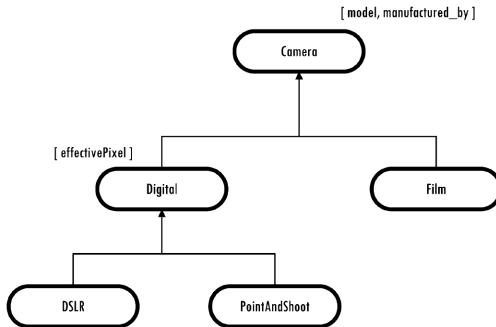


Fig. 4.1 A small vocabulary for the domain of photography

- Given the vocabulary in Fig. 4.1, the following inferences can be made.



RDFS + RDF: One...

List 4.1: *A Developer's Guide to the Semantic Web*, p 125

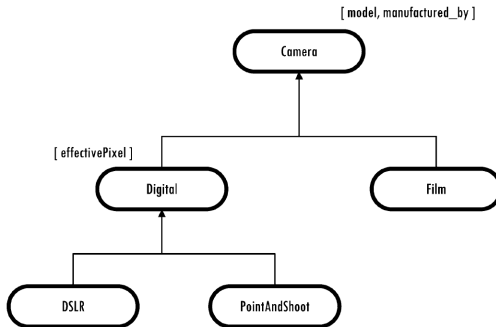


Fig. 4.1 A small vocabulary for the domain of photography

■ Resource

http://www.liyangyu.com/camera#Nikon_D300 is a Digital camera.

RDFS + RDF: One...

List 4.1: *A Developer's Guide to the Semantic Web*, p 125

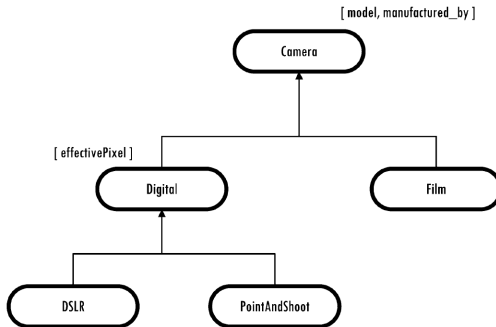


Fig. 4.1 A small vocabulary for the domain of photography

■ Resource

http://www.liyangyu.com/camera#Nikon_D300 is
a Camera.

RDFS + RDF: One...

List 4.1: *A Developer's Guide to the Semantic Web*, p 125

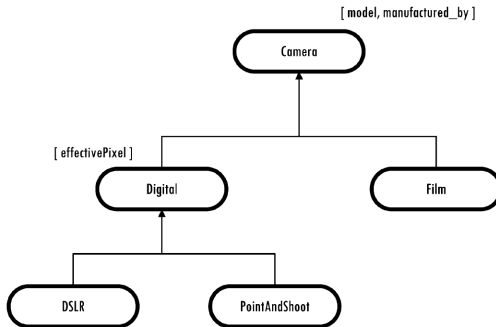


Fig. 4.1 A small vocabulary for the domain of photography

- This is all done by the machine.
- These inferred conclusions can be critical information for many applications.



RDFS + RDF: One...

List 4.1: *A Developer's Guide to the Semantic Web*, p 125

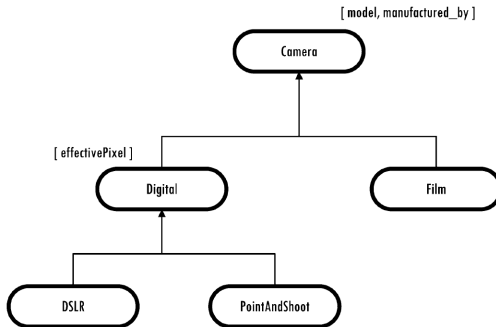


Fig. 4.1 A small vocabulary for the domain of photography

- In fact, a lot more inferencing can be done when a vocabulary is defined.



RDFS Core Elements

The Big Picture: RDFS Vocabulary: *A Developer's Guide to the Semantic Web*, p 126

- RDFS is a collection of terms we can use to define classes and properties for a specific application domain.
- Just like RDF terms and Dublin Core terms, all these RDFS terms are identified by predefined URIs.



RDFS Core Elements

The Big Picture: RDFS Vocabulary — *continued: A Developer's Guide to the Semantic Web*, p 126

- All these URIs share the following leading string:
`http://www.w3.org/2000/01/rdf-schema#`
- By convention, this URI prefix string is associated with namespace prefix `rdfs:`.
- This is typically used in RDF/XML format with the prefix `rdfs`.



RDFS Core Elements

The Big Picture: RDFS Vocabulary — *continued: A Developer's Guide to the Semantic Web, p 126*

- All these RDFS terms can be divided into the following groups based on their purposes:

Classes

- This group includes RDFS terms that can be used to define classes.
- More specifically, the following terms are included here:
 - `rdfs:Resource`,
 - `rdfs:Class`,
 - `rdfs:Literal`,
 - `rdfs:Datatype`.



RDFS Core Elements

The Big Picture: RDFS Vocabulary — *continued: A Developer's Guide to the Semantic Web, p 126*

- All these RDFS terms can be divided into the following groups based on their purposes:

Properties

- This group includes RDFS terms that can be used to define properties.
- The following terms are included:
 - `rdfs:range`,
 - `rdfs:domain`,
 - `rdfs:subClassOf`,
 - `rdfs:subPropertyOf`,
 - `rdfs:label` **and**
 - `rdfs:comment`.



RDFS Core Elements

The Big Picture: RDFS Vocabulary — *continued: A Developer's Guide to the Semantic Web, p 126*

- All these RDFS terms can be divided into the following groups based on their purposes:

Utilities

- This group of RDFS terms are used for miscellaneous purposes.
- This group contains the following terms:
 - `rdfs:seeAlso` and
 - `rdfs:isDefinedBy`.



Basic Syntax and Examples

Defining Classes: *A Developer's Guide to the Semantic Web*, p 127

- `rdfs:Resource` represents the root class.
- Every other class defined using RDFS terms is a subclass of this class.



Basic Syntax and Examples

Defining Classes — *continued: A Developer's Guide to the Semantic Web, p 127*

- In practice, this term is rarely used.
- It mainly acts as a logic root to hold everything together.
- All things described by RDF are instances of class `rdfs:Resource`.
- To define a class in a vocabulary, `rdfs:Class` is used.



Basic Syntax and Examples

List 4.2 Definition of the `Camera` class: *A Developer's Guide to the Semantic Web*, p 127

```
1 <?xml version="1.0"?>
2 <rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
3     xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
4     xmlns:myCamera="http://www.liyangyu.com/camera#">
5
6     <rdf:Description
7         rdf:about="http://www.liyangyu.com/camera#Camera">
8         <rdf:type
9             rdf:resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
10     </rdf:Description>
11
12 </rdf:RDF>
```

- Everything is defined between `<rdf:RDF>` and `</rdf:RDF>`.
- Indicates that this document is either a RDF document or a

RDF schema document.



Basic Syntax and Examples

List 4.2 Definition of the `Camera` class: *A Developer's Guide to the Semantic Web*, p 127

```
1 <?xml version="1.0"?>
2 <rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
3         xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
4         xmlns:myCamera="http://www.liyangyu.com/camera#">
5
6     <rdf:Description
7         rdf:about="http://www.liyangyu.com/camera#Camera">
8         <rdf:type
9             rdf:resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
10     </rdf:Description>
11
12 </rdf:RDF>
```

- Lines 2-4 have defined several namespaces.



Basic Syntax and Examples

List 4.2 Definition of the `Camera` class: *A Developer's Guide to the Semantic Web*, p 127

```
1 <?xml version="1.0"?>
2 <rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
3     xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
4     xmlns:myCamera="http://www.liyangyu.com/camera#">
5
6   <rdf:Description
7     rdf:about="http://www.liyangyu.com/camera#Camera">
8     <rdf:type
9       rdf:resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
10   </rdf:Description>
11
12 </rdf:RDF>
```

- The new one here is the `rdfs` namespace (line 3).
- This includes all the predefined terms in RDF Schema.



Basic Syntax and Examples

List 4.2 Definition of the `Camera` class: *A Developer's Guide to the Semantic Web*, p 127

```
1 <?xml version="1.0"?>
2 <rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
3     xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
4     xmlns:myCamera="http://www.liyangyu.com/camera#">
5
6     <rdf:Description
7         rdf:about="http://www.liyangyu.com/camera#Camera">
8         <rdf:type
9             rdf:resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
10     </rdf:Description>
11
12 </rdf:RDF>
```

-
- Line 4 defines the namespace for our camera vocabulary.



Basic Syntax and Examples

List 4.2 Definition of the `Camera` class: *A Developer's Guide to the Semantic Web*, p 127

```
1 <?xml version="1.0"?>
2 <rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
3     xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
4     xmlns:myCamera="http://www.liyangyu.com/camera#">
5
6     <rdf:Description
7         rdf:about="http://www.liyangyu.com/camera#Camera">
8         <rdf:type
9             rdf:resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
10     </rdf:Description>
11
12 </rdf:RDF>
```

- The key lines are lines 6-10.



Basic Syntax and Examples

List 4.2 Definition of the `Camera` class: *A Developer's Guide to the Semantic Web*, p 127

```
1 <?xml version="1.0"?>
2 <rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
3     xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
4     xmlns:myCamera="http://www.liyangyu.com/camera#">
5
6     <rdf:Description
7         rdf:about="http://www.liyangyu.com/camera#Camera">
8         <rdf:type
9             rdf:resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
10     </rdf:Description>
11
12 </rdf:RDF>
```

- Line 6 defines a new resource by using the term `rdf:Description` from RDF vocabulary.



Basic Syntax and Examples

List 4.2 Definition of the `Camera` class: *A Developer's Guide to the Semantic Web*, p 127

```
1 <?xml version="1.0"?>
2 <rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
3     xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
4     xmlns:myCamera="http://www.liyangyu.com/camera#">
5
6     <rdf:Description
7         rdf:about="http://www.liyangyu.com/camera#Camera">
8         <rdf:type
9             rdf:resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
10     </rdf:Description>
11
12 </rdf:RDF>
```

- This new resource has the following URI:
`http://www.liyangyu.com/camera#Camera.`



Basic Syntax and Examples

List 4.2 Definition of the `Camera` class: *A Developer's Guide to the Semantic Web*, p 127

```
1 <?xml version="1.0"?>
2 <rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
3     xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
4     xmlns:myCamera="http://www.liyangyu.com/camera#">
5
6     <rdf:Description
7         rdf:about="http://www.liyangyu.com/camera#Camera">
8         <rdf:type
9             rdf:resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
10     </rdf:Description>
11
12 </rdf:RDF>
```

- Line 8 specifies the type property of this resource by using RDF term `rdf:type`.



Basic Syntax and Examples

List 4.2 Definition of the `Camera` class: *A Developer's Guide to the Semantic Web*, p 127

```
1 <?xml version="1.0"?>
2 <rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
3     xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
4     xmlns:myCamera="http://www.liyangyu.com/camera#">
5
6     <rdf:Description
7         rdf:about="http://www.liyangyu.com/camera#Camera">
8         <rdf:type
9             rdf:resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
10     </rdf:Description>
11
12 </rdf:RDF>
```

- Its value is another resource.
- Indicated by using RDF term `rdf:resource`.



Basic Syntax and Examples

List 4.2 Definition of the `Camera` class: *A Developer's Guide to the Semantic Web*, p 127

```
1 <?xml version="1.0"?>
2 <rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
3     xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
4     xmlns:myCamera="http://www.liyangyu.com/camera#">
5
6     <rdf:Description
7         rdf:about="http://www.liyangyu.com/camera#Camera">
8         <rdf:type
9             rdf:resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
10     </rdf:Description>
11
12 </rdf:RDF>
```

- This has the following URI:

<http://www.w3.org/2000/01/rdf-schema#Class>.



Basic Syntax and Examples

List 4.2 Definition of the `Camera` class: *A Developer's Guide to the Semantic Web*, p 127

```
1 <?xml version="1.0"?>
2 <rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
3     xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
4     xmlns:myCamera="http://www.liyangyu.com/camera#">
5
6     <rdf:Description
7         rdf:about="http://www.liyangyu.com/camera#Camera">
8         <rdf:type
9             rdf:resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
10     </rdf:Description>
11
12 </rdf:RDF>
```

- Obviously, this URI is a predefined term in RDFS vocabulary.
- Its QName is given by `rdfs:Class`.



