RDF: building block for the Semantic Web

how do we code meaning/knowledge?

RDF: in plain English

- had experience shopping online?
- read product reviews before you buy!

You are a quality engineer who works for Nikon. Your assignment is to read reviews and summarize what people have said about Nikon SLR cameras, and report back to Nikon design department so they can make better designs.

- write an application that will read all these reviews, and generate a report automatically
- you can run this application as often as you want, just to gather the latest reviews
- a great idea with only one flaw: such an application is not easy to develop, since the reviews published online are intended for human eyes, not for machines to read

RDF: in plain English

- assume all the review publishers are willing to accept and follow some *standard* when they publish their reviews
- *what standard would you create so to make it easier to develop such an application?*

RDF: in plain English

- assume all the review publishers are willing to accept and follow some *standard* when they publish their reviews
- *what standard would you create so to make it easier to develop such an application?*

here is what we are looking for:

- something so flexible that it can express any fact easily
- yet so structured that machines can do useful things with these facts
- it should provide a mechanism to connect the distributed information (knowledge) over the Web
- in addition, it should come with very little constraints, so the participants do not need to agree upon too much

RDF: in plain English

something so flexible that it can express any fact easily

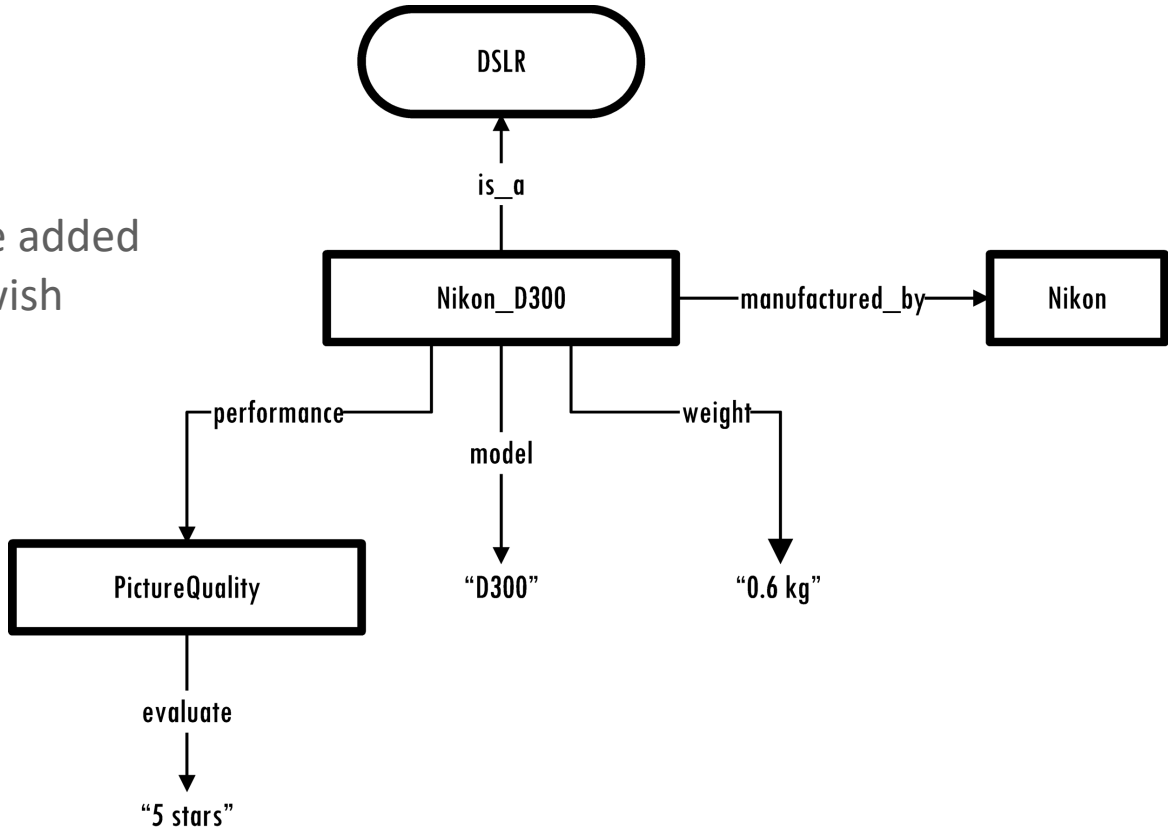
- each reviewer has different things to say about a given Nikon camera
- and whatever she/he wants to say, the standard must provide a way to allow it

RDF: in plain English

something so flexible that it can express any fact easily

- each reviewer has different things to say about a given Nikon camera
- and whatever she/he wants to say, the standard has to provide a way to allow it

- how about this?
- any new information can be added freely - just grow it as you wish



