

Internettechnologie

academiejaar 2012–2013

Practicum 3: Semantisch Web

18 april 2013

Pieterjan De Potter

Multimedia Lab

vakgroep Elektronica en Informatiesystemen

faculteit Ingenieurswetenschappen

Universiteit Gent

Overzicht

- Introductie
- Resource Description Framework (RDF)
- RDF Schema (RDFS)
- Web Ontology Language (OWL)
- SPARQL Protocol And RDF Query Language (SPARQL)
- Opgave

De inleiding van semantisch-webtechnologieën is grotendeels gebaseerd op de tutorial van Ivan Herman (Semantic Web Activity lead):

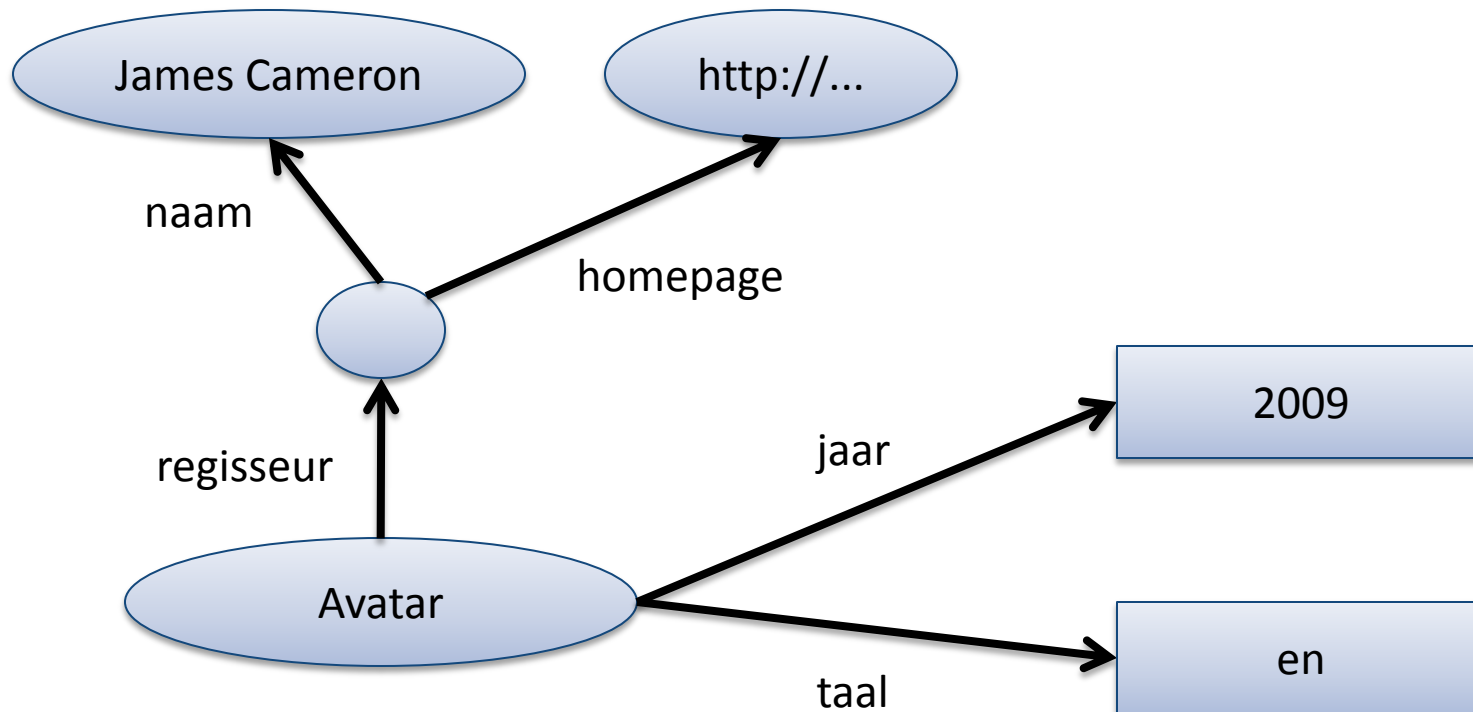
<http://www.w3.org/People/Ivan/CorePresentations/SWTutorial/Slides.pdf>

Herhaling semantisch web (1/2)

- Nood aan combinatie van verschillende data op het Web
 - bv. zoeken naar boeken op verschillende online bibliotheken
 - bv. het zoeken naar een geschikte vlucht over verschillende maatschappijen heen
- Deze data is niet machinaal toegankelijk
 - Onvolledige data
 - Verschillende semantiek
- Semantisch web
 - Stelt data op een machinaal begrijpbare manier voor
 - Laat machinaal redeneren op de data toe

Herhaling semantisch web (2/2)

- Informatie voorgesteld door middel van gerichte grafen
 - bv. beschrijving van een film



Gebruikte technologieën in dit practicum

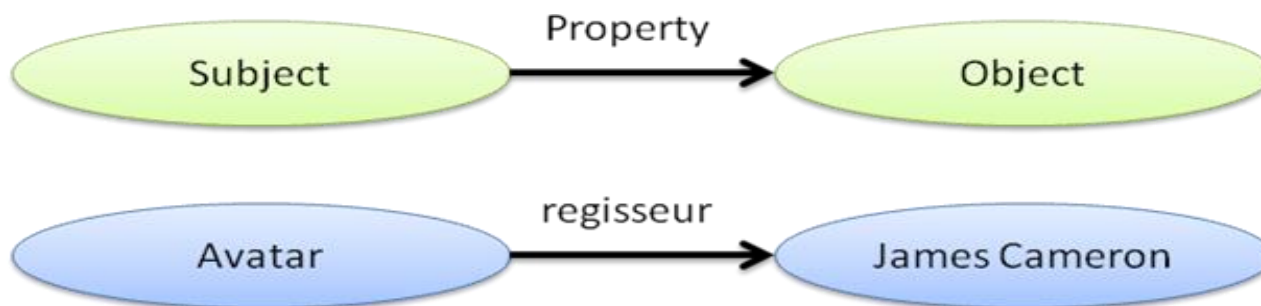
- Resource Description Framework (RDF)
 - <http://www.w3.org/TR/2004/REC-rdf-primer-20040210/>
- RDF Schema (RDFS)
 - <http://www.w3.org/TR/rdf-schema/>
- Web Ontology Language (OWL)
 - in dit practicum: OWL 1.0
 - <http://www.w3.org/TR/owl-features/>
- SPARQL Protocol And RDF Query Language (SPARQL)
 - <http://www.w3.org/TR/rdf-sparql-query/>