Basic Syntax and Examples

List 4.2 Definition of the `Camera` class: *A Developer's Guide to the Semantic Web*, p 127

```
1 <?xml version="1.0"?>
2 <rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
3     xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
4     xmlns:myCamera="http://www.liyangyu.com/camera#">
5
6     <rdf:Description
7         rdf:about="http://www.liyangyu.com/camera#Camera">
8         <rdf:type
9             rdf:resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
10     </rdf:Description>
11
12 </rdf:RDF>
```

- We have defined a new class.



Basic Syntax and Examples

List 4.2 Definition of the `Camera` class: *A Developer's Guide to the Semantic Web*, p 127

```
1 <?xml version="1.0"?>
2 <rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
3     xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
4     xmlns:myCamera="http://www.liyangyu.com/camera#">
5
6     <rdf:Description
7         rdf:about="http://www.liyangyu.com/camera#Camera">
8         <rdf:type
9             rdf:resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
10     </rdf:Description>
11
12 </rdf:RDF>
```

- We can read it as follows.
- Here we declare: this resource, `http://www.liyangyu.com/camera#Camera`, is a class.



Basic Syntax and Examples

List 4.2 Definition of the `Camera` class: *A Developer's Guide to the Semantic Web*, p 127

```
1 <?xml version="1.0"?>
2 <rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
3     xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
4     xmlns:myCamera="http://www.liyangyu.com/camera#">
5
6     <rdf:Description
7         rdf:about="http://www.liyangyu.com/camera#Camera">
8         <rdf:type
9             rdf:resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
10     </rdf:Description>
11
12 </rdf:RDF>
```

- Notice `Camera` class is by default a subclass of `rdfs:Resource`, the root class of all classes.



Basic Syntax and Examples

List 4.2 Definition of the `Camera` class: *A Developer's Guide to the Semantic Web*, p 127

```
1 <?xml version="1.0"?>
2 <rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
3     xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
4     xmlns:myCamera="http://www.liyangyu.com/camera#">
5
6     <rdf:Description
7         rdf:about="http://www.liyangyu.com/camera#Camera">
8         <rdf:type
9             rdf:resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
10     </rdf:Description>
11
12 </rdf:RDF>
```

- In addition, pay attention not to mix together these two terms, `rdfs:Resource` and `rdf:resource`.



Basic Syntax and Examples

List 4.2 Definition of the `Camera` class: *A Developer's Guide to the Semantic Web*, p 127

```
1 <?xml version="1.0"?>
2 <rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
3     xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
4     xmlns:myCamera="http://www.liyangyu.com/camera#">
5
6     <rdf:Description
7         rdf:about="http://www.liyangyu.com/camera#Camera">
8         <rdf:type
9             rdf:resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
10     </rdf:Description>
11
12 </rdf:RDF>
```

- `rdfs:Resource` is a class defined in RDFS.
- `rdf:resource` is simply an XML attribute that goes together with a specific property element.



Basic Syntax and Examples

Defining Classes — *continued: A Developer's Guide to the Semantic Web, p 127*

- Sometimes, you will see `rdf:ID` is used instead of `rdf:about`.
- List 4.3 is equivalent to List 4.2.
- List 4.3 uses `rdf:ID`.



Basic Syntax and Examples

List 4.3 Use `rdf:ID` to define Camera class: *A Developer's Guide to the Semantic Web*, p 127

```
1 <?xml version="1.0"?>
2 <rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
3     xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
4     xmlns:myCamera="http://www.liyangyu.com/camera#"
5     xml:base="http://www.liyangyu.com/camera#">
6
7     <rdf:Description rdf:ID="Camera">
8         <rdf:type
9             rdf:resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
10    </rdf:Description>
11
12 </rdf:RDF>
```



Basic Syntax and Examples

List 4.4 Short form that is equivalent to Lists 4.2 and 4.3: *A Developer's Guide to the Semantic Web*, p 127

```
1 <?xml version="1.0"?>
2 <rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
3     xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
4     xmlns:myCamera="http://www.liyangyu.com/camera#"
5     xml:base="http://www.liyangyu.com/camera#">
6
7   <rdfs:Class rdf:ID="Camera">
8     </rdfs:Class>
9
10 </rdf:RDF>
```



Basic Syntax and Examples

List 4.5 A short form using `rdf:about`: *A Developer's Guide to the Semantic Web*, p 127

```
1 <?xml version="1.0"?>
2 <rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
3     xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
4     xmlns:myCamera="http://www.liyangyu.com/camera#">
5
6     <rdfs:Class rdf:about="http://www.liyangyu.com/camera#Camera">
7     </rdfs:Class>
8
9 </rdf:RDF>
```



Basic Syntax and Examples

List 4.6 Adding class Lens into the vocabulary: *A Developer's Guide to the Semantic Web*, p 127

```
1 <?xml version="1.0"?>
2 <rdf:RDF
3     xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
4     xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
5     xmlns:myCamera="http://www.liyangyu.com/camera#"
6     xml:base="http://www.liyangyu.com/camera#">
7
8     <rdfs:Class rdf:about="http://www.liyangyu.com/camera#Camera">
9     </rdfs:Class>
10
11     <rdfs:Class rdf:about="http://www.liyangyu.com/camera#Lens">
12     </rdfs:Class>
13
14     <rdfs:Class rdf:about="http://www.liyangyu.com/camera#Body">
15     </rdfs:Class>
16
17     <rdfs:Class
18         rdf:about="http://www.liyangyu.com/camera#ValueRange">
19     </rdfs:Class>
20
21 </rdf:RDF>
```

Basic Syntax and Examples

List 4.7 Subclass definitions are added: *A Developer's Guide to the Semantic Web*,
p 127

```
1 <?xml version="1.0"?>
2 <rdf:RDF
3   xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
4   xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
5   xmlns:myCamera="http://www.liyangyu.com/camera#"
6   xml:base="http://www.liyangyu.com/camera#">
7
8   <rdfs:Class rdf:about="http://www.liyangyu.com/camera#Camera">
9     </rdfs:Class>
10
11   <rdfs:Class rdf:about="http://www.liyangyu.com/camera#Lens">
12     </rdfs:Class>
13
14   <rdfs:Class rdf:about="http://www.liyangyu.com/camera#Body">
15     </rdfs:Class>
16
17   <rdfs:Class
18     rdf:about="http://www.liyangyu.com/camera#ValueRange">
19     </rdfs:Class>
20
21   <rdfs:Class
22     rdf:about="http://www.liyangyu.com/camera#Digital">
23     <rdfs:subClassOf rdf:resource="#Camera"/>
24   </rdfs:Class>
25
26   <rdfs:Class rdf:about="http://www.liyangyu.com/camera#Film">
27     <rdfs:subClassOf rdf:resource="#Camera"/>
28   </rdfs:Class>
29
30   <rdfs:Class rdf:about="http://www.liyangyu.com/camera#DSLR">
31     <rdfs:subClassOf rdf:resource="#Digital"/>
32   </rdfs:Class>
33
34   <rdfs:Class rdf:about="http://www.liyangyu.com/camera#PointAndShoot">
35     <rdfs:subClassOf rdf:resource="#Digital"/>
36   </rdfs:Class>
37
38   <rdfs:Class rdf:about="http://www.liyangyu.com/camera#Photographer">
39     <rdfs:subClassOf rdf:resource="http://xmlns.com/foaf/0.1/Person"/>
40   </rdfs:Class>
41
42 </rdf:RDF>
```

Basic Syntax and Examples

List 4.7 Subclass definitions are added: *A Developer's Guide to the Semantic Web*, p 127

```
8 <rdfs:Class rdf:about="http://www.liyangyu.com/camera#Camera">
9 </rdfs:Class>
10
11 <rdfs:Class rdf:about="http://www.liyangyu.com/camera#Lens">
12 </rdfs:Class>
13
14 <rdfs:Class rdf:about="http://www.liyangyu.com/camera#Body">
15 </rdfs:Class>
16
17 <rdfs:Class
18     rdf:about="http://www.liyangyu.com/camera#ValueRange">
19 </rdfs:Class>
20
21 <rdfs:Class
22     rdf:about="http://www.liyangyu.com/camera#Digital">
23     <rdfs:subClassOf rdf:resource="#Camera"/>
24 </rdfs:Class>
```



Basic Syntax and Examples

List 4.7 Subclass definitions are added — *continued: A Developer's Guide to the Semantic Web, p 127*

```
26 <rdfs:Class rdf:about="http://www.liyangyu.com/camera#Film">
27   <rdfs:subClassOf rdf:resource="#Camera"/>
28 </rdfs:Class>
29
30 <rdfs:Class rdf:about="http://www.liyangyu.com/camera#DSLR">
31   <rdfs:subClassOf rdf:resource="#Digital"/>
32 </rdfs:Class>
33
34 <rdfs:Class rdf:about="http://www.liyangyu.com/camera#PointAndShoot">
35   <rdfs:subClassOf rdf:resource="#Digital"/>
36 </rdfs:Class>
37
38 <rdfs:Class rdf:about="http://www.liyangyu.com/camera#Photographer">
39   <rdfs:subClassOf rdf:resource="http://xmlns.com/foaf/0.1/Person"/>
40 </rdfs:Class>
```



Basic Syntax and Examples

Defining Classes — *continued: A Developer's Guide to the Semantic Web, p 127*

- Another important fact about `rdfs:subClassOf` property is that you can use it multiple times when defining a class.
- If you do so, all the base classes introduced by `rdfs:subClassOf` will be ANDed together to create the new class.



Basic Syntax and Examples

Defining Classes — *continued: A Developer's Guide to the Semantic Web, p 127*

- For instance, let us say you have already defined a class called `Journalist`.
- You can now define a new class called `Photojournalist` as follows:

```
1 <rdf:Class
2   rdf:about="http://www.liyangyu.com/camera#Photojournalist">
3   <rdf:subClassOf rdf:resource="#Photographer"/>
4   <rdf:subClassOf rdf:resource="#Journalist"/>
5 </rdf:Class>
```



Basic Syntax and Examples

Defining Classes — *continued: A Developer's Guide to the Semantic Web, p 127*

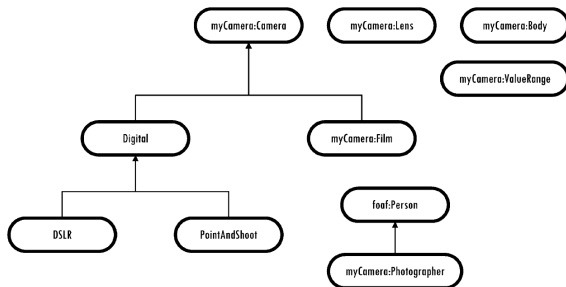


Fig. 4.2 Classes defined for our camera ontology

```
8 <rdfs:Class rdf:about="http://www.liyangyu.com/camera#Camera">
9 </rdfs:Class>
```



Basic Syntax and Examples

Defining Classes — *continued: A Developer's Guide to the Semantic Web, p 127*

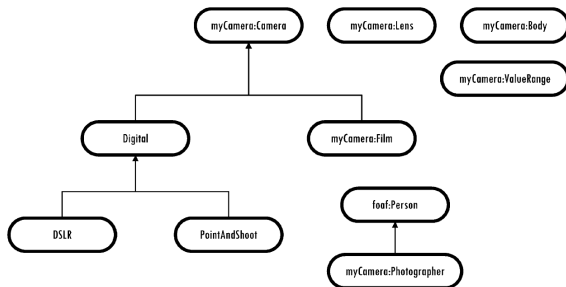


Fig. 4.2 Classes defined for our camera ontology

```
11 <rdfs:Class rdf:about="http://www.liyangyu.com/camera#Lens">
12 </rdfs:Class>
```



Basic Syntax and Examples

Defining Classes — *continued: A Developer's Guide to the Semantic Web, p 127*

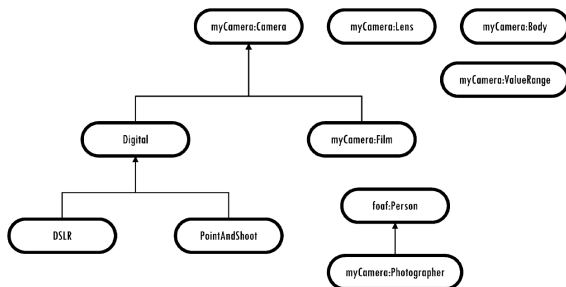


Fig. 4.2 Classes defined for our camera ontology

```
14 <rdfs:Class rdf:about="http://www.liyangyu.com/camera#Body">
15 </rdfs:Class>
```