RDF: in plain English

- yet so structured that machines can do useful things with these facts:

table is considered as "structured" information

start node	edge label	end node
Nikon_D300	is_a	DSLR
Nikon_D300	manufactured_by	Nikon
Nikon_D300	performance	PictureQuality
Nikon_D300	model	“D300”
Nikon_D300	weight	“0.6 kg”
PictureQuality	evaluate	“5 stars”

- this tabular notation is exactly equivalent to the graph shown on previous page
- each row represents one arrow in the graph, including the start node, the edge with the arrow, and the end node
- no matter how the graph grows and no matter how complex it grows into, this table will always be able to represent it correctly

RDF: in plain English

- yet so structured that machines can do useful things with these facts:

start node	edge label	end node
Nikon_D300	is_a	DSLR
Nikon_D300	manufactured_by	Nikon
Nikon_D300	performance	PictureQuality
Nikon_D300	model	“D300”
Nikon_D300	weight	“0.6 kg”
PictureQuality	evaluate	“5 stars”

table is considered as "structured" information

machine can answer these questions easily:

- how many reviews have mentioned `weight`?
- what is the average star number?
- more...

RDF: in plain English

it should provide a mechanism to connect the distributed information (knowledge) over the Web

- reviews created by reviewers are distributed all over the Web
- different reviewers can use different names for the same product
- the standard must provide a way to eliminate this ambiguity so our application can process the reviews with certainty

RDF: in plain English

it should provide a mechanism to connect the distributed information (knowledge) over the Web

- reviews created by reviewers are distributed all over the Web
- different reviewers can use different names for the same product
- the standard must provide a way to eliminate this ambiguity so our application can process the reviews with certainty

and finally, it should come with very little constraints, so the participants do not need to agree upon too much.

RDF: in plain English

- W3C has long realized the need for such a standard, and the standard has been published and called **RDF**

Resource: a person, a book, a company, a tree, an idea…

Description: attributes, features, and relations of the resources

Framework: model, languages and syntax for these descriptions

RDF is a standard published by W3C, and it can be used to represent distributed information/knowledge in a way that computer applications can use and process in a scalable manner.

- RDF is the basic building block for supporting the vision of the Semantic Web
- RDF is for the Semantic Web what HTML has been for the Web

The Abstract Model of RDF

- there are at least six documents composing the RDF specification by W3C
- these documents all together describe different aspects of RDF
- one fundamental concept of RDF is its abstract model

abstract model: to decompose information/knowledge into small pieces, with some simple rules about the semantics (meaning) of each one of these pieces

3 key components:

- statement
- subject and object resources
- predicate

The Abstract Model of RDF

- a given small piece of knowledge is called a **statement**
- and it has the following basic rule (**Rule #1**):

  knowledge (or information) is expressed as a list of statements, each statement takes the form of Subject-Predicate-Object, and this order should never be changed.

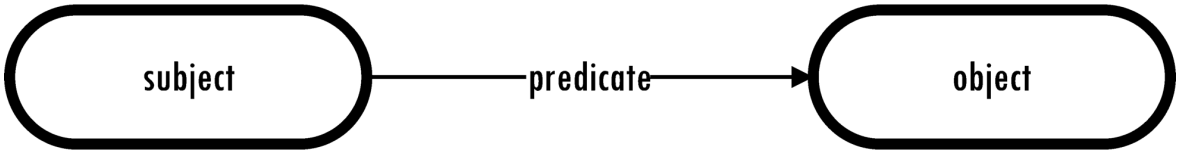
- an RDF statement always has the following structure:

```
subject predicate object
```

subject and object are names for two things in the world, with the predicate being the name of a relation that connects these two things

The Abstract Model of RDF

- graph structure of an RDF statement:



the subject is contained in the oval on the left, the object is the oval on the right, and the predicate is the label on the arrow, which points from the subject to the object.

- since an **RDF statement** always consists of three fixed components, it is also called an RDF **triple**
- a collection of statements or triples is called an **RDF graph**
- the “thing” that a given subject or object denotes, be it concrete or abstract, is called **resource**

The Abstract Model of RDF

how does the abstract model satisfy these conditions?

- something so flexible that it can express any fact easily
- yet so structured that machines can do useful things with these facts
- it should provide a mechanism to connect the distributed information (knowledge) over the Web
- in addition, it should come with very little constraints, so the participants do not need to agree upon too much

The Abstract Model of RDF

how does the abstract model satisfy these conditions?