Overzicht

- Introductie
- Resource Description Framework (RDF)
- RDF Schema (RDFS)
- Web Ontology Language (OWL)
- SPARQL Protocol And RDF Query Language (SPARQL)
- Opgave

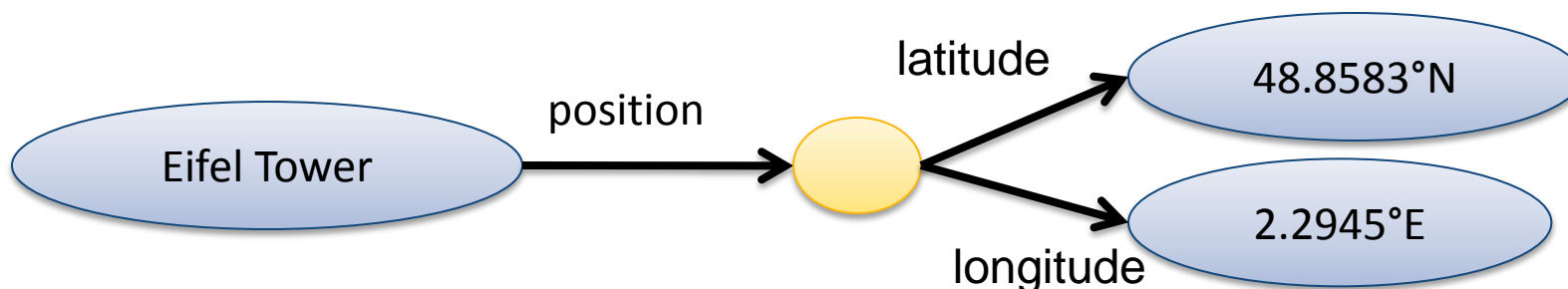
Resource Description Framework (RDF)

- Resources worden beschreven a.h.v. ‘triples’
- Gegeven een RDF-triple (s, p, o) geldt
 - s , p en o staan respectievelijk voor subject, property en object
 - s en p zijn URI's (i.e., resources op het Web); o is een URI of een literal
 - bv. ($\langle \text{http://.../Avatar} \rangle$, $\langle \text{http://.../jaar} \rangle$, 2009)
- Resources kunnen overeenkomen met om het even welke URI
 - moet bv. niet ‘dereferencable’ zijn
- RDF-triples vormen een gerichte, gelabelde graaf!



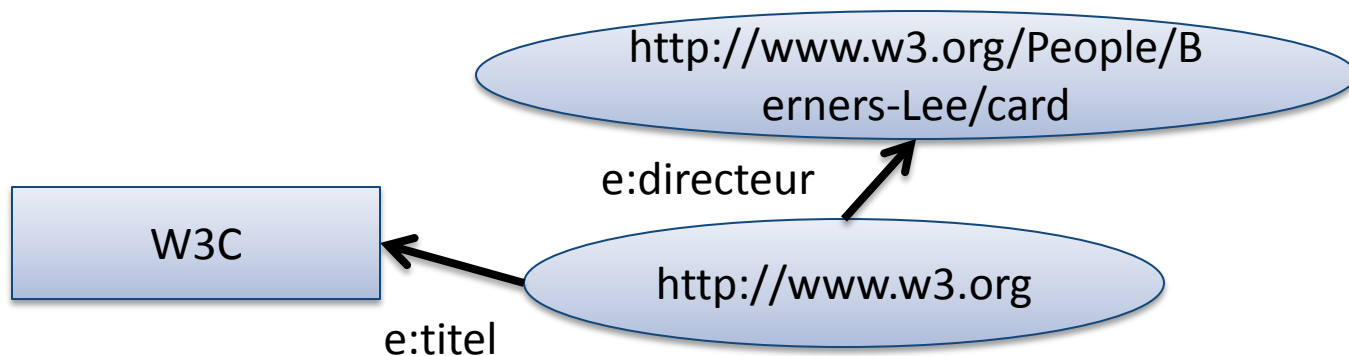
Interne knopen in RDF

- Beschouw het volgende statement
 - een “position” is “iets” dat een “latitude” en “longitude” coördinaat heeft.
- Knopen kunnen overeenkomen met een URI, een literal of een blanke knoop (Eng. *blank node*)
 - blanke knopen komen overeen met “iets”
- In de logica komen blanke knopen overeen met een ‘existentieel’ statement
 - er bestaat een resource zodat ...