Basic Syntax and Examples

Defining Classes — *continued: A Developer's Guide to the Semantic Web, p 127*

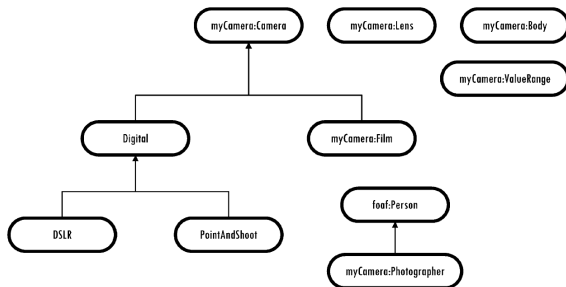


Fig. 4.2 Classes defined for our camera ontology

```
17 <rdfs:Class
18     rdf:about="http://www.liyangyu.com/camera#ValueRange">
19 </rdfs:Class>
```



Basic Syntax and Examples

Defining Classes — *continued: A Developer's Guide to the Semantic Web, p 127*

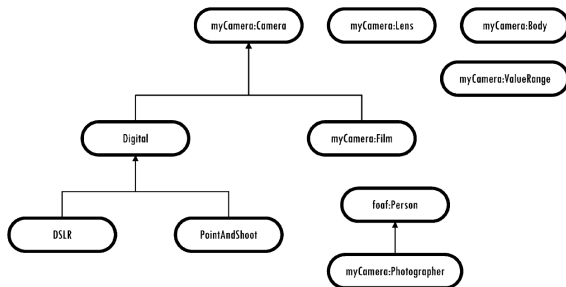


Fig. 4.2 Classes defined for our camera ontology

```
21 <rdf:Class
22     rdf:about="http://www.liyangyu.com/camera#Digital">
23     <rdf:subClassOf rdf:resource="#Camera"/>
24 </rdf:Class>
```



Basic Syntax and Examples

Defining Classes — *continued: A Developer's Guide to the Semantic Web, p 127*

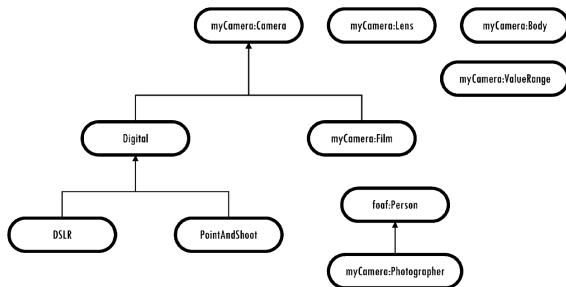


Fig. 4.2 Classes defined for our camera ontology

```
26 <rdfs:Class rdf:about="http://www.liyangyu.com/camera#Film">
27   <rdfs:subClassOf rdf:resource="#Camera"/>
28 </rdfs:Class>
```



Basic Syntax and Examples

Defining Classes — *continued: A Developer's Guide to the Semantic Web, p 127*

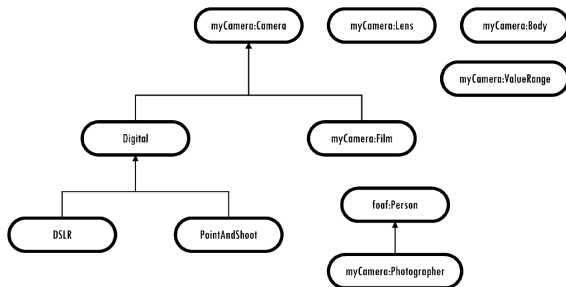


Fig. 4.2 Classes defined for our camera ontology

```
30 <rdfs:Class rdf:about="http://www.liyangyu.com/camera#DSLR">
31   <rdfs:subClassOf rdf:resource="#Digital"/>
32 </rdfs:Class>
```



Basic Syntax and Examples

Defining Classes — *continued: A Developer's Guide to the Semantic Web, p 127*

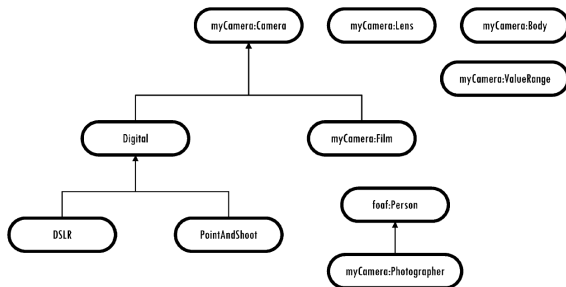


Fig. 4.2 Classes defined for our camera ontology

```
34 <rdfs:Class rdf:about="http://www.liyangyu.com/camera#PointAndShoot">
35   <rdfs:subClassOf rdf:resource="#Digital"/>
36 </rdfs:Class>
```



Basic Syntax and Examples

Defining Classes — *continued: A Developer's Guide to the Semantic Web, p 127*

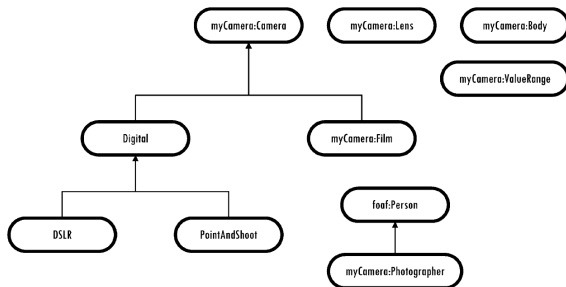


Fig. 4.2 Classes defined for our camera ontology

```
38 <rdfs:Class rdf:about="http://www.liyangyu.com/camera#Photographer">
39   <rdfs:subClassOf rdf:resource="http://xmlns.com/foaf/0.1/Person"/>
40 </rdfs:Class>
```



Basic Syntax and Examples

Defining Properties: *A Developer's Guide to the Semantic Web*, p 133

- To define a property, `rdf:Property` type is used.
- `rdf:about` specifies the URI of the property.
- `rdfs:domain` and `rdfs:range` together indicate how the property should be used.



Basic Syntax and Examples

List 4.8 Define property `owned_by`: *A Developer's Guide to the Semantic Web*, p 133

```
1 <?xml version="1.0"?>
2 <rdf:RDF
3     xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
4     xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
5     xmlns:myCamera="http://www.liyangyu.com/camera#"
6     xml:base="http://www.liyangyu.com/camera#">
7
8     ... // classes, sub-classes definitions as shown in List 4.6
9
10    <rdf:Property rdf:about="http://www.liyangyu.com/camera#owned_by">
11        <rdfs:domain rdf:resource="#DSLR"/>
12        <rdfs:range rdf:resource="#Photographer"/>
13    </rdf:Property>
14
15 </rdf:RDF>
```



Basic Syntax and Examples

List 4.8 Define property `owned_by`: *A Developer's Guide to the Semantic Web*, p 133

```
10 <rdf:Property rdf:about="http://www.liyangyu.com/camera#owned_by">
11   <rdfs:domain rdf:resource="#DSLR"/>
12   <rdfs:range rdf:resource="#Photographer"/>
13 </rdf:Property>
```

-
- We define a property called `owned_by`.
 - It can only be used to describe the characteristics of class `DSLR`.
 - Its possible values can only be instances of class `Photographer`.



Basic Syntax and Examples

List 4.8 Define property `owned_by`: *A Developer's Guide to the Semantic Web, p 133*

```
10 <rdf:Property rdf:about="http://www.liyangyu.com/camera#owned_by">
11   <rdfs:domain rdf:resource="#DSLR"/>
12   <rdfs:range rdf:resource="#Photographer"/>
13 </rdf:Property>
```

subject: DSLR

predicate: `owned_by`

object: Photographer



Basic Syntax and Examples

List 4.8 Define property `owned_by`: *A Developer's Guide to the Semantic Web, p 133*

```
10 <rdf:Property rdf:about="http://www.liyangyu.com/camera#owned_by">
11   <rdfs:domain rdf:resource="#DSLR"/>
12   <rdfs:range rdf:resource="#Photographer"/>
13 </rdf:Property>
```

-
- The new RDFS terms here are `rdfs:domain` and `rdfs:range`.
 - More specifically, property `rdfs:domain` is used to specify which class the property being defined can be used with.



Basic Syntax and Examples

List 4.8 Define property `owned_by`: *A Developer's Guide to the Semantic Web, p 133*

```
10 <rdf:Property rdf:about="http://www.liyangyu.com/camera#owned_by">
11   <rdfs:domain rdf:resource="#DSLR"/>
12   <rdfs:range rdf:resource="#Photographer"/>
13 </rdf:Property>
```

■ `rdfs:domain` is optional.



Basic Syntax and Examples

List 4.8 Define property `owned_by`: *A Developer's Guide to the Semantic Web*, p 133

```
10 <rdf:Property rdf:about="http://www.liyangyu.com/camera#owned_by">
11   <rdfs:domain rdf:resource="#DSLR"/>
12   <rdfs:range rdf:resource="#Photographer"/>
13 </rdf:Property>
```

```
1 <rdf:Property
2   rdf:about="http://www.liyangyu.com/camera#owned_by">
3   <rdfs:range rdf:resource="#Photographer"/>
4 </rdf:Property>
```



Basic Syntax and Examples

Defining Properties — *continued: A Developer's Guide to the Semantic Web, p 133*

```
1 <rdf:Property
2     rdf:about="http://www.liyangyu.com/camera#owned_by">
3   <rdfs:range rdf:resource="#Photographer"/>
4 </rdf:Property>
```

- This means property `owned_by` can be used to describe any class.
- For instance, you can say something like “a `Person` is `owned_by` a `Photographer`”.



Basic Syntax and Examples

Defining Properties — *continued: A Developer's Guide to the Semantic Web, p 133*

```
1 <rdf:Property
2     rdf:about="http://www.liyangyu.com/camera#owned_by">
3   <rdfs:range rdf:resource="#Photographer"/>
4 </rdf:Property>
```

-
- In most cases, this is not what we want.
 - The definition with `rdfs:domain` as shown in List 4.8 is much better.
 - It says that `owned_by` can only be used on the instances of class `DSLR`.



Basic Syntax and Examples

Defining Properties — *continued: A Developer's Guide to the Semantic Web, p 133*

- Notice when defining a property, multiple `rdfs:domain` properties can be specified.
- In that case, we are indicating that the property can be used with a resource that is an instance of every class defined by `rdfs:domain` property.



Basic Syntax and Examples

Defining Properties — *continued: A Developer's Guide to the Semantic Web, p 133*

```
1 <rdf:Property
2     rdf:about="http://www.liyangyu.com/camera#owned_by">
3   <rdfs:domain rdf:resource="#DSLR"/>
4   <rdfs:domain rdf:resource="#PointAndShoot"/>
5   <rdfs:range  rdf:resource="#Photographer"/>
6 </rdf:Property>
```

- This says property `owned_by` can only be used with something that is a `DSLR camera` and a `PointAndShoot camera` at the same time.
- In fact, a `DSLR camera` can be used as a `point-and-shoot camera`, so the above definition does hold.



Basic Syntax and Examples

Defining Properties — *continued: A Developer's Guide to the Semantic Web, p 133*

- For `rdfs:range`, all the above discussion is true.



Basic Syntax and Examples

Defining Properties — *continued: A Developer's Guide to the Semantic Web, p 133*

■ `rdfs:range` is optional.

```
1 <rdf:Property
2     rdf:about="http://www.liyangyu.com/camera#owned_by">
3     <rdfs:domain rdf:resource="#DSLR"/>
4 </rdf:Property>
```



Basic Syntax and Examples

Defining Properties — *continued: A Developer's Guide to the Semantic Web, p 133*

```
1 <rdf:Property
2   rdf:about="http://www.liyangyu.com/camera#owned_by">
3   <rdfs:domain rdf:resource="#DSLR"/>
4 </rdf:Property>
```

- This says property `owned_by` can be use with `DSLR` class, but its value can be anything.
- Therefore, in our RDF document, we can add a statement that says a DSLR camera is owned by another DSLR camera.
- This certainly does not make much sense.



