



RDF Σχήμα (RDFS)

Σημασιολογικός Ιστός και Ευφυείς Εφαρμογές, 7^ο Εξάμηνο
Γεώργιος Μεδίτσκος, Επίκουρος Καθηγητής, Τμήμα Πληροφορικής ΑΠΘ

Σημασιολογικός Ιστός: Δεδομένα, Λογική και Ερωτήματα

Αποτελέσματα:

$\{(x \mapsto \text{Ireland}, y \mapsto \text{Europe}),$
 $(x \mapsto \text{Dublin}, y \mapsto \text{Ireland}),$
 $(x \mapsto \text{Dublin}, y \mapsto \text{Europe})\}$

Ερωτήματα:

$“(x, \text{partOf}, y)”$

Λογική:

$“(b, \text{capital}, a) \rightarrow (a, \text{partOf}, b)”$
 $“(a, \text{partOf}, b), (b, \text{partOf}, c) \rightarrow (a, \text{partOf}, c)”$

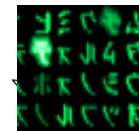
Δεδομένα:

Ireland



$(\text{Ireland}, \text{partOf}, \text{Europe})$
 $(\text{Ireland}, \text{isA}, \text{Country})$
 $(\text{Ireland}, \text{capital}, \text{Dublin})$

Dublin



$(\text{Ireland}, \text{capital}, \text{Dublin})$
 $(\text{Dublin}, \text{population}, 1000000)$

Αρχιτεκτονική ΣΙ

■ Resource Description Framework (RDF)

- Είναι κάτι ανάλογο της HTML στον ΣΙ
- Αποτελεί έναν απλό τρόπο για την περιγραφή πόρων
- Επιτρέπει το σταδιακό χτίσιμο της γνώσης, καθώς και το διαμοιρασμό και την επαναχρησιμοποίησή της.
- Βασίζεται στην έννοια της τριπλέτας (subject, predicate, object)

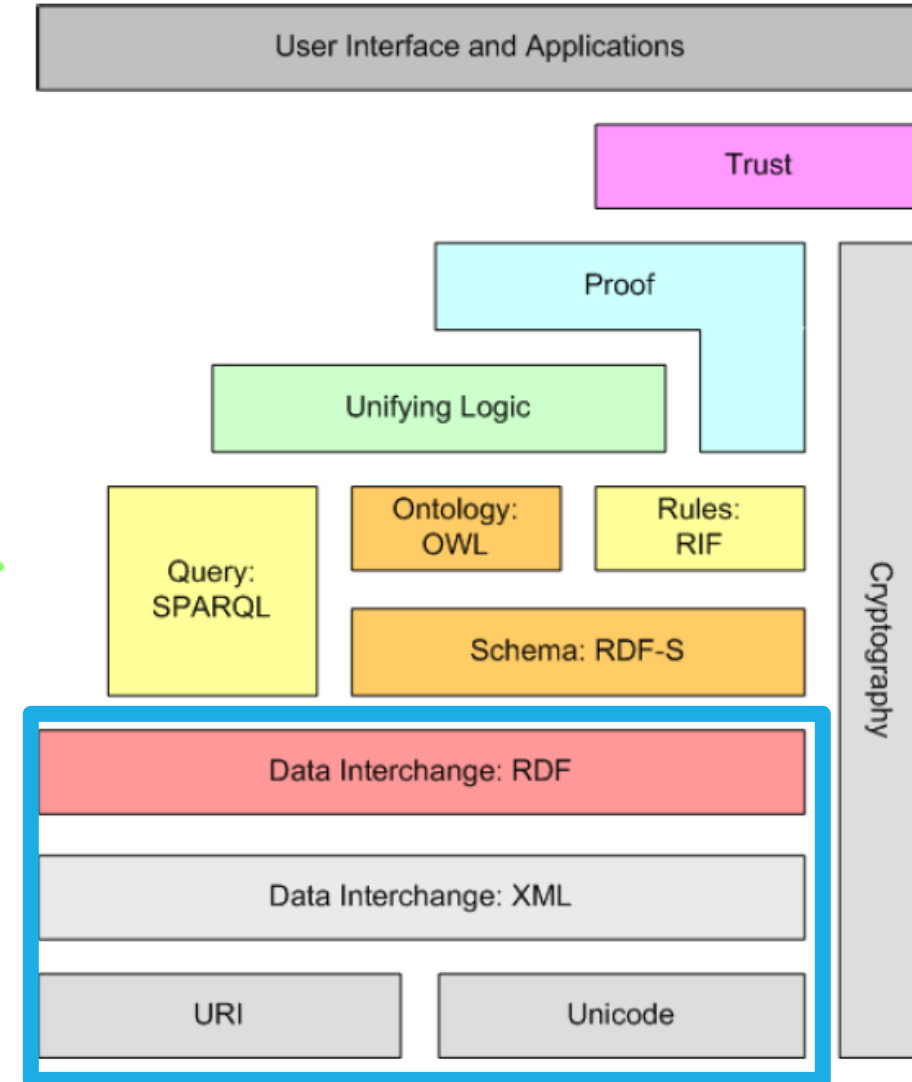
Turtle (syntax) format

```
@prefix eric: <http://www.w3.org/People/EM/contact#> .
@prefix contact: <http://www.w3.org/2000/10/swap/pim/contact#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .

eric:me contact:fullName "Eric Miller" .
eric:me contact:mailbox <mailto:e.miller123(at)example> .
eric:me contact:personalTitle "Dr." .
eric:me rdf:type contact:Person .
```

RDF/XML format

```
<?xml version="1.0" encoding="utf-8"?>
<rdf:RDF xmlns:contact="http://www.w3.org/2000/10/swap/pim/contact#" xmlns:eric="http://www.w3.org/People/EM/contact#"
  <rdf:Description rdf:about="http://www.w3.org/People/EM/contact#me">
    <contact:fullName>Eric Miller</contact:fullName>
  </rdf:Description>
  <rdf:Description rdf:about="http://www.w3.org/People/EM/contact#me">
    <contact:mailbox rdf:resource="mailto:e.miller123(at)example"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://www.w3.org/People/EM/contact#me">
    <contact:personalTitle>Dr.</contact:personalTitle>
  </rdf:Description>
  <rdf:Description rdf:about="http://www.w3.org/People/EM/contact#me">
    <rdf:type rdf:resource="http://www.w3.org/2000/10/swap/pim/contact#Person"/>
  </rdf:Description>
</rdf:RDF>
```