RDF-serialisaties

- RDF/XML



```
<rdf:Description rdf:about="http://www.w3.org">
  <e:directeur rdf:resource="http://.../Berners-Lee/card"/>
  <e:titel>W3C</e:titel>
</rdf:Description>
```

- N3

```
<http://www.w3.org>
  e:directeur <http://.../Berners-Lee/card> ;
  e:titel "W3C" .
```

Notation 3 (N3)

- Voorgesteld door Tim Berners-Lee in 1998
- Leesbaar, eenvoudiger alternatief voor RDF/XML
 - N3 en RDF/XML zijn equivalent
 - representeren beide het RDF-model
 - online vertaalservices zijn beschikbaar
 - <http://www.mindswap.org/2002/rdfconvert/>
 - <http://www.rdfabout.com/demo/validator/>

N3: basisprincipes

- RDF-triples worden als volgt voorgesteld
 - subject property object .
 - URIs staan tussen ‘<’ en ‘>’
 - literals staan tussen dubbele quotes
 - datatypes kunnen meegegeven worden m.b.v. ‘^^’
 - N3-statements eindigen altijd met een punt
- Voorbeeld
 - `<http://.../Avatar> <http://.../jaar> “2009”^^xsd:int .`
- Commentaar
 - lijn die begint met #

N3: naamruimten

- URIs zijn typisch lang
 - lastig om dezelfde URI-prefix telkens te herhalen
- Naamruimten bieden een oplossing voor dit probleem
 - cfr. XML-naamruimten

<http://www.w3.org> <http://purl.org/dc/elements/1.1/title> “W3C” .



@prefix **dc**: <http://purl.org/dc/elements/1.1/> .

<http://www.w3.org> **dc**:title “W3C” .

N3: shortcuts (1/2)

- Gebruik van puntkomma
 - meerdere properties voor hetzelfde subject

```
@prefix : <http://example.org/> .  
@prefix foaf: <http://xmlns.com/foaf/0.1/> .  
  
:me foaf:name "John" .  
:me foaf:title "M.Sc." .
```



```
@prefix : <http://example.org/> .  
@prefix foaf: <http://xmlns.com/foaf/0.1/> .  
  
:me foaf:name "John" ;  
    foaf:title "M.Sc." .
```

N3: shortcuts (2/2)

- Gebruik van de komma
 - meerdere objects voor dezelfde subject en property

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

```
:me foaf:title "Dr." .  
:me foaf:title "M.Sc." .
```



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

```
:me foaf:title "Dr." , "M.Sc." .
```

N3: blanke knopen

- Blanke knopen beginnen met “_:”
- Vermijd naamgeving van blanke knopen door gebruik te maken van ‘[’ en ‘]’
 - properties van de blanke knopen staan tussen vierkante haken

@prefix : <http://example.org/> .

@prefix dbpedia:<http://dbpedia.org/resource/>.

dbpedia:Eiffel_Tower :position **_:bnode0** .

_:bnode0 :latitude “48.8583°N” .

_:bnode0 :longitude “2.2945°E” .

@prefix : <http://example.org/> .

@prefix dbpedia:<http://dbpedia.org/resource/> .

dbpedia:Eiffel_Tower :position [:latitude “48.8583°N”;
:longitude “2.2945°E”] .

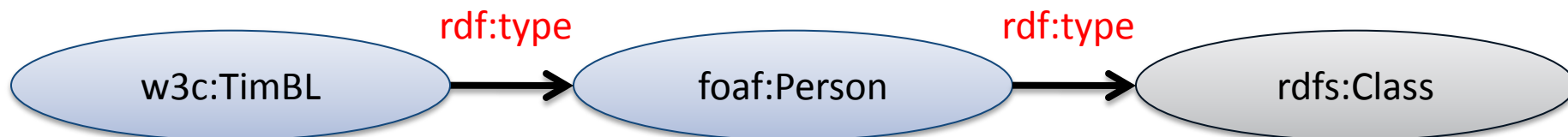


Overzicht

- Introductie
- Resource Description Framework (RDF)
- **RDF Schema (RDFS)**
- Web Ontology Language (OWL)
- SPARQL Protocol And RDF Query Language (SPARQL)
- Opgave

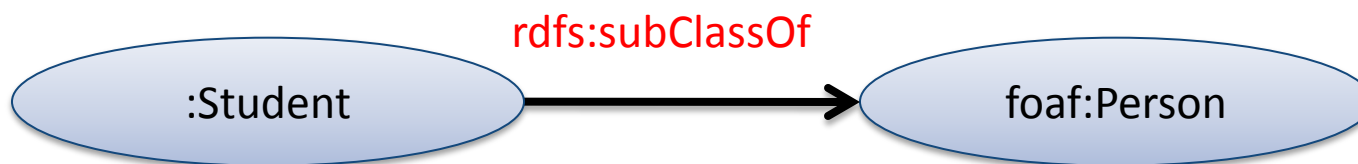
RDF Schema (RDFS)

- Er is nood aan een extra laag bovenop RDF
 - introductie van extra kennis
 - groeperen van resources in klassen
 - semantische betekenis van properties
- RDFS is eigenlijk niets anders dan een verzameling resources die een speciale betekenis gekregen hebben
- Definitie van een klasse in RDFS
 - **rdfs-prefix:** <<http://www.w3.org/2000/01/rdf-schema#>>
 - **rdf-prefix:** <<http://www.w3.org/1999/02/22-rdf-syntax-ns#>>



RDFS subclassing

- Een klassenhiërarchie kan bekomen worden door subklassen te definiëren
 - m.b.v. de property *rdfs:subClassOf*



RDFS classes: enkele opmerkingen

- Een resource kan tot meerdere klassen tegelijk behoren
 - `rdf:type` is gewoon een property met een speciale betekenis
- Informatie over klassen kan heel interessant zijn voor applicaties
 - opdelen van resources in categorieën
 - `rdf:type` is waarschijnlijk de meest gebruikte RDF property
- Volgende voorstellingen in N3 zijn equivalent