Basic Syntax and Examples

Defining Properties — *continued: A Developer's Guide to the Semantic Web, p 133*

```
1 <rdf:Property
2   rdf:about="http://www.liyangyu.com/camera#owned_by">
3   <rdfs:domain rdf:resource="#DSLR"/>
4 </rdf:Property>
```

-
- Therefore, most likely, we will need to use at least one `rdfs:range` property when defining a property.



Basic Syntax and Examples

Defining Properties — *continued: A Developer's Guide to the Semantic Web, p 133*

- We can also use multiple `rdfs:range` properties.
- Assume we have already defined a class call `Journalist`.

```
1 <rdf:Property
2   rdf:about="http://www.liyangyu.com/camera#owned_by">
3   <rdfs:domain rdf:resource="#DSLR"/>
4   <rdfs:range rdf:resource="#Photographer"/>
5   <rdfs:range rdf:resource="#Journalist"/>
6 </rdf:Property>
```



Basic Syntax and Examples

Defining Properties — *continued: A Developer's Guide to the Semantic Web, p 133*

```
1 <rdf:Property
2     rdf:about="http://www.liyangyu.com/camera#owned_by">
3   <rdfs:domain rdf:resource="#DSLR"/>
4   <rdfs:range rdf:resource="#Photographer"/>
5   <rdfs:range rdf:resource="#Journalist"/>
6 </rdf:Property>
```

-
- This says property `owned_by` can be used to depict DSLRs.
 - Its value has to be someone who is a `Photographer` and `Journalist` at the same time.
 - In other words, this someone has to be a photojournalist.



```
1 <?xml version="1.0"?>
2 <rdf:RDF
3   xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
4   xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
5   xmlns:myCamera="http://www.liyangyu.com/camera#"
6   xml:base="http://www.liyangyu.com/camera#"
7
8   <rdfs:Class
9     rdf:about="http://www.liyangyu.com/camera#Camera*"
10  </rdfs:Class>
11
12   <rdfs:Class
13     rdf:about="http://www.liyangyu.com/camera#Lens*"
14  </rdfs:Class>
15
16   <rdfs:Class
17     rdf:about="http://www.liyangyu.com/camera#Body*"
18  </rdfs:Class>
19
20   <rdfs:Class
21     rdf:about="http://www.liyangyu.com/camera#ValueRange*"
22  </rdfs:Class>
23
24   <rdfs:Class
25     rdf:about="http://www.liyangyu.com/camera#Digital*"
26     <rdfs:subClassOf rdf:resource="#Camera"/*
27  </rdfs:Class>
28
29   <rdfs:Class
30     rdf:about="http://www.liyangyu.com/camera#Film*"
31     <rdfs:subClassOf rdf:resource="#Camera"/*
32  </rdfs:Class>
33
34   <rdfs:Class
35     rdf:about="http://www.liyangyu.com/camera#DSLR*"
36     <rdfs:subClassOf rdf:resource="#Digital"/*
37  </rdfs:Class>
38
39   <rdfs:Class
40     rdf:about="http://www.liyangyu.com/camera#PointAndShoot*"
41     <rdfs:subClassOf rdf:resource="#Digital"/*
42  </rdfs:Class>
43
44   <rdfs:Class
45     rdf:about="http://www.liyangyu.com/camera#Photographer*"
46     <rdfs:subClassOf
47       rdf:resource="http://xmlns.com/foaf/0.1/Person"/*
48  </rdfs:Class>
49
50   <rdf:Property
51     rdf:about="http://www.liyangyu.com/camera#owned_by*"
52     <rdfs:domain rdf:resource="#DSLR"/*
53     <rdfs:range rdf:resource="#Photographer"/*
54  </rdf:Property>
55
56   <rdf:Property
57     rdf:about="http://www.liyangyu.com/camera#manufactured_by*"
58     <rdfs:domain rdf:resource="#Camera"/*
59  </rdf:Property>
```

```
60 <rdf:Property
61   rdf:about="http://www.liyangyu.com/camera#body*"
62 <rdfs:domain rdf:resource="#Camera"/*
63 <rdfs:range rdf:resource="#Body"/*
64 </rdf:Property>
65
66 <rdf:Property
67   rdf:about="http://www.liyangyu.com/camera#lens*"
68 <rdfs:domain rdf:resource="#Camera"/*
69 <rdfs:range rdf:resource="#Lens"/*
70 </rdf:Property>
71
72 <rdf:Property rdf:about="http://www.liyangyu.com/camera#model*"
73 <rdfs:domain rdf:resource="#Camera"/*
74 <rdfs:range
75   rdf:resource="http://www.w3.org/2001/XMLSchema#string"/*
76 </rdf:Property>
77
78 <rdf:Property rdf:about="http://www.liyangyu.com/camera#effectivePixel*"
79 <rdfs:domain rdf:resource="#Digital"/*
80 <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#decimal"/*
81 </rdf:Property>
82
83 <rdf:Property
84   rdf:about="http://www.liyangyu.com/camera#shutterSpeed*"
85 <rdfs:domain rdf:resource="#Body"/*
86 <rdfs:range rdf:resource="#ValueRange"/*
87 </rdf:Property>
88
89 <rdf:Property
90   rdf:about="http://www.liyangyu.com/camera#focalLength*"
91 <rdfs:domain rdf:resource="#Lens"/*
92 <rdfs:range
93   rdf:resource="http://www.w3.org/2001/XMLSchema#string"/*
94 </rdf:Property>
95
96 <rdf:Property
97   rdf:about="http://www.liyangyu.com/camera#aperture*"
98 <rdfs:domain rdf:resource="#Lens"/*
99 <rdfs:range rdf:resource="#ValueRange"/*
100 </rdf:Property>
101
102 <rdf:Property
103   rdf:about="http://www.liyangyu.com/camera#minValue*"
104 <rdfs:domain rdf:resource="#ValueRange"/*
105 <rdfs:range
106   rdf:resource="http://www.w3.org/2001/XMLSchema#float"/*
107 </rdf:Property>
108
109 <rdf:Property
110   rdf:about="http://www.liyangyu.com/camera#maxValue*"
111 <rdfs:domain rdf:resource="#ValueRange"/*
112 <rdfs:range
113   rdf:resource="http://www.w3.org/2001/XMLSchema#float"/*
114 </rdf:Property>
115
116 </rdf:RDF>
```

Basic Syntax and Examples

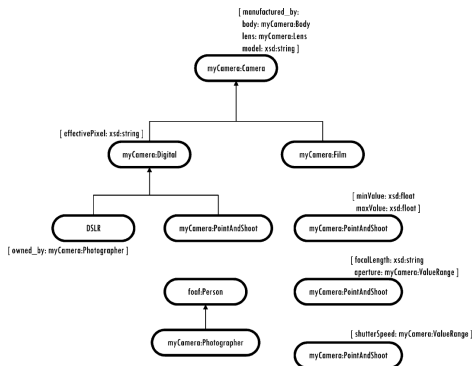


Fig. 4.3 Our camera ontology so far

```

50 <rdf:Property
51   rdf:about="http://www.liyangyu.com/camera#owned_by">
52   <rdfs:domain rdf:resource="#DSLR"/>
53   <rdfs:range rdf:resource="#Photographer"/>
54 </rdf:Property>
  
```



Basic Syntax and Examples

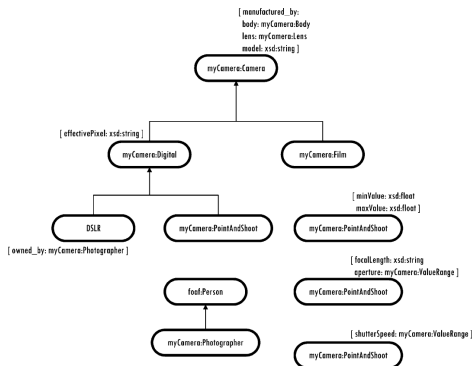


Fig. 4.3 Our camera ontology so far

```
56 <rdf:Property
57     rdf:about="http://www.liyangyu.com/camera#manufactured_by">
58     <rdfs:domain rdf:resource="#Camera"/>
59 </rdf:Property>
```



Basic Syntax and Examples

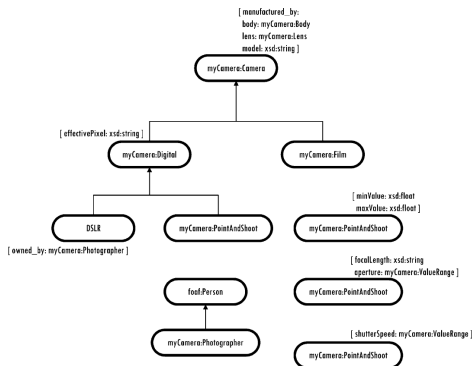


Fig. 4.3 Our camera ontology so far

```

61 <rdf:Property
62     rdf:about="http://www.liyangyu.com/camera#body">
63     <rdfs:domain rdf:resource="#Camera"/>
64     <rdfs:range rdf:resource="#Body"/>
65 </rdf:Property>
  
```



Basic Syntax and Examples

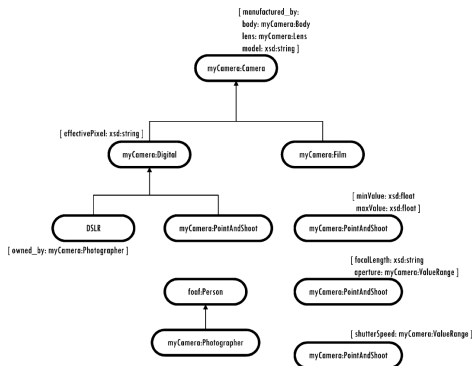


Fig. 4.3 Our camera ontology so far

```
67 <rdf:Property
68     rdf:about="http://www.liyangyu.com/camera#lens">
69     <rdfs:domain rdf:resource="#Camera"/>
70     <rdfs:range rdf:resource="#Lens"/>
71 </rdf:Property>
```



Basic Syntax and Examples

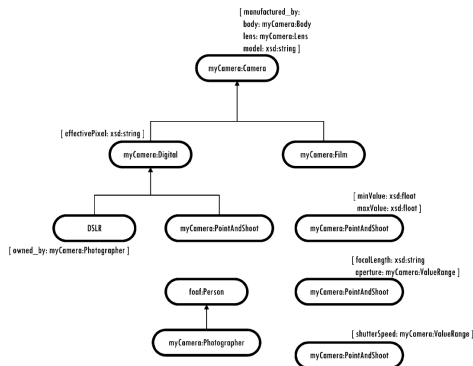


Fig. 4.3 Our camera ontology so far

```
<rdf:Property rdf:about="http://www.liyanguyu.com/camera#model">
  <rdfs:domain rdf:resource="#Camera"/>
  <rdfs:range
    rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
</rdf:Property>
```



Basic Syntax and Examples

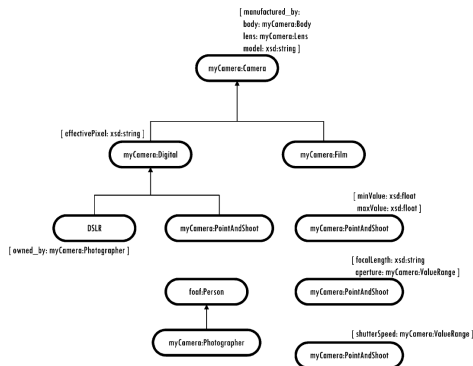


Fig. 4.3 Our camera ontology so far

```

79 <rdf:Property rdf:about="http://www.liyangyu.com/camera#effectivePixel">
80   <rdfs:domain rdf:resource="#Digital"/>
81   <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#decimal"/>
82 </rdf:Property>
  
```



Basic Syntax and Examples

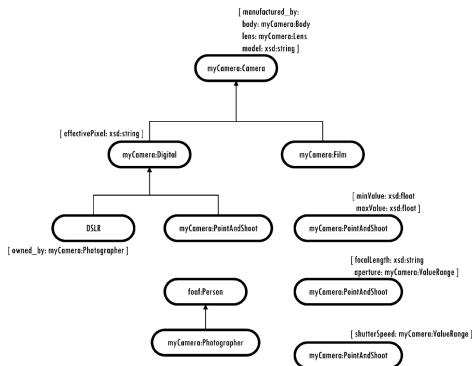


Fig. 4.3 Our camera ontology so far

```
84 <rdf:Property
85     rdf:about="http://www.liyangyu.com/camera#shutterSpeed">
86     <rdfs:domain rdf:resource="#Body"/>
87     <rdfs:range rdf:resource="#ValueRange"/>
88 </rdf:Property>
```