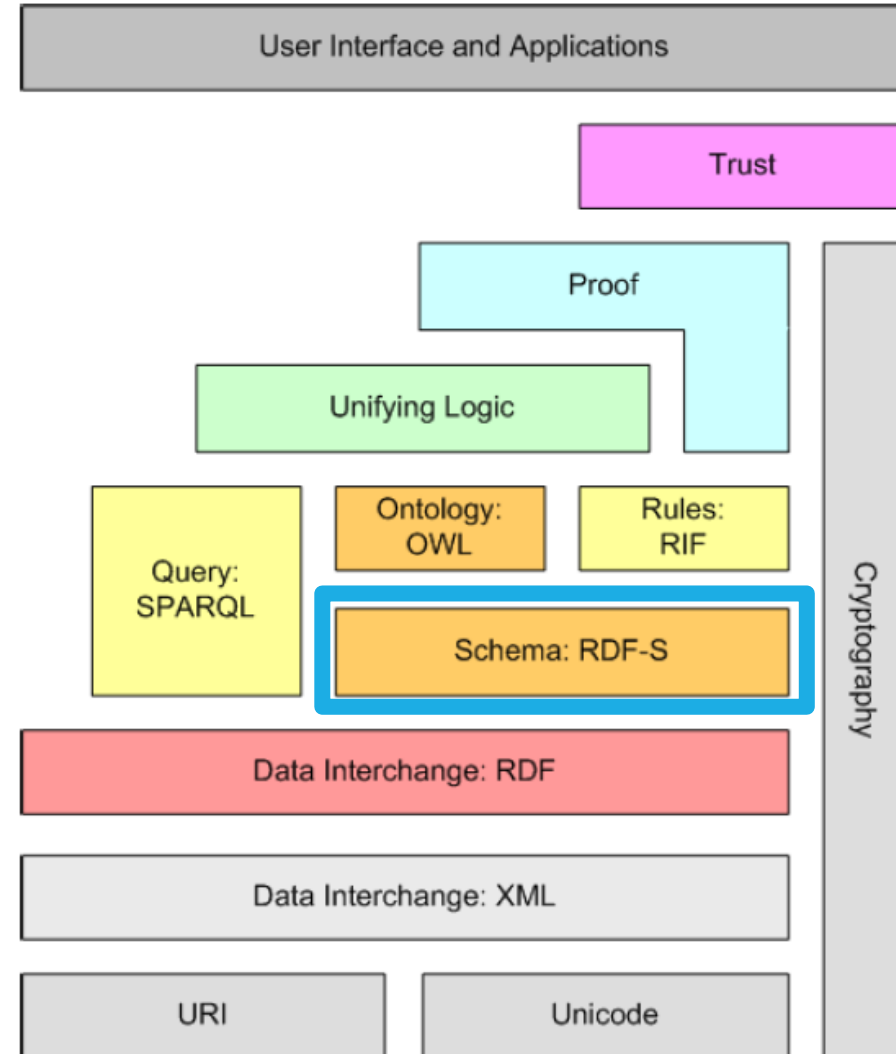


Αρχιτεκτονική ΣΙ

■ RDF Schema

- Είναι μια στοιχειώδης γλώσσα οντολογιών
- Παρέχει συγκεκριμένα θεμελιώδη στοιχεία μοντελοποίησης
 - κλάσεις, σχέσεις υποκλάσης, οι ιδιότητες, οι σχέσεις υπο-ιδιότητας, και οι περιορισμοί στο πεδίο ορισμού και στο σύνολο τιμών
- Στην ουσία περιγράφει το λεξιλόγιο (schema) και τις συσχετίσεις ανάμεσα στις διάφορες έννοιες

```
:AllStarPlayer rdf:type rdfs:Class.  
:MajorLeaguePlayer rdf:type rdfs:Class.  
:Surgeon rdf:type rdfs:Class.  
:Staff rdf:type rdfs:Class.  
:Physician rdf:type rdfs:Class.
```



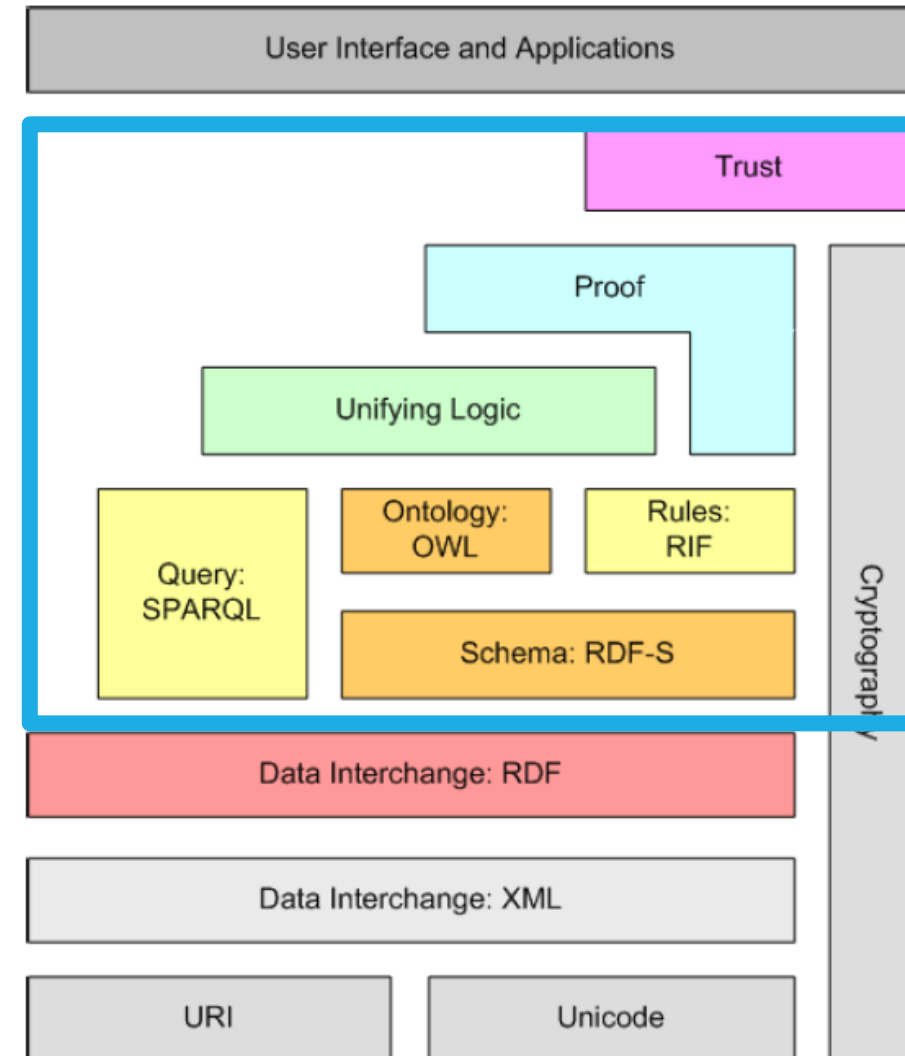
Ανασκόπηση

- Ο ΣΙ προσφέρει ένα μοντέλο έκφρασης δεδομένων (RDF)
 - Ρητή αναπαράσταση γνώσης και συσχετίσεων
- Δημιουργείς (μετα) δεδομένα
- Διασυνδέεις δεδομένα
- Γενικά τα δεδομένα περιγράφουν τον τρόπο με τον οποίο κάποιος μπορεί να τα χρησιμοποιήσει
- Ποιο είναι το επόμενο βήμα;