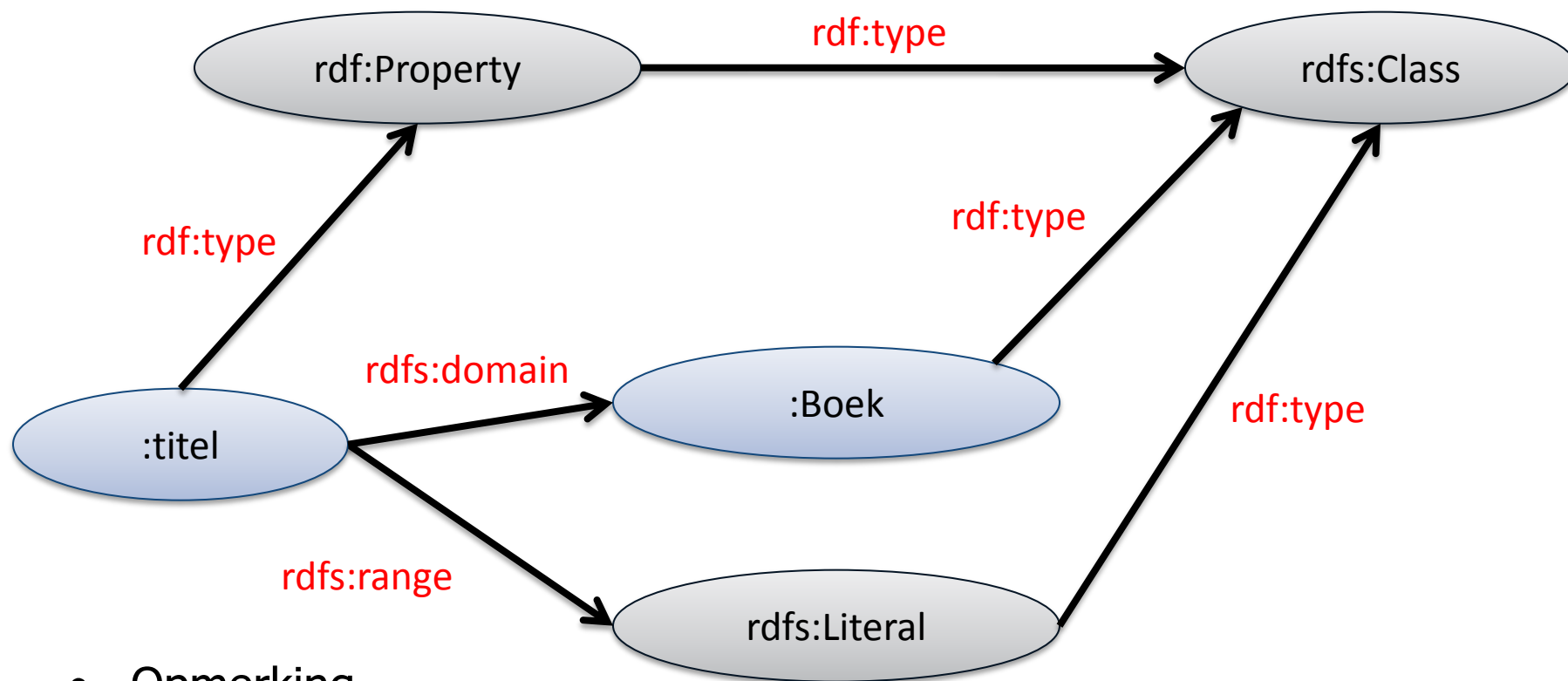
```
w3c:TimBL rdf:type foaf:Person .  
foaf:Person rdf:type rdfs:Class .
```

```
w3c:TimBL a foaf:Person .  
foaf:Person a rdfs:Class .
```

RDF properties

- Properties zijn resources en zijn tevens lid van een klasse (i.e., `rdf:Property`)
- Subproperties kunnen gedefinieerd worden
 - `rdfs:subPropertyOf`
- Range en domain van properties kunnen gedefinieerd worden
 - range: welke type resource komt voor in het object
 - domain: welk type resource komt voor in het subject

RDF properties: voorbeeld



- Opmerking
 - dit betekent dat `:titel` *niet* gebruikt kan worden als property voor een klasse `:Film`

RDF properties: voorbeeld in N3

```
@prefix : <http://example.org/> .  
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .  
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .  
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .  
  
:Boek rdf:type rdfs:Class .  
  
:titel rdf:type rdf:Property ;  
    rdfs:domain :Boek ;  
    rdfs:range rdfs:Literal .  
  
:VoorbeeldBoek rdf:type :Boek ;  
    :titel "Voorbeeld titel"^^xsd:string .
```

Beschrijvende RDFS-properties

- `rdfs:label`
- `rdfs:comment`

@prefix : <http://example.org/> .

@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .

@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .

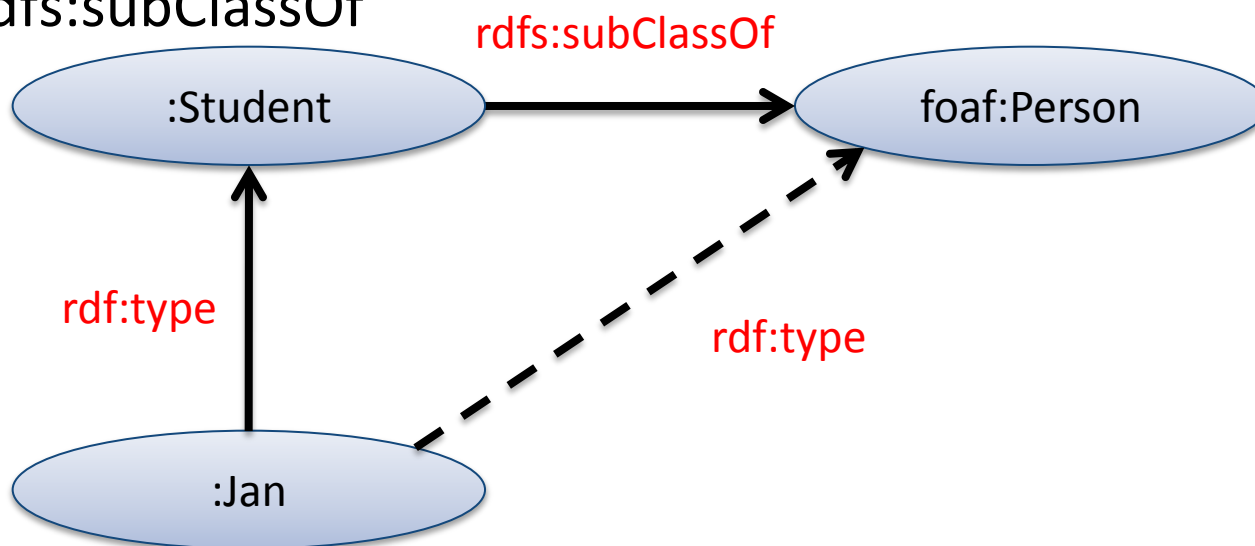
:Boek rdf:type rdfs:Class ;

 rdfs:label "Boek" ;

 rdfs:comment "Voorstelling van een boek" .

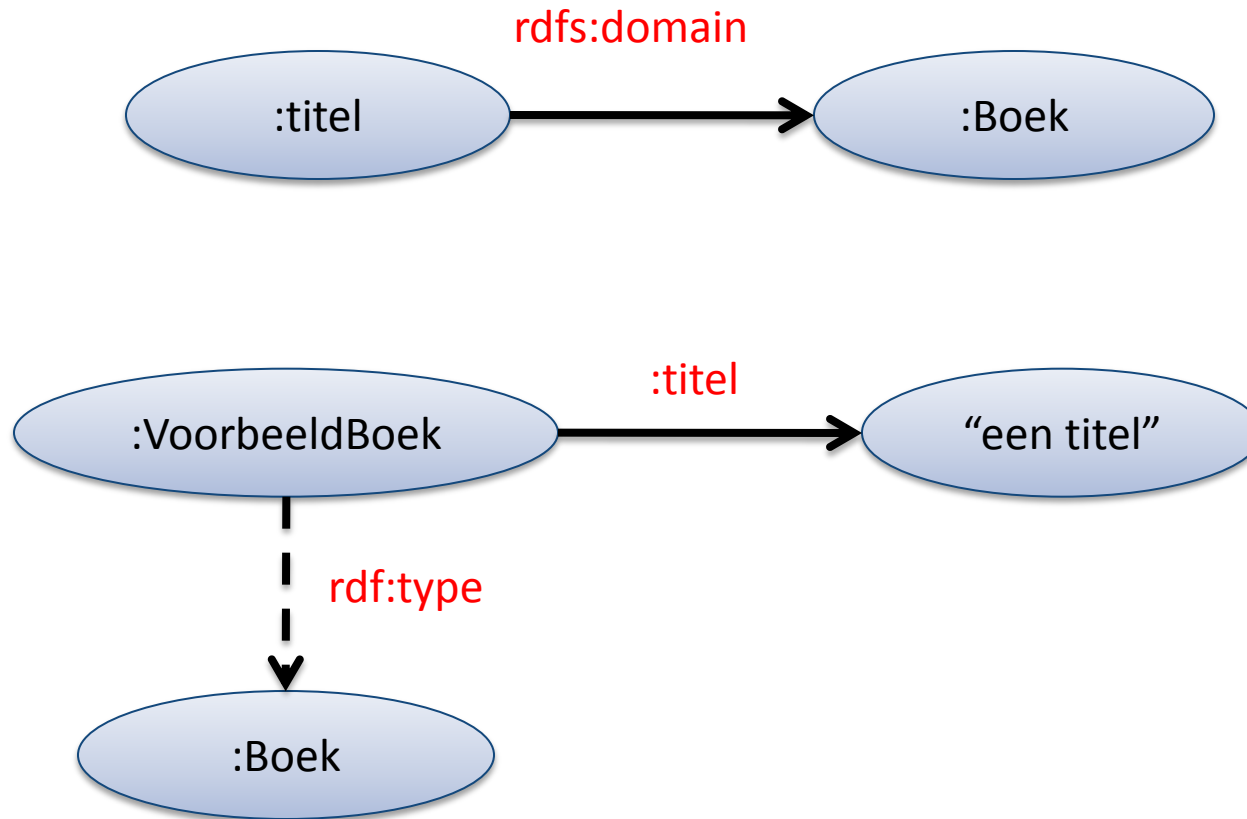
RDFS inferencing (1/2)

- Inferencing: nieuwe kennis wordt afgeleid uit bestaande kennis
- RDFS-concepten laten ons toe te redeneren
 - specifieke RDFS-reasoner nodig!
- Voorbeeld 1
 - `rdfs:subClassOf`



RDFS inferencing (2/2)

- Voorbeeld 2
 - rdfs:domain



Overzicht

- Introductie
- Resource Description Framework (RDF)
- RDF Schema (RDFS)
- **Web Ontology Language (OWL)**
- SPARQL Protocol And RDF Query Language (SPARQL)
- Opgave

Ontologieën

- RDFS is handig, maar heeft ook een aantal belangrijke beperkingen
 - beperkte karakterisatie van properties
 - verschillende URI's voor één object
 - equivalentie tussen klassen/properties
 - rijkere constructie van klassen
 - etc.
- Gedetailleerde domeinbeschrijving: **ontologie**
 - definieert concepten en relaties die gebruikt worden om kennis van een bepaald domein te beschrijven