Basic Syntax and Examples

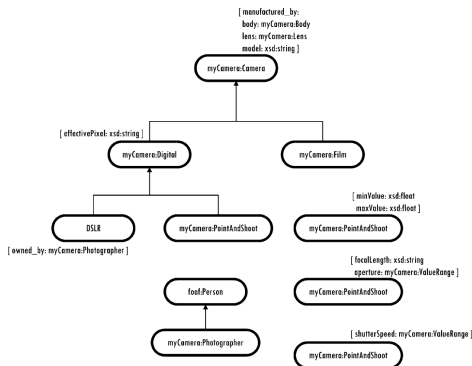


Fig. 4.3 Our camera ontology so far

```
<rdf:Property
  rdf:about="http://www.liyangyu.com/camera#focalLength">
  <rdfs:domain rdf:resource="#Lens"/>
  <rdfs:range
    rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
</rdf:Property>
```



Basic Syntax and Examples

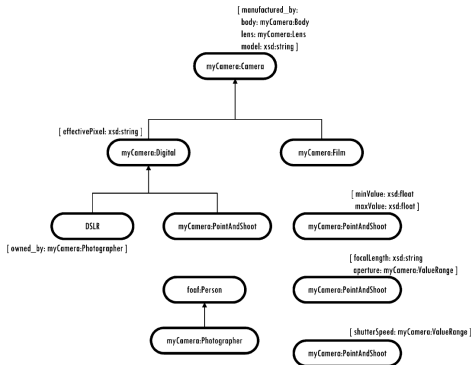


Fig. 4.3 Our camera ontology so far

```

97 <rdf:Property
98     rdf:about="http://www.liyangyu.com/camera#aperture">
99     <rdfs:domain rdf:resource="#Lens"/>
100     <rdfs:range rdf:resource="#ValueRange"/>
101 </rdf:Property>

```

Basic Syntax and Examples

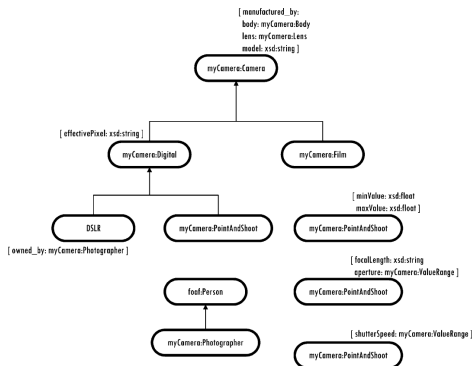


Fig. 4.3 Our camera ontology so far

```
103 <rdf:Property
104     rdf:about="http://www.liyangyu.com/camera#minValue">
105     <rdfs:domain rdf:resource="#ValueRange"/>
106     <rdfs:range
107         rdf:resource="http://www.w3.org/2001/XMLSchema#float"/>
108 </rdf:Property>
```



Basic Syntax and Examples

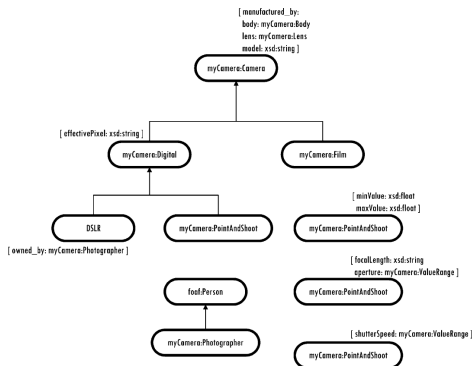


Fig. 4.3 Our camera ontology so far

```
110 <rdf:Property
111     rdf:about="http://www.liyangyu.com/camera#maxValue">
112     <rdfs:domain rdf:resource="#ValueRange"/>
113     <rdfs:range
114         rdf:resource="http://www.w3.org/2001/XMLSchema#float"/>
115 </rdf:Property>
```



Basic Syntax and Examples

More About Properties: *A Developer's Guide to the Semantic Web*, p 139

- Properties are inheritable from base classes to subclasses.



Basic Syntax and Examples

More About Properties — *continued: A Developer's Guide to the Semantic Web, p 139*

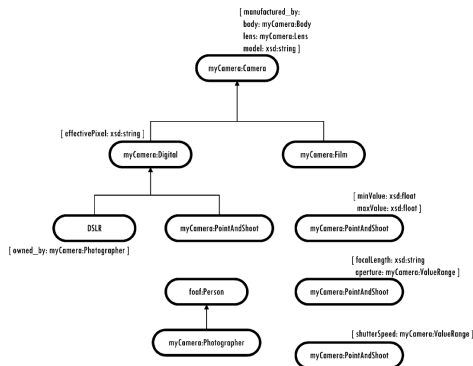


Fig. 4.3 Our camera ontology so far

- More specifically, class `Digital` has a property called `effectivePixel`.
- It also has two subclasses, namely, `DSLR` and `PointAndShoot`.

Basic Syntax and Examples

More About Properties — *continued: A Developer's Guide to the Semantic Web, p 139*

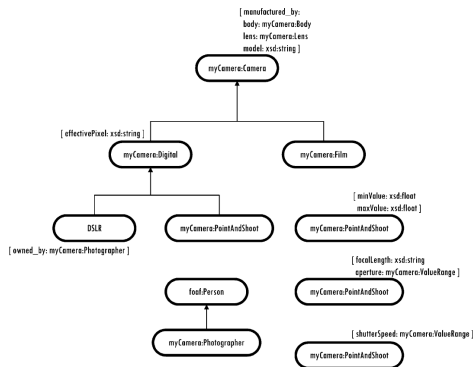


Fig. 4.3 Our camera ontology so far

- Then do these sub-classes also have the property `effectivePixel`?
- In other words, can we use `effectivePixel` to describe a DSLR instance?

Basic Syntax and Examples

More About Properties — *continued: A Developer's Guide to the Semantic Web, p 139*

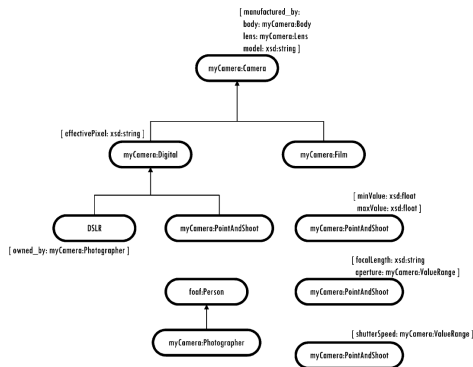


Fig. 4.3 Our camera ontology so far

- The answer is yes, since a subclass always inherits properties from its base class.
- Therefore, class `DSLR` and `PointAndShoot` both have a property called `effectivePixel`.

Basic Syntax and Examples

More About Properties — *continued: A Developer's Guide to the Semantic Web, p 139*

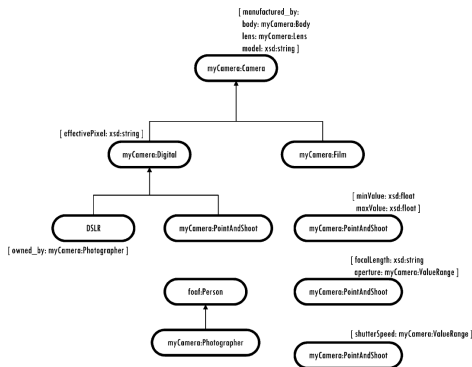


Fig. 4.3 Our camera ontology so far

- In fact, taken one step further, a class always inherits properties from all its base classes.

Basic Syntax and Examples

More About Properties — *continued: A Developer's Guide to the Semantic Web, p 139*

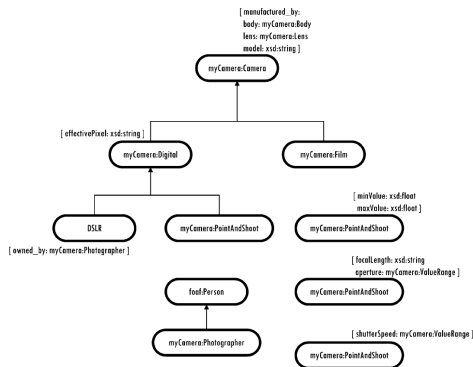


Fig. 4.3 Our camera ontology so far

- For instance, we can use `model` property on class `Camera`.

Basic Syntax and Examples

More About Properties — *continued: A Developer's Guide to the Semantic Web, p 139*

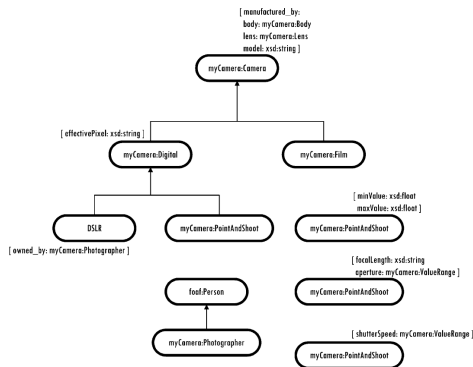


Fig. 4.3 Our camera ontology so far

- And since `Camera` is also a base class of `DSLR` (although not a direct base class), we can then use `model` property on class `DSLR` as well.

Basic Syntax and Examples

More About Properties — *continued: A Developer's Guide to the Semantic Web, p 139*

- The second important issue about property is the subproperty.
- We can define a property to be a subproperty of another property.
- This is done by using `rdfs:subPropertyOf`.



Basic Syntax and Examples

More About Properties — *continued: A Developer's Guide to the Semantic Web, p 139*

- The `model` property describes the “name” of a camera.
- However, the manufacturer could sell the same model using different model names.
- For instance, the same camera sold in North America could have a different model name than the one sold in Asia.



Basic Syntax and Examples

More About Properties — *continued: A Developer's Guide to the Semantic Web, p 139*

- We can define another property, `officialModel`, to be a subproperty of `model`.

```
1 <rdf:Property
2     rdf:about="http://www.liyangyu.com/camera#officialModel">
3   <rdfs:subPropertyOf rdf:resource="#model"/>
4 </rdf:Property>
```



Basic Syntax and Examples

More About Properties — *continued: A Developer's Guide to the Semantic Web, p 139*

```
1 <rdf:Property
2     rdf:about="http://www.liyangyu.com/camera#officialModel">
3   <rdfs:subPropertyOf rdf:resource="#model"/>
4 </rdf:Property>
```

- This declares the property `officialModel` as a specialization of property `model`.
- Property `officialModel` inherits `rdfs:domain` and `rdfs:range` values from its base property `model`.
- However, you can narrow the domain and/or the range as you wish.



Basic Syntax and Examples

More About Properties — *continued: A Developer's Guide to the Semantic Web, p 139*

- We can also use multiple `rdfs:subPropertyOf` when defining a property.
- If we do so, we are declaring the property being defined has to be a subproperty of each of the base properties.



Basic Syntax and Examples

More About Properties — *continued: A Developer's Guide to the Semantic Web, p 139*

- The third issue about property is that we have been using the abbreviated form to define properties.
- It is important to know this since you might see the long form in other documents.



Basic Syntax and Examples

List 4.10 Long form used to define property `owned_by`: *A Developer's Guide to the Semantic Web*, p 139

```
1 <rdf:Description
2   rdf:about="http://www.liyangyu.com/camera#owned_by">
3   <rdf:type
4     rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
5   <rdfs:domain rdf:resource="#DSLR"/>
6   <rdfs:range rdf:resource="#Photographer"/>
7 </rdf:Description>
```



Basic Syntax and Examples

More About Properties — *continued: A Developer's Guide to the Semantic Web, p 139*

- The fourth issue we would like to mention might also be something you have realized already.
- The separation of class definitions and property definitions in our vocabulary.
- Those who are used to the object-oriented world might find this fact uncomfortably strange.



Basic Syntax and Examples

More About Properties — *continued: A Developer's Guide to the Semantic Web, p 139*

- For instance, if we are using any object-oriented language (such as Java or C++), we may define a class called `DigitalCamera`.
- We will then encapsulate several properties to describe a digital camera.
- These properties will be defined at the same time we define the class.
- They are defined in the class scope as its member variables.
- Normally, these properties are not directly visible to the outside world.



Basic Syntax and Examples

More About Properties — *continued: A Developer's Guide to the Semantic Web, p 139*

- For RDF schema, it is quite a different story.
- We define a class, and very often we also indicate its relationships to other classes.