Συμπερασμός (inferencing / reasoning)

Συμπερασμός

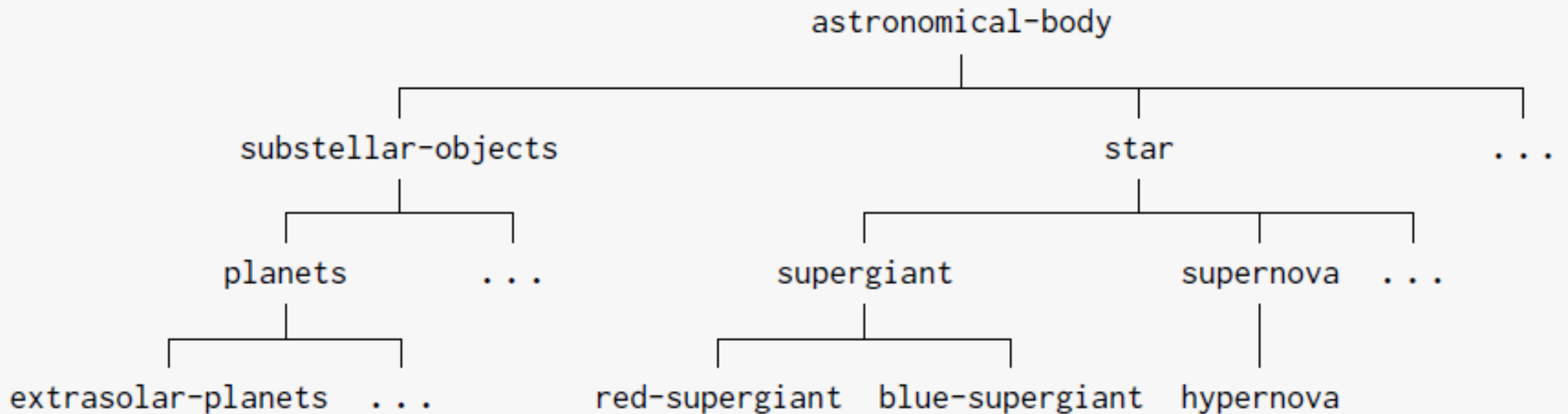
- Δεν είναι ρεαλιστικό να υποθέσεις ότι όλη η γνώση μπορεί να μοντελοποιηθεί από την αρχή από τον μηχανικό γνώσης
 - Πολυπλοκότητα πεδίου
 - Χρόνος – κόπος
 - Μπορεί όλες οι σχέσεις να μην είναι γνωστές από πριν
- Στο ΣΙ, βάσει κάποιων δηλώσεων (και της σημασιολογίας) μπορούμε να προσδιορίζουμε άλλες δηλώσεις





Wikidata

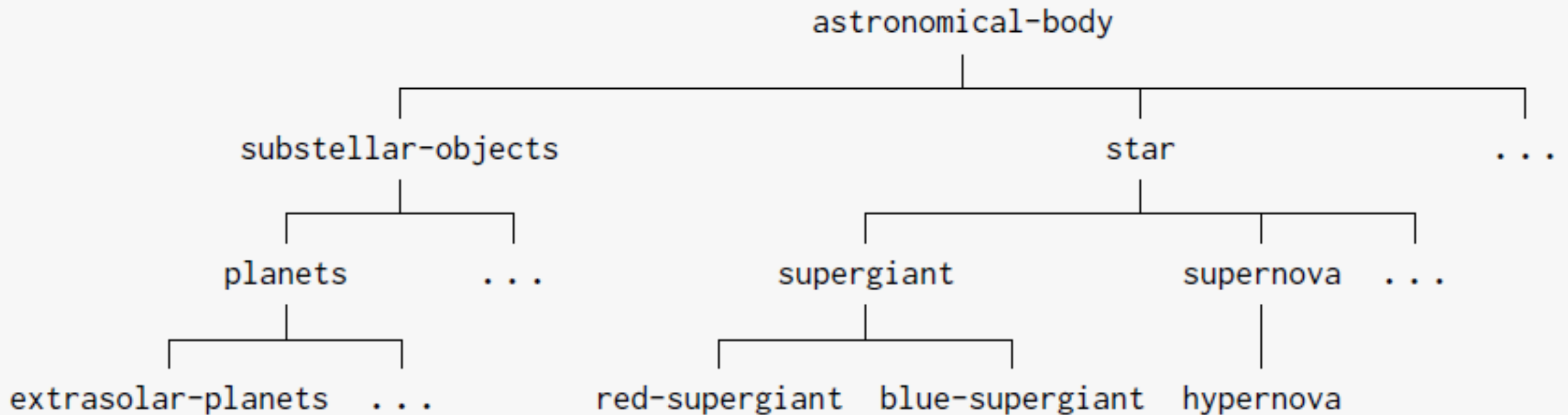
- “Wikidata is a free and open knowledge base that can be read and edited by both humans and machines”