Web Ontology Language (OWL)

- Taal om ontologieën te beschrijven
 - RDFS kan beschouwd worden als een heel eenvoudige ontologietaal
- OWL is een extra laag, bovenop RDFS
 - OWL definieert, net zoals RDFS, een aantal termen met speciale betekenis voor reasoners
- Twee versies van OWL
 - OWL 1 (2004)
 - deze versie zal gebruikt worden in dit practicum
 - OWL 2 (2009)

OWL complexiteit

- OWL is heel complex/uitgebreid
 - brede waaier aan extra termen
 - enkel een beperkt overzicht wordt gegeven in dit practicum
 - enkel de concepten besproken in deze presentatie zullen nodig zijn om het practicum op te lossen
 - complete informatie vind je terug in de specificatie en theorie lessen

Klassen in OWL

- OWL definieert nieuwe concept voor de definitie van klassen: *owl:Class*

```
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
```

```
@prefix owl: <http://www.w3.org/2002/07/owl#> .
```

```
owl:Class rdfs:subclassOf rdfs:Class .
```

- Constructie van klassen op basis van bestaande klassen
 - opsomming van mogelijke instanties
 - doorsnede van klassen
 - unie van klassen
 - complement van klassen
 - etc.

Klassen in OWL: voorbeeld

- Opsomming van mogelijke klassen: ***owl:oneOf***

@prefix : <http://example.org/> .

@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .

@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .

@prefix owl: <http://www.w3.org/2002/07/owl#> .

:Resultaat rdf:type owl:Class ;
 owl:oneOf (:Goed :Fout) .

:Goed rdf:type :Resultaat .

:Fout rdf:type :Resultaat .

Klassen in OWL: voorbeeld

- Unie van twee klassen: ***owl:unionOf***

@prefix : <http://example.org/> .

@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .

@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .

@prefix owl: <http://www.w3.org/2002/07/owl#> .

:Persoon rdf:type owl:Class ;
 owl:unionOf (:Man :Vrouw) .

:Man rdf:type owl:Class .

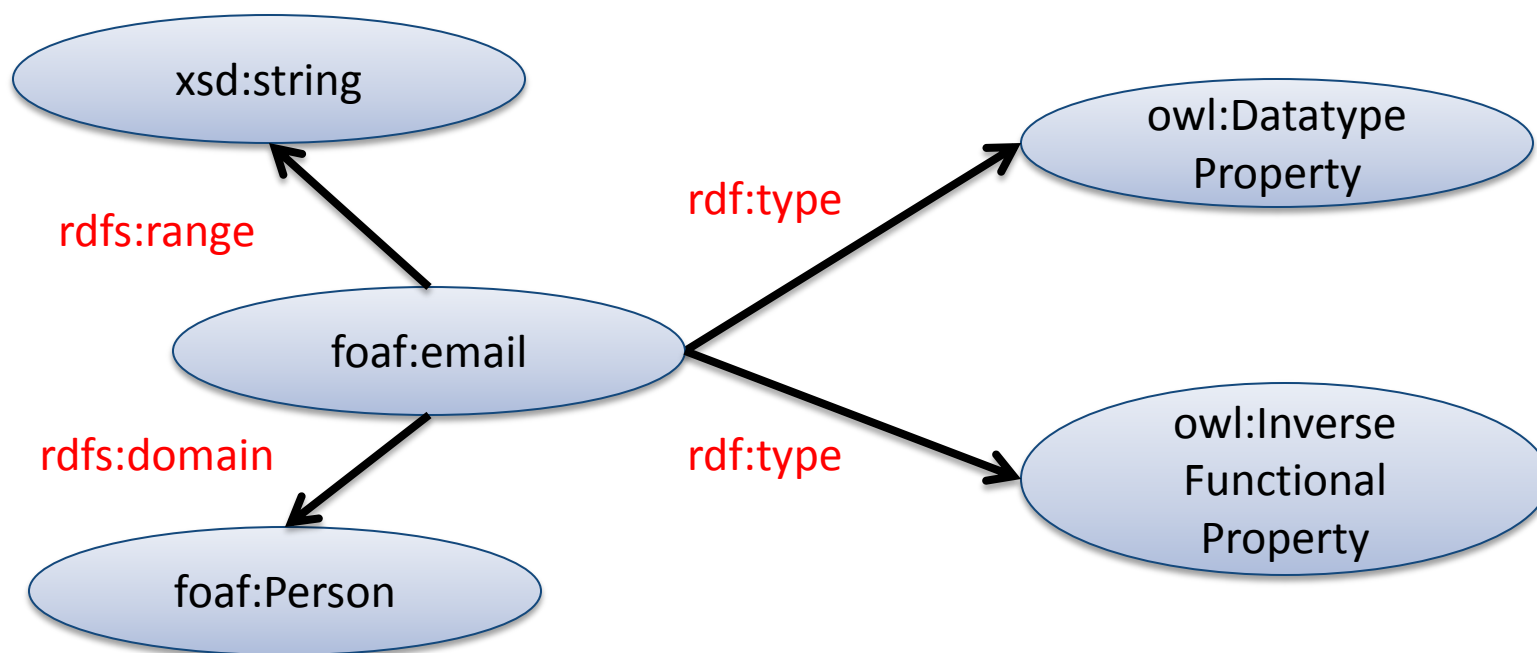
:Vrouw rdf:type owl:Class .