Basic Syntax and Examples

More About Properties — *continued: A Developer's Guide to the Semantic Web, p 139*

- However, this is it.
- We never declare its member variables, i.e., the properties it may have.
- A class is just an entity who may have relationships to other entities.
- What is inside this entity, i.e., its member variables/properties, is simply unknown.



Basic Syntax and Examples

More About Properties — *continued: A Developer's Guide to the Semantic Web, p 139*

- The truth is, we declare properties separately and associate the properties with classes if we wish to do so.
- Properties are never owned by any class.
- They are never local to any class either.



Basic Syntax and Examples

More About Properties — *continued: A Developer's Guide to the Semantic Web, p 139*

- If we do not associate a given property to any class, this property is simply independent and can be used to describe any class.



Basic Syntax and Examples

More About Properties — *continued: A Developer's Guide to the Semantic Web, p 139*

- What is the reason behind this?
- What is the advantage of separating the class definition and property definition?



Basic Syntax and Examples

More About Properties — *continued: A Developer's Guide to the Semantic Web, p 139*

- I can talk about any resource at will.
- I can choose to use an existing URI to identify the resource I am talking about.



Basic Syntax and Examples

More About Properties — *continued: A Developer's Guide to the Semantic Web, p 139*

- The resource I am talking about and the resource already identified by this existing URI are exactly the same thing or concept.
- Everything I have said about this resource is considered to be additional knowledge about that resource.



Basic Syntax and Examples

More About Properties — *continued: A Developer's Guide to the Semantic Web, p 139*

- More specifically, the separation of the class definition and property definition is just an implementation of this rule.
- The final result is that the application we build will have more power to automatically process the distributed information, together with a stronger inferencing engine.



Basic Syntax and Examples

More About Properties — *continued: A Developer's Guide to the Semantic Web, p 139*

- To see this, think about the case where someone else would like to add some new properties into our camera vocabulary.
- And then publish RDF documents that use these newly added properties.



Basic Syntax and Examples

More About Properties — *continued: A Developer's Guide to the Semantic Web, p 139*

- The camera reviewers fits into this example perfectly.
- Those reviewers will have an initial vocabulary they can use to publish their reviews.
- They also enjoy the freedom to come up with new terms to describe a given camera.



Basic Syntax and Examples

More About Properties — *continued: A Developer's Guide to the Semantic Web, p 139*

- Anyone, anywhere and anytime can talk about a resource by adding more properties to it.
- Adding new properties does not disturb any existing application.
- No change is needed to any existing application each time a new property is added.



Basic Syntax and Examples

More About Properties — *continued: A Developer's Guide to the Semantic Web, p 139*

- The reason behind this fact is the separation of class definitions and property definitions.
- If the definition of class were not separate from the definition of property, this would not have been accomplished.



Basic Syntax and Examples

More About Properties — *continued: A Developer's Guide to the Semantic Web, p 139*

- The final point about property is related to an important programming trick that you should know.
- Let us modify `owned_by` property.

```
1 <rdf:Property rdf:ID="owned_by">
2   <rdfs:domain rdf:resource="#Digital"/>
3   <rdfs:domain rdf:resource="#Film"/>
4   <rdfs:range rdf:resource="#Photographer"/>
5 </rdf:Property>
```



Basic Syntax and Examples

More About Properties — *continued: A Developer's Guide to the Semantic Web, p 139*

```
1 <rdf:Property rdf:ID="owned_by">
2   <rdfs:domain rdf:resource="#Digital"/>
3   <rdfs:domain rdf:resource="#Film"/>
4   <rdfs:range rdf:resource="#Photographer"/>
5 </rdf:Property>
```

- If we define the `owned_by` property like this, we are saying `owned_by` is to be used with instances that are both digital cameras and film cameras at the same time.
- Clearly, such a camera has not been invented yet.



Basic Syntax and Examples

More About Properties — *continued: A Developer's Guide to the Semantic Web, p 139*

```
1 <rdf:Property rdf:ID="owned_by">
2   <rdfs:domain rdf:resource="#Digital"/>
3   <rdfs:domain rdf:resource="#Film"/>
4   <rdfs:range rdf:resource="#Photographer"/>
5 </rdf:Property>
```

- Actually, what we wanted to express here is the fact that a photographer can own a digital camera or a film camera or both.
- How do we accomplish this?



Basic Syntax and Examples

More About Properties — *continued: A Developer's Guide to the Semantic Web, p 139*

- Given that a subclass inherits all the properties associated with its base class, we can associate `owned_by` property with the base class.

```
1 <rdf:Property rdf:ID="owned_by">
2   <rdfs:domain rdf:resource="#Camera"/>
3   <rdfs:range rdf:resource="#Photographer"/>
4 </rdf:Property>
```

```
1 <rdf:Property rdf:ID="owned_by">
2   <rdfs:domain rdf:resource="#Digital"/>
3   <rdfs:domain rdf:resource="#Film"/>
4   <rdfs:range rdf:resource="#Photographer"/>
5 </rdf:Property>
```



Basic Syntax and Examples

More About Properties — *continued: A Developer's Guide to the Semantic Web, p 139*

- Given that a subclass inherits all the properties associated with its base class, we can associate `owned_by` property with the base class.

```
1 <rdf:Property rdf:ID="owned_by">
2   <rdfs:domain rdf:resource="#Camera"/>
3   <rdfs:range rdf:resource="#Photographer"/>
4 </rdf:Property>
```

```
1 <rdf:Property rdf:ID="owned_by">
2   <rdfs:domain rdf:resource="#Digital"/>
3   <rdfs:domain rdf:resource="#Film"/>
4   <rdfs:range rdf:resource="#Photographer"/>
5 </rdf:Property>
```



Basic Syntax and Examples

More About Properties — *continued: A Developer's Guide to the Semantic Web, p 139*

```
1 <rdf:Property rdf:ID="owned_by">
2   <rdfs:domain rdf:resource="#Camera"/>
3   <rdfs:range rdf:resource="#Photographer"/>
4 </rdf:Property>
```

-
- Since both `Digital` and `Film` are subclasses of `Camera`, they all inherit property `owned_by`.
 - Now we can use the `owned_by` property with `Digital` class or `Film` class, and this has solved our problem.



Basic Syntax and Examples

More About Properties — *continued: A Developer's Guide to the Semantic Web, p 139*

- `Class` is in the `rdfs` namespace.
- `Property` is in the `rdf` namespace.



Why a class is written as `rdfs:Class` and a property is written as `rdf:Property`?

<http://stackoverflow.com/q/27074089/1337854>



Questions

Tags

Users

Why a class is written as `rdfs:Class` and a property is written as `rdf:Property`



There are numerous examples of this.

asked

4

Among the many, we can consider [this one](#).

viewed



The above one says,

active



- **`rdfs:Class`** declares a resource as a class for other resources.
- **`rdf:Property`** – the class of properties.

Now, my question is, why is a class considered to be defined in RDF schema and property is considered to be simple RDF.

rdf

rdfs

share edit close delete flag

edited Nov 22 '14 at 6:22

asked Nov 22 '14 at 4:37



Masroor

311 ● 3 ● 16

[add a comment](#)

Featu



Hot M

17 M

S

C

82 C

q

v



Why a class is written as `rdfs:Class` and a property is written as `rdf:Property`?

<http://stackoverflow.com/q/27074089/1337854>

1 Answer

active

oldest

votes



4



The reason is purely historic. The RDF model was defined *before* RDF Schema came along, and it already contained a vocabulary term for the notion of a property, so `rdf:Property` is part of the RDF namespace.

The notion of explicit classes was first introduced in RDF Schema, however, so it became `rdfs:Class`.

[share](#) [edit](#) [flag](#)

answered Nov 22 '14 at 8:55



Jeen Broekstra

9,471 ● 2 ● 18 ● 43

I never thought that the reason will be so trivial. Thanks anyway. – [Masroor](#) Nov 22 '14 at 10:47



Basic Syntax and Examples

RDFS Data Types: *A Developer's Guide to the Semantic Web*, p 142

- As we discussed earlier, property `rdfs:range` is used to specify the possible values of a property being declared.
- In some cases, the property being defined can simply have *plain* or *untyped* string as its value, represented by `rdfs:Literal` class contained in RDFS vocabulary.



Basic Syntax and Examples

RDFS Data Types — *continued: A Developer's Guide to the Semantic Web, p 142*

- For example, property `model` could have been defined as follows.
- It could then use any string as its value.

```
1 <rdf:Property
2   rdf:about="http://www.liyangyu.com/camera#model">
3   <rdfs:domain rdf:resource="#Camera"/>
4   <rdfs:range
5     rdf:resource="http://www.w3.org/2001/01/rdf-schema#Literal"/>
6 </rdf:Property>
```



Basic Syntax and Examples

RDFS Data Types — *continued: A Developer's Guide to the Semantic Web, p 142*

- However, using `rdfs:Literal` is not a recommended solution for most cases.
- A better idea is to always provide *typed* values if you can.



Basic Syntax and Examples

RDFS Data Types — *continued: A Developer's Guide to the Semantic Web, p 142*

- For example, we have specified the valid value for the `model` property has to be strings specified by the XML schema.
- Shown in List 4.9.

```
73 <rdf:Property rdf:about="http://www.liyangyu.com/camera#model">
74   <rdfs:domain rdf:resource="#Camera"/>
75   <rdfs:range
76     rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
77 </rdf:Property>
```



Basic Syntax and Examples

RDFS Data Types — *continued: A Developer's Guide to the Semantic Web, p 142*

```
73 <rdf:Property rdf:about="http://www.liyangyu.com/camera#model">
74   <rdfs:domain rdf:resource="#Camera"/>
75   <rdfs:range
76     rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
77 </rdf:Property>
```

- More specifically, the full URI of this data type is given by the following:

`http://www.w3.org/2001/XMLSchema#string`

- We can use this URI directly in our schema without explicitly indicating that it represents a datatype.



Basic Syntax and Examples

RDFS Data Types — *continued: A Developer's Guide to the Semantic Web, p 142*

- However, it is always useful to clearly declare that a given URI represents a datatype.

```
1 <rdf:Property
2   rdf:about="http://www.liyangyu.com/camera#model">
3   <rdfs:domain rdf:resource="#Camera"/>
4   <rdfs:range
5     rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
6 </rdf:Property>
7 <rdfs:Datatype
8   rdf:about="http://www.w3.org/2001/XMLSchema#string"/>
```



Basic Syntax and Examples

RDFS Data Types — *continued: A Developer's Guide to the Semantic Web, p 142*

- The next example shows that using `rdfs:Datatype` is not only a good practice, but it is also necessary in some cases.
- The following could be another definition of `effectivePixel` property.

```
1 <rdf:Property
2   rdf:about="http://www.liyangyu.com/camera#effectivePixel">
3   <rdfs:domain rdf:resource="#Digital"/>
4   <rdfs:range
5     rdf:resource="http://www.liyangyu.com/camera#MegaPixel"/>
6 </rdf:Property>
7 <rdfs:Datatype
8   rdf:about="http://www.liyangyu.com/camera#MegaPixel">
9   <rdfs:subClassOf
10     rdf:resource="http://www.w3.org/2001/XMLSchema#decimal"/>
11 </rdfs:Datatype>
```



Basic Syntax and Examples

RDFS Data Types — *continued: A Developer's Guide to the Semantic Web, p 142*

```
1 <rdf:Property
2     rdf:about="http://www.liyangyu.com/camera#effectivePixel">
3   <rdfs:domain rdf:resource="#Digital"/>
4   <rdfs:range
5     rdf:resource="http://www.liyangyu.com/camera#MegaPixel"/>
6 </rdf:Property>
7 <rdfs:Datatype
8     rdf:about="http://www.liyangyu.com/camera#MegaPixel">
9   <rdfs:subClassOf
10     rdf:resource="http://www.w3.org/2001/XMLSchema#decimal"/>
11 </rdfs:Datatype>
```

- When a RDF schema parser reaches the above code, it first concludes the property `effectivePixel`'s value should come from a resource with the following URI.
`http://www.liyangyu.com/camera#MegaPixel`



Basic Syntax and Examples

RDFS Data Types — *continued: A Developer's Guide to the Semantic Web, p 142*

```
1 <rdf:Property
2     rdf:about="http://www.liyangyu.com/camera#effectivePixel">
3   <rdfs:domain rdf:resource="#Digital"/>
4   <rdfs:range
5     rdf:resource="http://www.liyangyu.com/camera#MegaPixel"/>
6 </rdf:Property>
7 <rdfs:Datatype
8     rdf:about="http://www.liyangyu.com/camera#MegaPixel">
9   <rdfs:subClassOf
10     rdf:resource="http://www.w3.org/2001/XMLSchema#decimal"/>
11 </rdfs:Datatype>
```

- And once it reaches the next couple of lines, it realizes this URI is in fact identifying an `rdfs:Datatype` instance.
- This has a base class given by this URI,
`http://www.w3.org/2001/XMLSchema#decimal`.