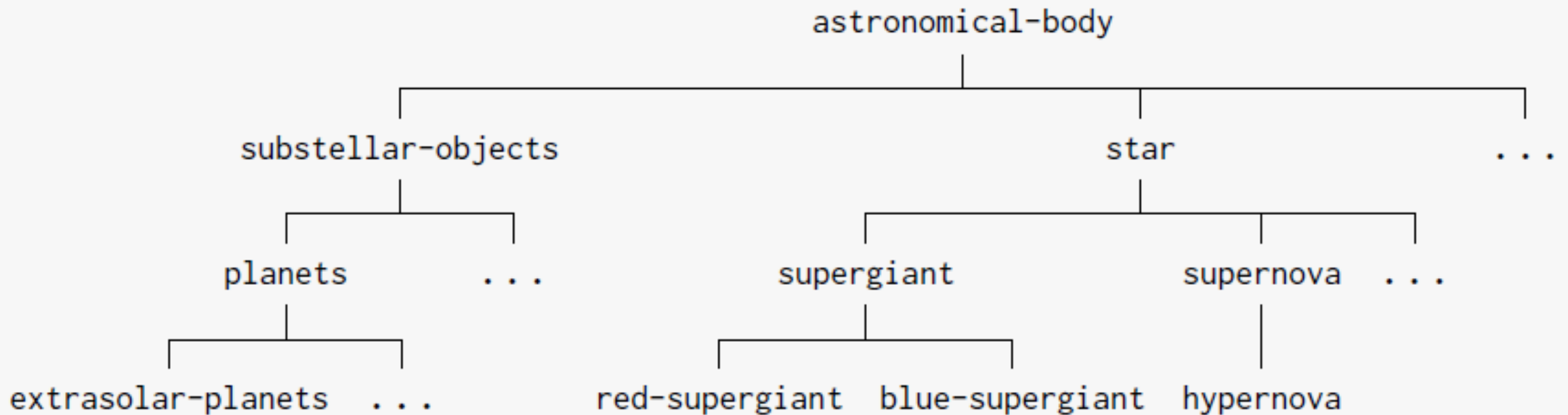
Wikidata

- Έστω ότι θέλουμε να μπορούμε να ικανοποιήσουμε ερωτήματα του στυλ: «δώσε μου όλα τα astronomical bodies».



Wikidata

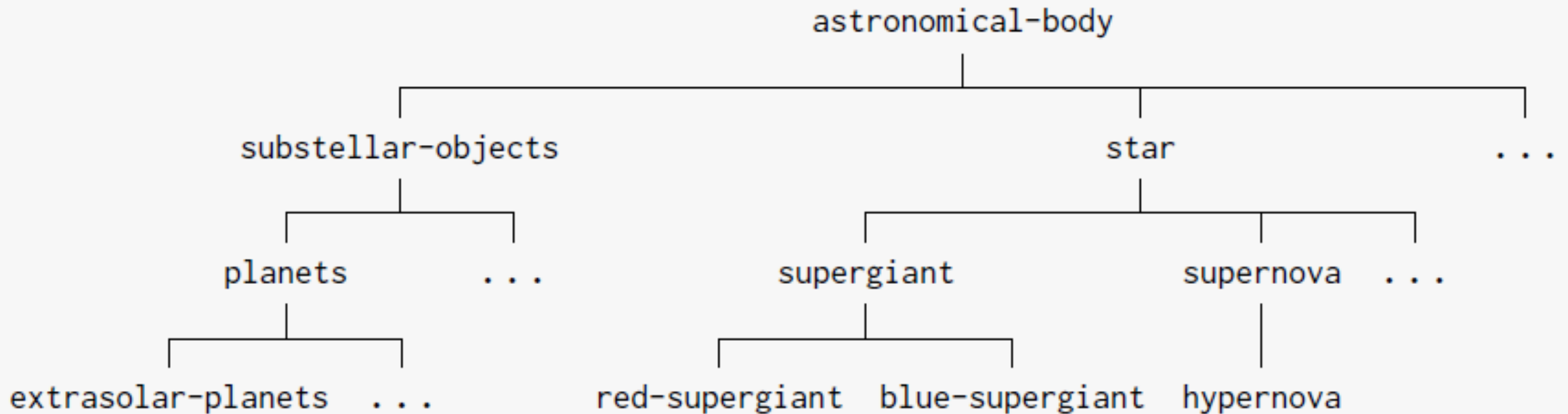
1. Το ερώτημα θα πρέπει να αναφέρεται στους τύπους όλης της ιεραρχίας
2. Οι ορισμοί των αντικειμένων θα πρέπει να περιέχουν όλους τους υπερ-τύπους, π.χ. `<x a planet>`, `<x a substellar-object>`,





Wikidata

3. Οι σχέσεις ανάμεσα στους τύπους μπορούν να δίνονται από τους ειδικούς και να λαμβάνονται υπόψη αυτόματα στην διαδικασία απάντησης ερωτημάτων



Σημασιολογία

- Για να μπορεί να υποστηριχτεί το 3., χρειάζεται ένα μηχανισμός ορισμού της σημασιολογίας (semantics) του λεξιλογίου
 - Κατανοητός από τα προγράμματα (τυπική σημασιολογία)
- Αυτός ο μηχανισμός πρέπει να είναι γενικός
 - Να μπορεί να χρησιμοποιηθεί σε πολλά πεδία
 - Να μπορεί να χρησιμοποιηθεί σε συνδυασμό δεδομένων
- Δύο βασικά πρότυπα έχουν προταθεί για τον ορισμό σημασιολογίας πάνω από την RDF: **RDFS** και **OWL**