- something so flexible that it can express any fact easily
- yet so structured that machines can do useful things with these facts
- it should provide a mechanism to connect the distributed information (knowledge) over the Web
- in addition, it should come with very little constraints, so the participants do not need to agree upon too much

RDF statement:
satisfies the first two conditions:

subject	predicate	object
Nikon_D300	is_a	DSLR
Nikon_D300	manufactured_by	Nikon
Nikon_D300	performance	PictureQuality
Nikon_D300	model	“D300”
Nikon_D300	weight	“0.6 kg”
PictureQuality	evaluate	“5 stars”

The Abstract Model of RDF

- it should provide a mechanism to connect the distributed information (knowledge) over the Web

  The solution proposed by RDF's abstract model is summarized in the following rule (**Rule #2**):

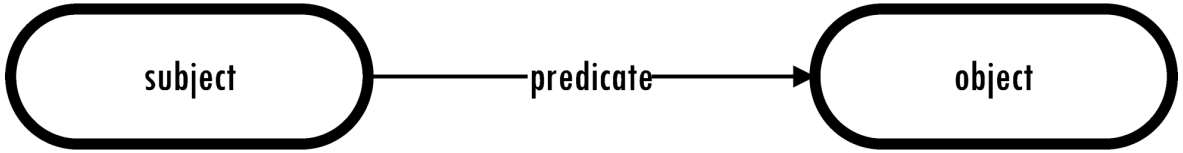
  > *The name of a resource must be global and should be identified by Uniform Resource Identifier (URI).*

why?

- globally unique, so we all know what you are talking about, there is no ambiguity of any kind;
- URI has a same syntax and format as website address, so looks familiar;
- de-referenceable URI (from the world of Linked Data, later topic) can naturally fit in;
- and more to come …

The Abstract Model of RDF

The name of a resource must be global and should be identified by Uniform Resource Identifier (URI).



- **subject**, **object**: resources (things in the world)
- **predciate**: relations the connects subject and object

so:

- we know both subject and object should be identified by URIs
- what about predicate? RDF abstract model requires the usage of URIs to identify predicates as well

The Abstract Model of RDF

So **Rule #2** should be changed to the following now:

The name of a resource must be global and should be identified by URI, and the name of predicate must also be global and should be identified by URI as well.

why?

- globally unique: *model* can be used to describe a camera, it can also be used to describe a car..., and now we know what you are talking about
- enable the predicates to be treated as resources, so we can say things about a given predicate (it is a sub-predicate of another predicate, it is the same predicate as another predicate...)
- using URIs to name subjects, predicates, and objects in RDF statements promotes the development and use of shared vocabularies on the Web (more on this later)

The Abstract Model of RDF

how does a URI look like?

- it is just a more general form of URL
- two different types: *slash URI* and *hash URI*

  slash URI: `http://www.example.com/camera/Nikon_D300`
  hash URI: `http://www.example.com/camera#Nikon_D300`

- we will understand the difference later
- for now, it is your choice which one to use

some examples:

we made up the first 2, the last 2 are existing ones

`http://www.example.com/camera#Nikon_D300`
`http://www.example.com/camera#DSLR`
`http://www.dbpedia.org/resource/Nikon`
`http://www.dbpedia.org/resource/Roger_Federer`

The Abstract Model of RDF

how do we produce URIs?

- make up our own, or,
- re-use existing ones

if a given resource has a URI that identifies it already, we should re-use this existing URI whenever we can

- for a given resource, how do we know if there exists some URI already? how do we find it?
- what if there are multiple URIs existing for this single resource?
- these will be answered later ...

The Abstract Model of RDF

- before we see more examples, it will be helpful for use to understand the concept of **QName**:

  a long URI is usually abbreviated by replacing it with its XML *qualified name* (QName), which contains a *prefix* that maps to a namespace URI, followed by a colon, and then a *local name* (borrowed from XML world)

Example:

prefix	namespace
myCamera	http://www.example.com/camera#
dbpedia	http://www.dbpedia.org/resource/

```
http://www.example.com/camera#Nikon_D300
http://www.dbpedia.org/resource/Nikon
```

can be written as:

```
myCamera:Nikon_D300
dbpedia:Nikon
```

The Abstract Model of RDF

- our review expressed in RDF statements, using URIs as resource and predicate names:

subject	predicate	object
myCamera:Nikon_D300	myCamera:is_a	myCamera:DSLR
myCamera:Nikon_D300	myCamera:manufactured_by	dbpedia:Nikon
myCamera:Nikon_D300	myCamera:performance	myCamera:PictureQuality
myCamera:Nikon_D300	myCamera:model	"D300"
myCamera:Nikon_D300	myCamera:weight	"0.6 kg"
myCamera:PictureQuality	myCamera:evaluate	"5 stars"

The Abstract Model of RDF

so how does machine understand and work with RDF statements?

- an application cannot associate any special meanings to the RDF statements
- however, with the fix structure of statement and some extra work, the logical pieces of meaning can be mechanically maneuvered by the given application
- it therefore can act as if it does understand these statements

The Abstract Model of RDF

- RDF data model provides a simple and elegant way to present facts – with well-defined structure that machine can understand
- RDF triples are created in a distributed fashion – you can say anything about anything
- RDF data model allows distributed information to be related in a meaningful way – use URI to represent resource/predicate
- if you are talking about Washington as a state (*not George Washington, or Washington DC, or …*), then use the URI that represents Washington as a state – semantic disambiguation