Basic Syntax and Examples

RDFS Data Types — *continued: A Developer's Guide to the Semantic Web, p 142*

```
1 <rdf:Property
2     rdf:about="http://www.liyangyu.com/camera#effectivePixel">
3   <rdfs:domain rdf:resource="#Digital"/>
4   <rdfs:range
5     rdf:resource="http://www.liyangyu.com/camera#MegaPixel"/>
6 </rdf:Property>
7 <rdfs:Datatype
8     rdf:about="http://www.liyangyu.com/camera#MegaPixel">
9   <rdfs:subClassOf
10     rdf:resource="http://www.w3.org/2001/XMLSchema#decimal"/>
11 </rdfs:Datatype>
```

- The parser then concludes that `effectivePixel` should always use a typed literal as its value.



Basic Syntax and Examples

RDFS Data Types — *continued: A Developer's Guide to the Semantic Web, p 142*

- Notice when `rdfs:Datatype` is used in our RDF schema document to indicate a data type, the corresponding RDF instance statements should then use `rdf:datatype` property.

```
1 <model rdf:datatype="http://www.w3.org/2001/XMLSchema#string">
2   Nikon_D300
3 </model>
4 <effectivePixel
5   rdf:datatype="http://www.liyangyu.com/camera#MegaPixel">
6   12.3
7 </effectivePixel>
```



Basic Syntax and Examples

RDFS Data Types — *continued: A Developer's Guide to the Semantic Web, p 142*

- A related topic here is the usage of `rdfs:XMLLiteral`.
- Remember, in most cases, its use should be avoided.
- To make our discussion complete, let us briefly talk about the reason here.



Basic Syntax and Examples

RDFS Data Types — *continued: A Developer's Guide to the Semantic Web, p 142*

- `rdfs:XMLLiteral` denotes a well-formed XML string.
- It is always used together with `rdf:parseType="Literal"`.
- For instance, if you used `rdfs:XMLLiteral` in an RDF schema document to define some property, the RDF statements which describe an instance of this property must use `rdf:parseType="Literal"`.



Basic Syntax and Examples

RDFS Data Types — *continued: A Developer's Guide to the Semantic Web, p 142*

- We have defined a new property called `features`.

```
1 <rdf:Property rdf:ID="features">
2   <rdfs:domain rdf:resource="#Digital"/>
3   <rdfs:range rdf:resource=
4     "http://www.w3.org/1999/02/22-rdf-syntax-ns#XMLLiteral"/>
5 </rdf:Property>
```



Basic Syntax and Examples

RDFS Data Types — *continued: A Developer's Guide to the Semantic Web, p 142*

■ An example RDF statement.

```
1 <features rdf:parseType="Literal">
2   Nikon D300 is <bold>good!</bold>, also, . . .
3 </features>
```



Basic Syntax and Examples

RDFS Data Types — *continued: A Developer's Guide to the Semantic Web, p 142*

```
1 <features rdf:parseType="Literal">  
2   Nikon D300 is <bold>good!</bold>, also, . . .  
3 </features>
```

-
- Notice the usage of `rdf:parseType="Literal"`.
 - This indicates the value here is a well-formed XML content.



Basic Syntax and Examples

RDFS Data Types — *continued: A Developer's Guide to the Semantic Web, p 142*

```
1 <features rdf:parseType="Literal">  
2   Nikon D300 is <bold>good!</bold>, also, . . .  
3 </features>
```

-
- Now, notice that although the content is a well-formed XML content, it does not have the resource/property/value structure in general.
 - This structure is one of the main reasons why a given application can understand the content.



Basic Syntax and Examples

RDFS Data Types — *continued: A Developer's Guide to the Semantic Web, p 142*

```
1 <features rdf:parseType="Literal">  
2   Nikon D300 is <bold>good!</bold>, also, . . .  
3 </features>
```

-
- Therefore, if we use XML paragraph as the value of some property, we have to accept the fact that no tools will be able to understand its meaning well.
 - So, avoid using `XMLLiteral` if you can.



Basic Syntax and Examples

RDFS Utility Vocabulary: *A Developer's Guide to the Semantic Web*, p 144

- `rdfs:seeAlso`
- `rdfs:isDefinedBy`
- `rdfs:label`
- `rdfs:comment`



Basic Syntax and Examples

RDFS Utility Vocabulary — *continued: A Developer's Guide to the Semantic Web, p 144*

- `rdfs:seeAlso` is a property that can be used on any resource.
- It indicates another resource may provide additional information about the given resource.



Basic Syntax and Examples

List 4.11 Example of using `rdfs:seeAlso`: *A Developer's Guide to the Semantic Web*, p 144

```
1 <?xml version="1.0"?>
2 <rdf:RDF
3     xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
4     xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
5     xmlns:myCamera="http://www.liyangyu.com/camera#">
6
7     <rdf:Description
8         rdf:about="http://www.liyangyu.com/camera#Nikon_D300">
9         <rdf:type
10             rdf:resource="http://www.liyangyu.com/camera#DSLR"/>
11         <rdfs:seeAlso
12             rdf:resource="http://dbpedia.org/resource/Nikon_D300"/>
13     </rdf:Description>
14
15 </rdf:RDF>
```



Basic Syntax and Examples

List 4.11 Example of using `rdfs:seeAlso`: *A Developer's Guide to the Semantic Web*, p 144

```
7 <rdf:Description
8   rdf:about="http://www.liyangyu.com/camera#Nikon_D300">
9   <rdf:type
10     rdf:resource="http://www.liyangyu.com/camera#DSLR"/>
11   <rdfs:seeAlso
12     rdf:resource="http://dbpedia.org/resource/Nikon_D300"/>
13 </rdf:Description>
```

- Line 11-12 says this: to understand more about the resource identified by this URI,
http://www.liyangyu.com/camera#Nikon_D300
you can take a look at the resource identified at this URI:
http://dbpedia.org/resource/Nikon_D300.



Basic Syntax and Examples

List 4.11 Example of using `rdfs:seeAlso`: *A Developer's Guide to the Semantic Web*, p 144

```
7 <rdf:Description
8   rdf:about="http://www.liyangyu.com/camera#Nikon_D300">
9   <rdf:type
10    rdf:resource="http://www.liyangyu.com/camera#DSLR"/>
11   <rdfs:seeAlso
12    rdf:resource="http://dbpedia.org/resource/Nikon_D300"/>
13 </rdf:Description>
```

- Notice that `rdfs:seeAlso` has no formal semantics defined.
- In real application, it only implies the fact that these two URIs are somehow related to each other.
- It is then up to the application to decide how to handle this situation.



Basic Syntax and Examples

List 4.11 Example of using `rdfs:seeAlso`: *A Developer's Guide to the Semantic Web*, p 144

```
7 <rdf:Description
8   rdf:about="http://www.liyangyu.com/camera#Nikon_D300">
9   <rdf:type
10    rdf:resource="http://www.liyangyu.com/camera#DSLR"/>
11   <rdfs:seeAlso
12    rdf:resource="http://dbpedia.org/resource/Nikon_D300"/>
13 </rdf:Description>
```

- For our case, recall the above URI is created by DBpedia to represent exactly the same resource, namely, Nikon D300 camera.
- Therefore, these two URIs are considered to be URI aliases.
- An application can act accordingly.



Basic Syntax and Examples

List 4.11 Example of using `rdfs:seeAlso`: *A Developer's Guide to the Semantic Web*, p 144

```
7 <rdf:Description
8   rdf:about="http://www.liyangyu.com/camera#Nikon_D300">
9   <rdf:type
10    rdf:resource="http://www.liyangyu.com/camera#DSLR"/>
11   <rdfs:seeAlso
12    rdf:resource="http://dbpedia.org/resource/Nikon_D300"/>
13 </rdf:Description>
```

- For example, the application can retrieve an RDF document from the second URI and collect more information from this new document.
- This is a typical example of information aggregation based on URI aliases.



Basic Syntax and Examples

List 4.11 Example of using `rdfs:seeAlso`: *A Developer's Guide to the Semantic Web*, p 144

```
7 <rdf:Description
8   rdf:about="http://www.liyangyu.com/camera#Nikon_D300">
9   <rdf:type
10    rdf:resource="http://www.liyangyu.com/camera#DSLR"/>
11   <rdfs:seeAlso
12    rdf:resource="http://dbpedia.org/resource/Nikon_D300"/>
13 </rdf:Description>
```

-
- This is also one of the key concepts in the world of linked data.



Basic Syntax and Examples

RDFS Utility Vocabulary — *continued: A Developer's Guide to the Semantic Web, p 144*

- `rdfs:isDefinedBy` is quite similar to `rdfs:seeAlso`.
- It is actually a `rdfs:subPropertyOf` of `rdfs:seeAlso`.
- It is intended to specify the primary source of information about a given resource.



Basic Syntax and Examples

RDFS Utility Vocabulary — *continued: A Developer's Guide to the Semantic Web, p 144*

- For example, the following statement,
`subject rdfs:isDefinedBy object`
says that the `subject` resource is defined by the `object` resource.
- More specifically, this `object` resource is supposed to be an original or authoritative description of the resource.



Basic Syntax and Examples

RDFS Utility Vocabulary — *continued: A Developer's Guide to the Semantic Web, p 144*

- The last two properties you may encounter in documents are `rdfs:label` and `rdfs:comment`.
- `rdfs:label` is used to provide a class/property name for human eyes.
- Similarly, `rdfs:comment` provides a human-readable description of the property/class being defined.



Basic Syntax and Examples

List 4.12 Example of using `rdfs:label` and `rdfs:comment`: *A Developer's Guide to the Semantic Web*, p 144

```
1 <rdf:Property rdf:ID="officialModel">
2   <rdfs:subPropertyOf rdf:resource="#model"/>
3   <rdfs:label xml:lang="EN">officialModelName</rdfs:label>
4   <rdfs:comment xml:lang="EN">
5     this is the official name of the camera.
6     the manufacturer may use different names when
7     the camera is sold in different regions/countries.
8   </rdfs:comment>
9 </rdf:Property>
```



Summary So Far

Our Camera Vocabulary, List 4.13: *A Developer's Guide to the Semantic Web*, p 146

```
1 <?xml version="1.0"?>
2 <rdf:RDF
3     xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
4     xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
5     xmlns:myCamera="http://www.liyangyu.com/camera#"
6     xml:base="http://www.liyangyu.com/camera#">
7
8     <rdfs:Class rdf:about="http://www.liyangyu.com/camera#Camera">
9     </rdfs:Class>
10
11     <rdfs:Class rdf:about="http://www.liyangyu.com/camera#Lens">
12     </rdfs:Class>
13
14     <rdfs:Class rdf:about="http://www.liyangyu.com/camera#Body">
15     </rdfs:Class>
16
17     <rdfs:Class rdf:about="http://www.liyangyu.com/camera#ValueRange">
18     </rdfs:Class>
```



Summary So Far

List 4.13 — *continued: A Developer's Guide to the Semantic Web, p 146*

```
20 <rdfs:Class rdf:about="http://www.liyangyu.com/camera#Digital">
21   <rdfs:subClassOf rdf:resource="#Camera"/>
22 </rdfs:Class>
23
24 <rdfs:Class rdf:about="http://www.liyangyu.com/camera#Film">
25   <rdfs:subClassOf rdf:resource="#Camera"/>
26 </rdfs:Class>
27
28 <rdfs:Class rdf:about="http://www.liyangyu.com/camera#DSLR">
29   <rdfs:subClassOf rdf:resource="#Digital"/>
30 </rdfs:Class>
31
32 <rdfs:Class rdf:about="http://www.liyangyu.com/camera#PointAndShoot">
33   <rdfs:subClassOf rdf:resource="#Digital"/>
34 </rdfs:Class>
35
36 <rdfs:Class rdf:about="http://www.liyangyu.com/camera#Photographer">
37   <rdfs:subClassOf rdf:resource="http://xmlns.com/foaf/0.1/Person"/>
38 </rdfs:Class>
```