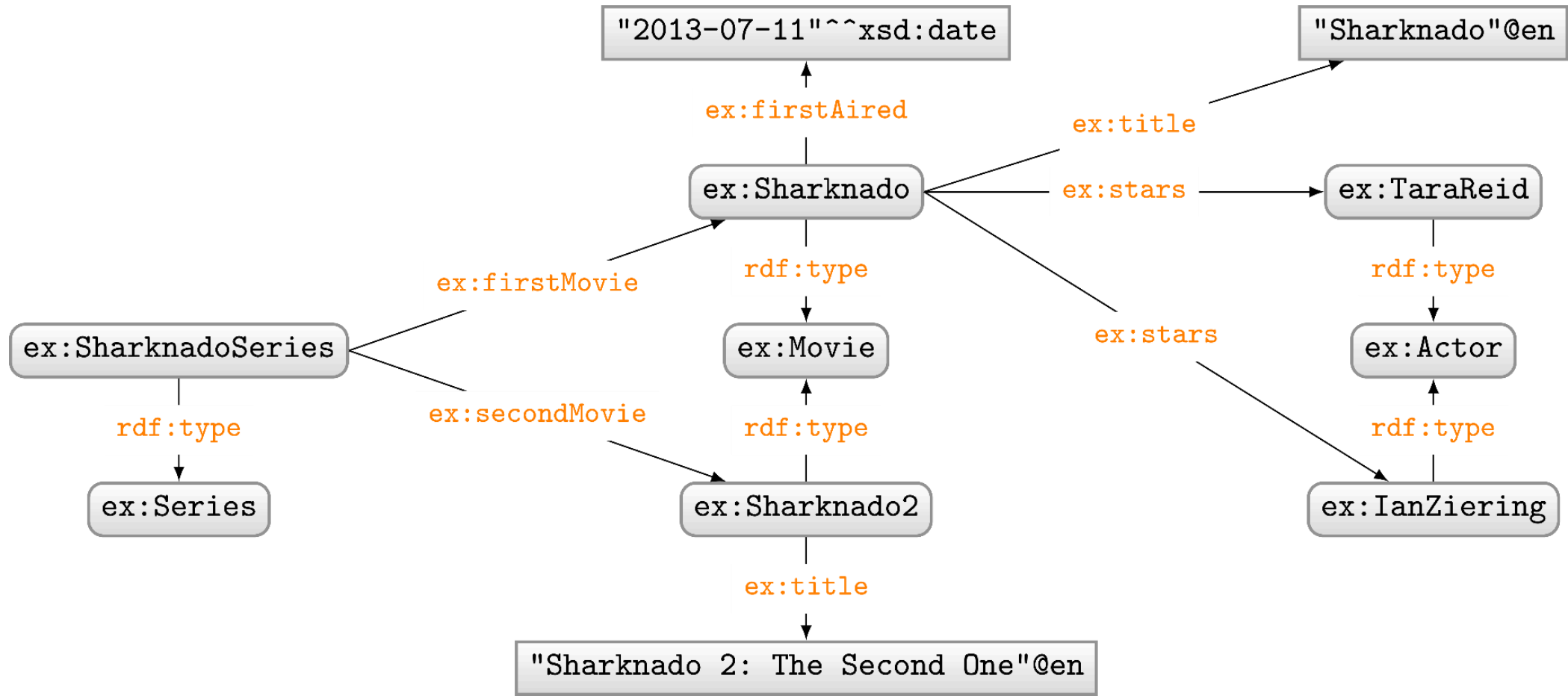
Quiz

- Έστω η τριπλέτα: `:WilliamGolding :wroteNovel :LordOfTheFlies`
- Και το λεξιλόγιο:
 - `:authored,`
`:contributedTo`
 - `:Novelist, :Author,`
`:Person, :Novel,`
`:LiteraryWork`
- Τι θα μπορούσατε να συμπεράνετε;

```
:WilliamGolding rdf:type :Novelist
:WilliamGolding rdf:type :Author
:WilliamGolding rdf:type :Person
:LordOfTheFlies  rdf:type :Novel
:LordOfTheFlies  rdf:type :LiteraryWork
:WilliamGolding :authored :LordOfTheFlies
:WilliamGolding :contributedTo :LordOfTheFlies
```

RDF: Ιδιότητες (properties)

- Πόροι που χρησιμοποιούνται ως κατηγορήματα
 - `rdf:type`, `ex:firstMovie`, `ex:stars`, κτλ.

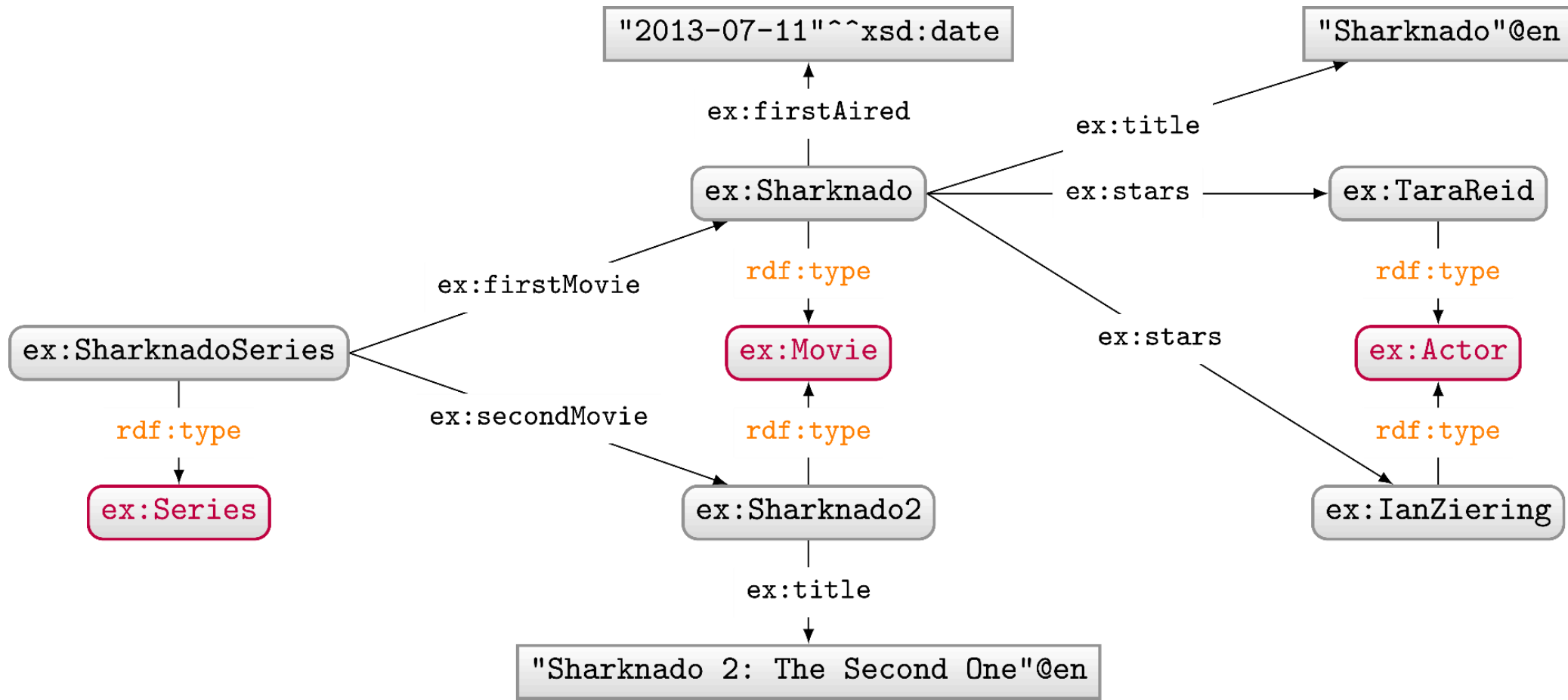


RDF: Κλάσεις (classes)

- Ομαδοποιούν πόρους

- `ex:Movie`, `ex:Actor`, `ex:Series`, κτλ.