Properties in OWL

- Het gedrag van properties kan gekarakteriseerd worden
 - symmetrisch, transitief, functioneel, invers functioneel
 - onderscheid tussen *datatype property* en *object property*
 - datatype property: range correspondeert met literals
 - object property: range correspondeert met een OWL-klasse

OWL properties: voorbeeld

- Invers functioneel uitdrukken in OWL
 - i.e. twee verschillende subjects kunnen geen identieke objects hebben



OWL-properties voor equivalenties

- Voor klassen
 - owl:equivalentClass: twee klassen hebben dezelfde individuals (i.e., instanties)
 - owl:disjointWith: twee klassen hebben geen enkele individual gemeen
- Voor properties
 - owl:equivalentProperty
 - owl:propertyDisjointWith
- Voor individuals
 - owl:sameAs: twee URI's refereren naar hetzelfde concept
 - owl:differentFrom

OWL inferencing: voorbeeld

```
:email rdf:type owl:InverseFunctionalProperty .  
<A> :email "mailto:a@b.c" .  
<B> :email "mailto:a@b.c" .
```



OWL inferencing

```
<A> owl:sameAs <B> .
```

- Nieuwe kennis kan worden afgeleid
 - net zoals RDFS, maar veel geavanceerder

Voorbeeld OWL-ontologie

```
@prefix : <http://berio.elis.ugent.be/ontology/itechX/> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix foaf: <http://xmlns.com/foaf/0.1/> .

:      rdf:type owl:Ontology ;
      rdfs:comment "Ontologie voor de beschrijving van songs." ;
      owl:imports <http://downloads.dbpedia.org/3.2/en/dbpedia-ontology.owl> ;
      owl:imports <http://xmlns.com/foaf/spec/index.rdf> .

:Song  rdf:type owl:Class ;
      rdfs:comment "Stelt een song voor." .

:creator  rdf:type owl:ObjectProperty ;
          rdfs:comment "De maker van deze song (evt. URI).";
          rdfs:domain :Song ;
          rdfs:range foaf:Person .
```

Overzicht

- Introductie
- Resource Description Framework (RDF)
- RDF Schema (RDFS)
- Web Ontology Language (OWL)
- SPARQL Protocol And RDF Query Language (SPARQL)
- Opgave

SPARQL

- SPARQL Protocol And RDF Query Language
- Manier om toegang te krijgen tot RDF-data
 - bv. geef alle instanties van het type foaf:Person
- SPARQL-query
 - specificatie van de query syntax
- SPARQL-protocol
 - specificatie van de boodschappen tussen een client en een *SPARQL-endpoint*

SPARQL-query

- Gedeeltelijk gebaseerd op SQL
- PREFIX
 - declaratie van naamruimten
- SELECT
 - identificeer de waarden die teruggegeven moeten worden
- WHERE
 - het triplepatroon waarvoor een match gevonden moet worden

SPARQL-query: voorbeeld

- Selecteer de namen van alle instanties van het type foaf:Person

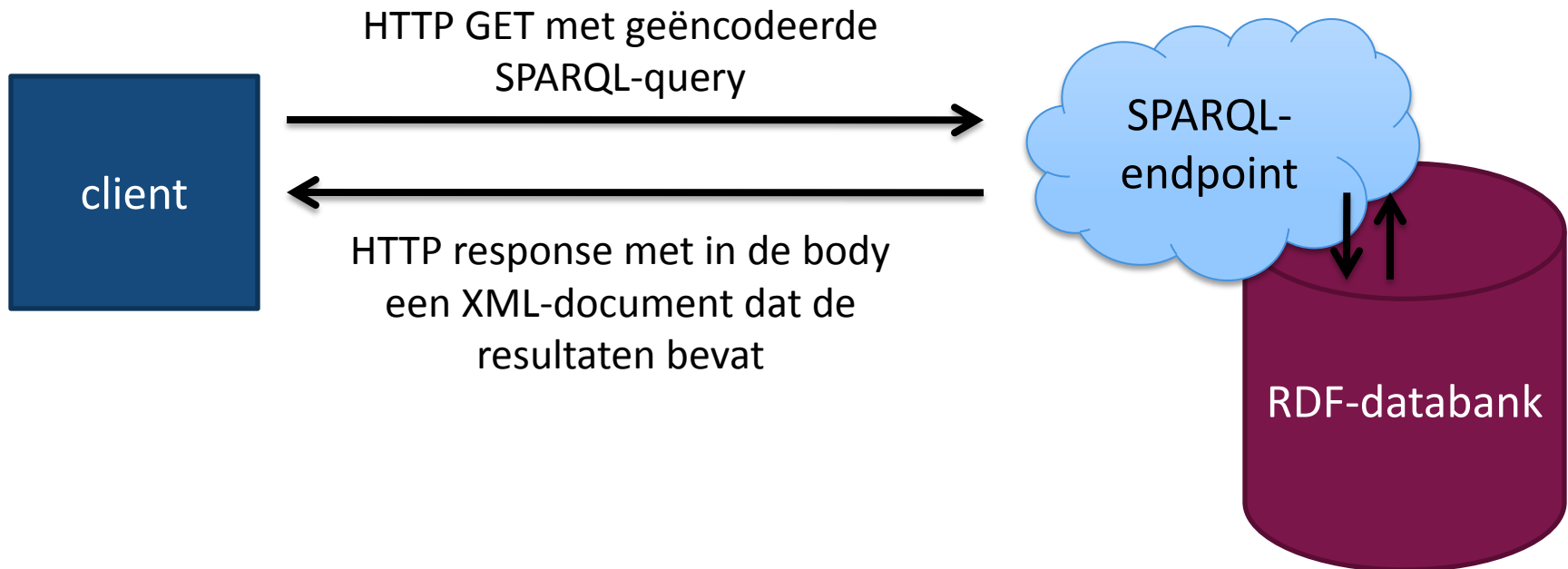
```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
SELECT ?name ?person
WHERE {
    ?person rdf:type foaf:Person .
    ?person foaf:name ?name .
}
```