Summary So Far

List 4.13 — *continued: A Developer's Guide to the Semantic Web, p 146*

```
40 <rdf:Property rdf:about="http://www.liyangyu.com/camera#owned_by">
41   <rdfs:domain rdf:resource="#DSLR"/>
42   <rdfs:range rdf:resource="#Photographer"/>
43 </rdf:Property>
44
45 <rdf:Property rdf:about="http://www.liyangyu.com/camera#manufactured">
46   <rdfs:domain rdf:resource="#Camera"/>
47 </rdf:Property>
48
49 <rdf:Property rdf:about="http://www.liyangyu.com/camera#body">
50   <rdfs:domain rdf:resource="#Camera"/>
51   <rdfs:range rdf:resource="#Body"/>
52 </rdf:Property>
53
54 <rdf:Property rdf:about="http://www.liyangyu.com/camera#lens">
55   <rdfs:domain rdf:resource="#Camera"/>
56   <rdfs:range rdf:resource="#Lens"/>
57 </rdf:Property>
```



Summary So Far

List 4.13 — *continued: A Developer's Guide to the Semantic Web, p 146*

```
59 <rdf:Property
60     rdf:about="http://www.liyangyu.com/camera#model">
61     <rdfs:domain rdf:resource="#Camera"/>
62     <rdfs:range
63         rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
64 </rdf:Property>
65 <rdfs:Datatype
66     rdf:about="http://www.w3.org/2001/XMLSchema#string"/>
67
68 <rdf:Property
69     rdf:about="http://www.liyangyu.com/camera#effectivePixel">
70     <rdfs:domain rdf:resource="#Digital"/>
71     <rdfs:range
72         rdf:resource="http://www.liyangyu.com/camera#MegaPixel"/>
73 </rdf:Property>
74 <rdfs:Datatype
75     rdf:about="http://www.liyangyu.com/camera#MegaPixel">
76     <rdfs:subClassOf
77         rdf:resource="http://www.w3.org/2001/XMLSchema#decimal"/>
78 </rdfs:Datatype>
```



Summary So Far

List 4.13 — *continued: A Developer's Guide to the Semantic Web, p 146*

```
80 <rdf:Property
81     rdf:about="http://www.liyangyu.com/camera#shutterSpeed">
82     <rdfs:domain rdf:resource="#Body"/>
83     <rdfs:range rdf:resource="#ValueRange"/>
84 </rdf:Property>
85
86 <rdf:Property
87     rdf:about="http://www.liyangyu.com/camera#focalLength">
88     <rdfs:domain rdf:resource="#Lens"/>
89     <rdfs:range
90         rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
91 </rdf:Property>
92 <rdfs:Datatype
93     rdf:about="http://www.w3.org/2001/XMLSchema#string"/>
94
95 <rdf:Property
96     rdf:about="http://www.liyangyu.com/camera#aperture">
97     <rdfs:domain rdf:resource="#Lens"/>
98     <rdfs:range rdf:resource="#ValueRange"/>
99 </rdf:Property>
```



Summary So Far

List 4.13 — *continued: A Developer's Guide to the Semantic Web, p 146*

```
101 <rdf:Property
102     rdf:about="http://www.liyangyu.com/camera#minValue">
103     <rdfs:domain rdf:resource="#ValueRange"/>
104     <rdfs:range
105         rdf:resource="http://www.w3.org/2001/XMLSchema#float"/>
106 </rdf:Property>
107 <rdfs:Datatype
108     rdf:about="http://www.w3.org/2001/XMLSchema#float"/>
109
110 <rdf:Property
111     rdf:about="http://www.liyangyu.com/camera#maxValue">
112     <rdfs:domain rdf:resource="#ValueRange"/>
113     <rdfs:range
114         rdf:resource="http://www.w3.org/2001/XMLSchema#float"/>
115 </rdf:Property>
116 <rdfs:Datatype
117     rdf:about="http://www.w3.org/2001/XMLSchema#float"/>
118
119 </rdf:RDF>
```



Summary So Far

Where Is the Knowledge?: *A Developer's Guide to the Semantic Web*, p 150

- We have created a simply camera vocabulary by using some predefined classes and properties from RDF Schema.
- So how is the knowledge encoded in this vocabulary?



Summary So Far

Where Is the Knowledge? — *continued: A Developer's Guide to the Semantic Web*,
p 150

- And here is the answer.
- In a given vocabulary, the meaning of a term is expressed and understood by defining the following.
 - all the properties that can be used on it, and
 - the types of those objects that can be used as the values of these properties.



Summary So Far

Where Is the Knowledge? — *continued: A Developer's Guide to the Semantic Web*,
p 150

- For example, let us take a look at term `Camera`.



Summary So Far

Where Is the Knowledge? — *continued: A Developer's Guide to the Semantic Web*,
p 150

- As far as any application is concerned, a `Camera` is something like this:
 - It is a class.
 - We can use property `manufactured_by` on it, and any resource can be the value of this property.
 - We can use property `body` on it, with a `Body` instance as this property's value.
 - We can use property `lens` on it, with a `Lens` instance as this property's value.
 - We can use property `model` on it, with an XML string as this property's value.



Summary So Far

Where Is the Knowledge? — *continued: A Developer's Guide to the Semantic Web, p 150*

- And similarly, for any application, a Digital camera is something like this:
 - It is a class.
 - We can use property `manufacturedby` on it, and any resource can be the value of this property.
 - We can use property `body` on it, with a `Body` instance as this property's value.
 - We can use property `lens` on it, with a `Lens` instance as this property's value.
 - We can use property `model` on it, with an XML string as this property's value.
 - We can use property `effectivePixel` on it, with an XML decimal as this property's value.



Summary So Far

Where Is the Knowledge? — *continued: A Developer's Guide to the Semantic Web*,
p 150

- You can come up with the meaning of the word DSLR just as above.



Summary So Far

Where Is the Knowledge? — *continued: A Developer's Guide to the Semantic Web, p 150*

- How can the knowledge be used and understood by applications?



The Concept of Ontology

A Developer's Guide to the Semantic Web, p 150

- Ontology plays a critical role for the Semantic Web.
- It is necessary to understand ontology in order to fully appreciate the idea of the Semantic Web.



The Concept of Ontology — *continued*

A Developer's Guide to the Semantic Web, p 150

- Its concept, however, seems quite abstract and hard to grasp from the beginning.
- It does take a while to get used to, but the more you know it, the more you see the value of it.



The Concept of Ontology

What Is Ontology: *A Developer's Guide to the Semantic Web*, p 151

- We have already built an ontology.
- List 4.13 is in fact a tiny ontology in the domain of photography.



The Concept of Ontology

What Is Ontology — *continued: A Developer's Guide to the Semantic Web, p 151*

- There are many definitions of ontology.
- Perhaps each single one of these definitions starts from a different angle of view.



The Concept of Ontology

What Is Ontology — *continued: A Developer's Guide to the Semantic Web, p 151*

- And some of these definitions can be confusing as well.
- For example, the most popular definition of ontology is “ontology is a formalization of a conceptualization”!



The Concept of Ontology

What Is Ontology — *continued: A Developer's Guide to the Semantic Web, p 151*

- For us, in the world of the Semantic Web, the definition presented in the W3Cs OWL Use Cases and Requirements Documents is good enough.

“An ontology formally defines a common set of terms that are used to describe and represent a domain . . .”

“An ontology defines the terms used to describe and represent an area of knowledge.”



The Concept of Ontology

What Is Ontology — *continued: A Developer's Guide to the Semantic Web, p 151*

“An ontology formally defines a common set of terms that are used to describe and represent a domain . . .”

“An ontology defines the terms used to describe and represent an area of knowledge.”

-
- Several things need to be made clear from this definition.



The Concept of Ontology

What Is Ontology — *continued: A Developer's Guide to the Semantic Web, p 151*

“An ontology formally defines a common set of terms that are used to describe and represent a domain . . .”

“An ontology defines the terms used to describe and represent an area of knowledge.”

-
- First of all, ontology is domain-specific.
 - It is used to describe and represent an area of knowledge.



The Concept of Ontology

What Is Ontology — *continued: A Developer's Guide to the Semantic Web, p 151*

“An ontology formally defines a common set of terms that are used to describe and represent a domain . . .”

“An ontology defines the terms used to describe and represent an area of knowledge.”

-
- A domain is simply a specific subject area or area of knowledge, such as the area of photography, medicine, real estate, education, etc.



The Concept of Ontology

What Is Ontology — *continued: A Developer's Guide to the Semantic Web, p 151*

“An ontology formally defines a common set of terms that are used to describe and represent a domain . . .”

“An ontology defines the terms used to describe and represent an area of knowledge.”

-
- Second, ontology contains terms and the relationships among these terms.



The Concept of Ontology

What Is Ontology — *continued: A Developer's Guide to the Semantic Web, p 151*

“An ontology formally defines a common set of terms that are used to describe and represent a domain . . .”

“An ontology defines the terms used to describe and represent an area of knowledge.”

-
- Terms are often called classes, or concepts, and these words are interchangeable.



The Concept of Ontology

What Is Ontology — *continued: A Developer's Guide to the Semantic Web, p 151*

“An ontology formally defines a common set of terms that are used to describe and represent a domain . . .”

“An ontology defines the terms used to describe and represent an area of knowledge.”

-
- The relationships between these classes can be expressed by using a hierarchical structure.
 - Super-classes represent higher-level concepts, and subclasses represent finer concepts.
 - The finer concepts have all the attributes and features that the higher concepts have.



The Concept of Ontology

What Is Ontology — *continued: A Developer's Guide to the Semantic Web, p 151*

“An ontology formally defines a common set of terms that are used to describe and represent a domain . . .”

“An ontology defines the terms used to describe and represent an area of knowledge.”

-
- Third, besides the above relationships among the classes, there is another level of relationship expressed by using a special group of terms: properties.



The Concept of Ontology

What Is Ontology — *continued: A Developer's Guide to the Semantic Web, p 151*

“An ontology formally defines a common set of terms that are used to describe and represent a domain . . .”

“An ontology defines the terms used to describe and represent an area of knowledge.”

-
- These property terms describe various features and attributes of the concepts.
 - They can also be used to associate different classes together.



The Concept of Ontology

What Is Ontology — *continued: A Developer's Guide to the Semantic Web, p 151*

“An ontology formally defines a common set of terms that are used to describe and represent a domain . . .”

“An ontology defines the terms used to describe and represent an area of knowledge.”

-
- Therefore, the relationships among classes are not only super-class or subclass relationships, but also relationships expressed in the term of properties.



The Concept of Ontology

What Is Ontology — *continued: A Developer's Guide to the Semantic Web, p 151*

- Thus ontology clearly defines the terms and the relationships among these terms.
- Ontology encodes the knowledge of the domain in such a way that the knowledge can be understood by a computer.
- This is the basic idea of ontology.

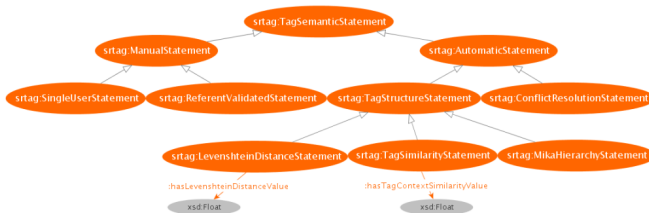
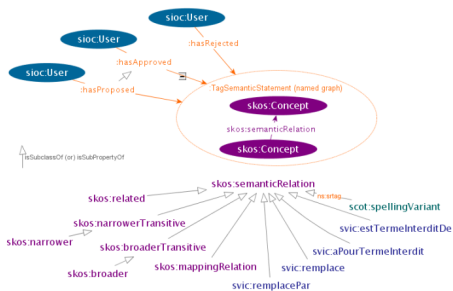


The Concept of Ontology

The Benefits of Ontology: *A Developer's Guide to the Semantic Web*, p 151

- We can summarize the benefits of ontology as follows
 - It provides a common and shared understanding/definition about certain key concepts in the domain.
 - It offers the terms one can use when creating RDF documents in the domain.
 - It provides a way to reuse domain knowledge.
 - It makes the domain assumptions explicit.
 - Together with ontology description languages, it provides a way to encode knowledge and semantics such that machines can understand.
 - It makes automatic large-scale machine processing possible.





End of Slides