- Χρήση του `rdf:type` για να αποδώσουμε τον τύπο ενός πόρου



RDFS: Γενικά

- Η μοντελοποίηση στην RDFS αφορά σύνολα πόρων
 - Στην RDF αφορά συγκεκριμένους πόρους / οντότητες
 - Δημιουργούμε μια δομή γράφου
- Στην ουσία ορίζουμε «κανόνες» και «οδηγίες» για τον ορισμό και την χρήση γράφων
 - Ποιο λεξιλόγιο μπορεί να χρησιμοποιηθεί, σημασιολογία, πώς οι οντότητες σχετίζονται, κτλ.
- Εν ολίγοις, περιγράφει το σχήμα του μοντέλου, τον τρόπο με τον οποίον μπορούμε να περιγράψουμε τα δεδομένα



RDF Schema 1.1

W3C Recommendation 25 February 2014

This version:

<http://www.w3.org/TR/2014/REC-rdf-schema-20140225/>

Latest published version:

<http://www.w3.org/TR/rdf-schema/>

Previous version:

<http://www.w3.org/TR/2014/PER-rdf-schema-20140109/>

Editors:

[Dan Brickley](#), Google

R.V. Guha, Google

Previous Editors:

Brian McBride

Please check the [errata](#) for any errors or issues reported since publication.

This document is also available in this non-normative format: [diff w.r.t. 2004 Recommendation](#)

Γλώσσα Αναπαράστασης της RDFS

- Χρησιμοποιεί την RDF(!) σε αντίθεση με άλλα συστήματα
 - π.χ. βάσεις δεδομένων έχουν διαφορετικό τρόπο ορισμού του σχήματος
- Επομένως «κληρονομεί» και όλα τα πλεονεκτήματα της RDF
 - Διαμοιρασμός, ενοποίηση, κτλ.
- Παράδειγμα:
 - `rdfs:Class` -> καθορισμός ενός συνόλου (κλάσης), δηλαδή ότι «κάτι είναι κλάση»
 - Αυτό ορίζεται με τριπλέτες

```
:AllStarPlayer rdf:type rdfs:Class.  
:MajorLeaguePlayer rdf:type rdfs:Class.  
:Surgeon rdf:type rdfs:Class.  
:Staff rdf:type rdfs:Class.  
:Physician rdf:type rdfs:Class.
```


rdfs:Resource

- Όλα τα αντικείμενα τα οποία περιγράφονται με RDF ονομάζονται πόροι
- Είναι στιγμιότυπα της κλάσης rdfs:Resource.
 - Αυτή η κλάση περιλαμβάνει τα πάντα. Όλες οι άλλες κλάσεις είναι υποκλάσεις της κλάσης αυτής.
- Η rdfs:Resource είναι ένα στιγμιότυπο της rdfs:Class

rdfs:Class

- Προσδιορίζει σύνολο ομοειδών οντοτήτων (οντότητες που μοιράζονται τις ίδιες ιδιότητες/χαρακτηριστικά)

```
:Surgeon rdf:type rdfs:Class.  
:Staff rdf:type rdfs:Class.  
:Physician rdf:type rdfs:Class.  
:Patient rdf:type rdfs:Class.
```



rdfs:subClassOf

- Άφορά κλάσεις και δηλώνει σχέση υπαγωγής (κλάσης υπο-κλάσης).
 - Κάθε στοιχείο της υπο-κλάσης είναι ταυτόχρονα και στοιχείο της κλάσης.

Για παράδειγμα:

(RDFS)

`:Pregnant rdfs:subClassOf :Female .`

`:FemalePatient rdfs:subClassOf :Female.`

(RDF)

`:maria rdf:type :Pregnant .`

Τι σημαίνει:

«Αν x ανήκει στο A και το A είναι υποσύνολο του B, τότε το x ανήκει και στο B.»

rdfs:subClassOf

- Είναι μεταβατική ιδιότητα

:VeganRecipe rdfs:subClassOf :VegetarianRecipe

:VegetarianRecipe rdfs:subClassOf :Recipe

:LemonPie rdf:type :VeganRecipe



:LemonPie rdf:type :VegetarianRecipe

:LemonPie rdf:type :Recipe

| Quiz

Αυτό είναι σωστό;

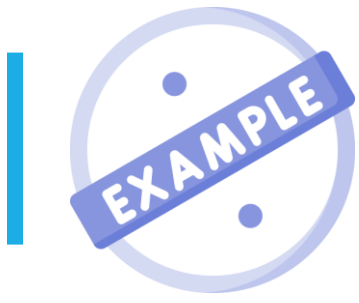
```
:VeganRecipe rdfs:subClassOf :VeganRecipe
```

rdfs:subPropertyOf

- Αφορά ιδιότητες (properties) και δηλώνει εξειδίκευση ιδιότητας: η περισσότερο εξειδικευμένη ιδιότητα είναι rdfs:subPropertyOf της γενικότερης
 - :brother rdfs:subPropertyOf :relative
- Σημασιολογία
 - Αν $\langle x P y \rangle$ και $\langle P \text{ subPropertyOf } R \rangle$, τότε ισχύει και $\langle x R y \rangle$
- Είναι μεταβατική ιδιότητα
- Κάθε ιδιότητα είναι υπο-ιδιότητα του εαυτού της