- ?x : variabele met naam x

SPARQL-protocol

- SPARQL wordt meestal gebruikt over het netwerk
 - communicatieprotocol nodig
 - SPARQL over HTTP
 - SPARQL-resultaten in XML-formaat
- SPARQL-endpoint
 - een webservice die compatibel is met de SPARQL-specificatie
 - voorbeelden
 - <http://dbpedia.org/sparql/>
 - <http://data.linkedmdb.org/sparql/>

SPARQL-protocol over HTTP



SPARQL-protocol: HTTP-communicatie

```
GET /sparql?query=<SPARQLQuery> HTTP/1.1
```

```
User-Agent: my-sparql-client/0.0
```

```
Host: example.com
```

```
Accept: application/sparql-results+xml
```

```
HTTP/1.1 200 OK
```

```
Server: my-sparql-server/0.0
```

```
Content-Type: application/sparql-results+xml
```

```
{binary content}
```

- <SPARQLQuery> : URL-encoded SPARQL-query

SPARQL-protocol: resultaat in XML

```
<sparql xmlns="http://www.w3.org/2005/sparql-results#">
  <head>
    <variable name="name"/>
    <variable name="person"/>
    ...
  </head>
  <results>
    <result ordered="false" distinct="false">
      <binding name="name"><literal>Jan</literal></binding>
      <binding name="person"><uri>http://...</uri></binding>
      ...
    </result>
    <result> ... </result>
  </results>
</sparql>
```

Overzicht

- Introductie
- Resource Description Framework (RDF)
- RDF Schema (RDFS)
- Web Ontology Language (OWL)
- SPARQL Protocol And RDF Query Language (SPARQL)
- Opgave

Opgave 1

- Maak een OWL-ontologie voor de beschrijving van digitale foto's
- Creëer minstens twee voorbeeldinstanties
- De volgende zaken moeten beschreven worden
 - maker van de foto
 - wat er te zien is op de foto
 - thema
 - plaats waar de foto gemaakt is
 - datum waarop de foto gemaakt is
 - korte beschrijving van de foto
 - titel

Opgave 1: opmerkingen

- Niet alles wat in de ontologie beschreven staat, moet ook daadwerkelijk in de instanties voorkomen
 - bv. de maker van de foto kan niet gekend zijn
- Het thema van de foto kan *enkel* volgende waarden aannemen
 - dbpedia:People, dbpedia:Animals, dbpedia:Architecture, dbpedia:Nature, dbpedia:Politics, dbpedia:Humor, dbpedia:Culture, dbpedia:News
 - dbpedia: = <http://dbpedia.org/resource/Category:>
 - bv. <http://dbpedia.org/resource/Category:People>

Opgave 1: opmerkingen

- Probeer zoveel mogelijk gebruik te maken van bestaande RDF-data (bv. op DBPedia)
 - bv. gebruik <http://dbpedia.org/resource/Brussels> indien je wil aangeven dat de foto in Brussel genomen is
- Opgelegde naamruimte
 - foto-ontologie:
<http://vstudent.elis.ugent.be/ontology/itechX/>
 - met X gelijk aan het groepsnummer

Opgave 2

- Creëer links tussen de properties van jullie foto-ontologie en de Dublin Core properties
- Dublin Core (DC) is een verzameling van een 15-tal algemene metadata-elementen voor de annotatie van Web resources
 - meer info over de elementen in HTML-vorm
 - <http://dublincore.org/documents/dces/>
 - in RDF-vorm
 - <http://dublincore.org/2008/01/14/dcelements.rdf>
- Zorg ervoor dat de foto-ontologie gelinkt is met de klasse <http://xmlns.com/foaf/0.1/Image> (foaf:Image)
- Hou rekening met het feit dat we enkel beschikken over RDFS-inferencing
 - OWL wordt niet ondersteund door onze inferencing engine

Opgave 3

- Ontwikkel in ASP.NET een applicatie voor het opzoeken van foto's op basis van DC-properties
 - maak intern gebruik van SPARQL voor het uitvoeren van de zoekopdrachten
 - scherm SPARQL af voor de eindgebruiker
 - bv. geef alle foto's van het genre politiek met als locatie Brussel
- ASP.NET-applicatie
 - presentatie van een foto en bijhorende metadata
 - query-mogelijkheden
 - gebaseerd op dropdown boxen of invulvelden, afhankelijk van het soort property
 - originaliteit en layout worden beloond

Opgave 3: SPARQL-endpoint

- Op Minerva>Files zal een SPARQL-endpoint ter beschikking gesteld worden
 - RDF databank geschreven in java
 - Voert alle queries “in memory” uit
 - Parst alle n3 files in de zelfde directory als de executable
 - Opdrachtregel:
 - `java -jar itech-jar-with-dependencies.jar poortnr`
- Voer deze endpoint op de zelfde machine uit als de ASP webserver. Verbinden kan dan met “localhost:poortnr”

Opmerkingen bij uitvoeren op Athena

- Athena is een cloud systeem dus applicaties draaien niet noodzakelijk op de zelfde node
- Servers op verschillende nodes hebben geen toegang tot elkaar
- “kind processen” draaien wel op de zelfde node
- Start de server op via Visual Studio “tools” menu niet via de Athena opdrachtprompt (zie opgave)
- Niet alle poorten zijn beschikbaar op een node
 - Gebruik “groepsnr” + 11000

Oplossingen worden op Athena getest tijdens het verbeteren!

Practicum 3

- Opgave
 - Staat op Minerva
 - zie PDF met verdere uitleg
- Via Minerva dropbox bestanden uploaden in .zip file:
 - **semanticweb_groep[x].zip**
 - correcte bestandsnamen (!)
 - kopieer het volledige solution/project!
- Oplossingen maken
 - PC-klas 'Konrad Zuse'