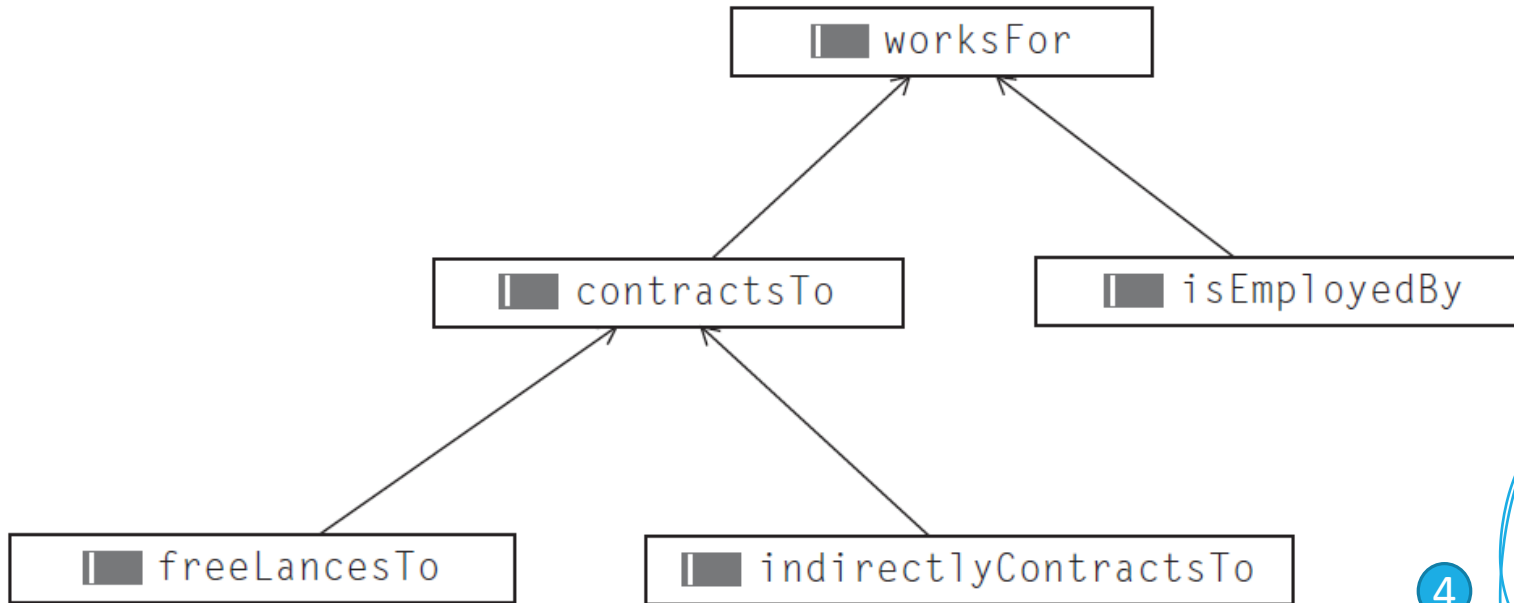


- Μια εταιρεία απασχολεί διάφορους ανθρώπους σε διάφορες θέσεις
 - Απασχολούνται άμεσα από την εταιρεία, με συμβόλαιο (ελεύθερος επαγγελματίας, μακροχρόνια σύμβαση, μέσω τρίτης εταιρείας, κτλ.)
- Υπάρχουν διάφορες σχέσεις
 - `contractsTo`, `freeLancesTo`, `indirectlyContractsTo`, `isEmployedBy`, `worksFor`
- Κάποιες χρήσιμες συσχετίσεις
 - Όλοι οι παραπάνω εργάζονται για την εταιρεία (`worksFor`)
 - Ένας ελεύθερος επαγγελματίας και ένας που έχει συμβόλαιο έχουν συμβόλαιο και εργάζονται για την εταιρεία



Στιγμιότυπα

A:Goldman :isEmployedBy :TheFirm.
B:Spence :freeLancesTo :TheFirm.
C:Long :indirectlyContractsTo :TheFirm.



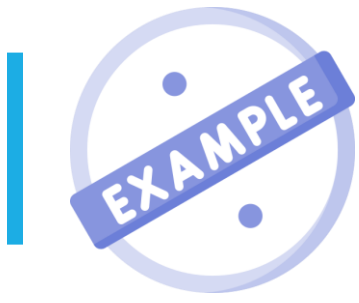
- 1 :freeLancesTo rdfs:subPropertyOf contractsTo.
- 2 :indirectlyContractsTo rdfs:subPropertyOf contractsTo.
- 3 :isEmployedBy rdfs:subPropertyOf worksFor.
- 4 :contractsTo rdfs:subPropertyOf worksFor.

A 3
:Goldman :worksFor :TheFirm.

B 1
:Spence :contractsTo :TheFirm.

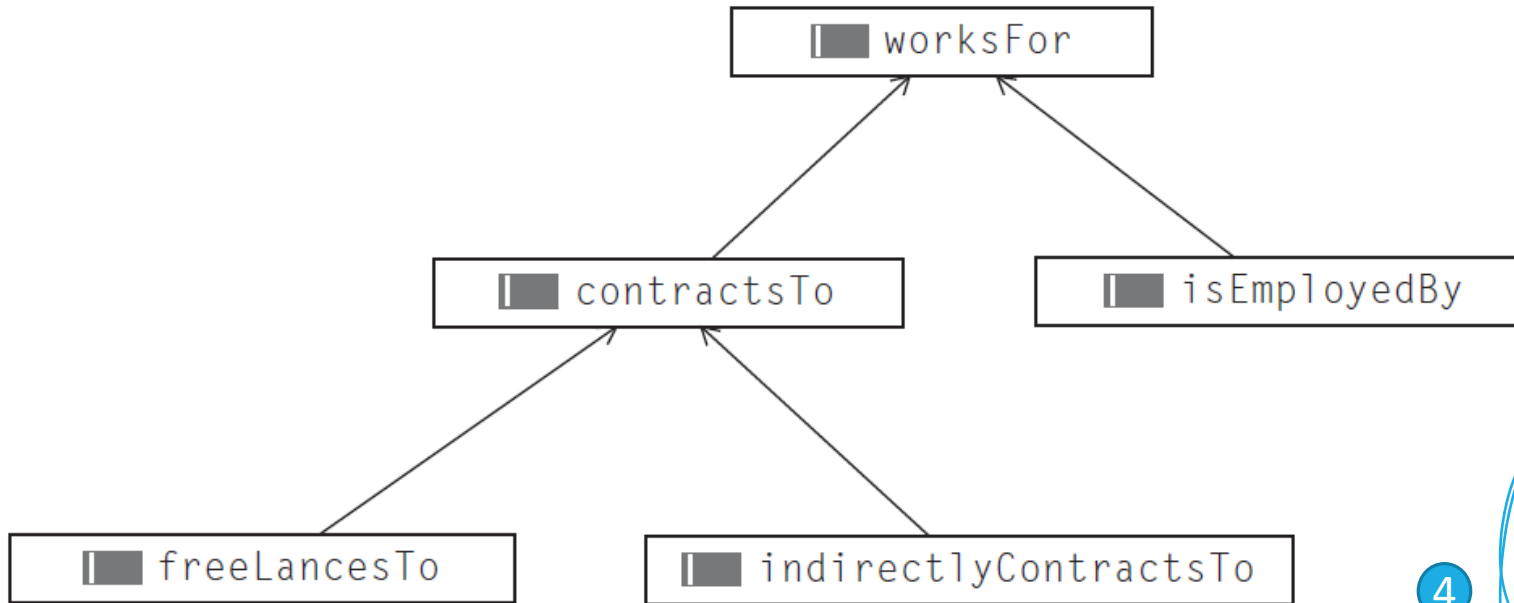
C 2
:Long contractsTo :TheFirm.

4
:Spence :worksFor :TheFirm.
:Long :worksFor :TheFirm.



Στιγμιότυπα

- A: Goldman :isEmployedBy :TheFirm.
- B: Spence :freeLancesTo :TheFirm.
- C: Long :indirectlyContractsTo :TheFirm.




- 1 :freeLancesTo rdfs:subPropertyOf contractsTo.
- 2 :indirectlyContractsTo rdfs:subPropertyOf contractsTo.
- 3 :isEmployedBy rdfs:subPropertyOf worksFor.
- 4 :contractsTo rdfs:subPropertyOf worksFor.

A 3
:Goldman :worksFor :TheFirm.

B 1
:Spence :contractsTo :TheFirm.

C 2
:Long contractsTo :TheFirm.

4
:Spence :worksFor :TheFirm.
:Long :worksFor :TheFirm.

 :Goldman :isEmployedBy :TheFirm.

TheFirm.

Firm.

eFirm.

rm.

irm.

m.

Το `rdfs:subPropertyOf` δεν έχει ανάλογο στον αντικειμενοστραφή προγραμματισμό, όπου οι σχέσεις δεν είναι οντότητες πρώτης-τάξης (δεν μπορούν να συσχετιστούν μεταξύ τους, ανεξαρτήτως της κλάσης στην οποία εφαρμόζονται).

Για το λόγο αυτό, σε αντίθεση με την περίπτωση της `rdfs:subClassOf`, οι προγραμματιστές αντικειμενοστραφούς λογισμικού δεν αντιμετωπίζουν σύγχυση με κάποια γνωστή τους έννοια.

Η μόνη περίπτωση σύγχυσης είναι όταν τα διαγράμματα υπο-ιδιοτήτων μερικές φορές θεωρούνται διαγράμματα κλάσεων.

- 1 :freeLa
- 2 :indire
- 3 :isEmpl
- 4 :contra

freeLan

■ `rdfs:domain` – `rdfs:range`

- Τρόπος χρήσης των ιδιοτήτων σε σχέση με τις κλάσεις
 - Το `subject` ή το `object` της τριπλέτας ανήκει σε μια κλάση
- Πεδίο ορισμού μια συνάρτησης/σχέσης (`domain`)
 - Το σύνολο τιμών από το οποίο λαμβάνει τιμές μια συνάρτηση
- Πεδίο τιμών (`range`)
 - το σύνολο των τιμών

`P rdfs:domain D.`

`P rdfs:range R.`

| Quiz

:suffersFromDisease

?

:Patient .

:suffersFromDisease

?

:Disease .


| Quiz


```
:suffersFromDisease    rdfs:domain    :Patient .  
:suffersFromDisease    ?              :Disease .
```

| Quiz

```
:suffersFromDisease    rdfs:domain    :Patient .  
:suffersFromDisease    rdfs:range     :Disease .
```

Σημασιολογία

`?P rdfs:domain ?D .`  `?x rdf:type ?D .`
`?x ?P ?y .`

`?P rdfs:range ?D .`  `?y rdf:type ?D .`
`?x ?P ?y .`

Παράδειγμα

`:hasInfenction rdfs:domain :Patient .`
`:hasInfenction rdfs:range :Infenction .`
`:Marcus :hasInfenction :Hepatitis_A .`



`:Marcus rdf:type :Patient`
`:Hepatitis_A rdf:type :Infection .`



Προσοχή!

- Τα domain/range δεν είναι περιορισμοί (constraints)
- Στην RDFS δεν μπορούμε να δηλώσουμε ότι ένα στιγμιότυπο ΔΕΝ είναι μέλος μιας κλάσης
- Δεν υπάρχει η έννοια του ασυνεπούς συμπερασμού
- Περιγράφει το schema άλλα δεν κάνει δηλώσεις για την εγκυρότητα του αποτελέσματος



```
ex:hasIngredient rdfs:range ex:Ingredient  
ex:LemonPie ex:hasIngredient ex:Lemon  
ex:Lemon rdf:type ex:Colour
```

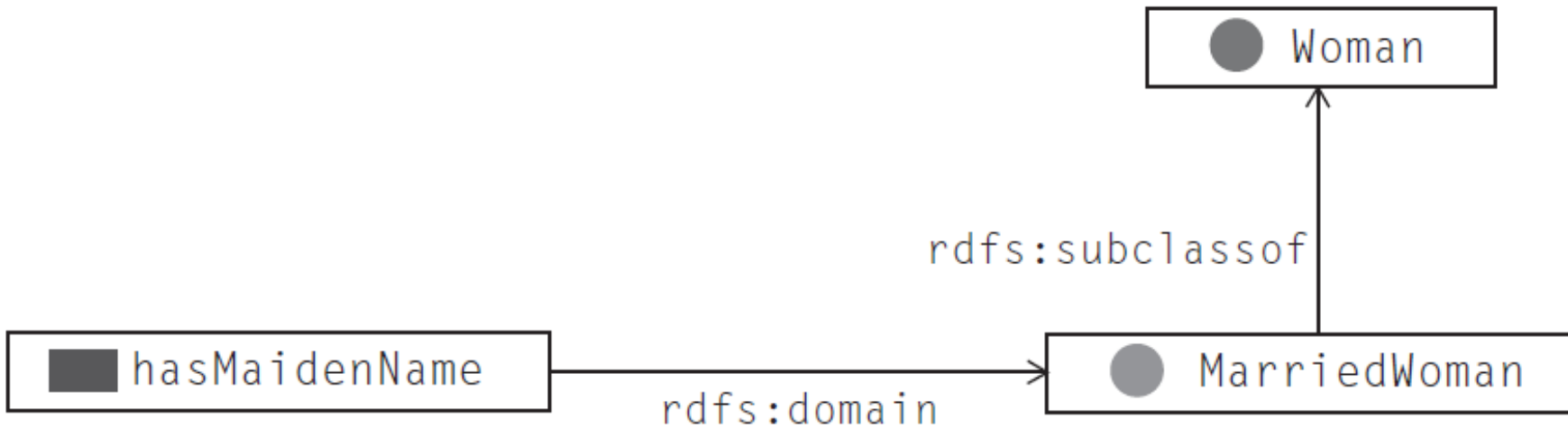
- Το παραπάνω δεν είναι πρόβλημα στην RDFS

```
ex:Lemon rdf:type ex:Ingredient
```



Συνδυασμός domain/range με subClassOf

- Τριπλέτα `:MarriedWoman` `rdfs:subClassOf` `:Woman`
- Ιδιότητα: `hasMaidenName` με domain το `MarriedWoman`



- Σύμφωνα με την σημασιολογία του domain: «αν κάτι χρησιμοποιήσει την ιδιότητα `hasMaidenName`, τότε θα πρέπει να είναι `MarriedWoman`»

`:Karen :hasMaidenName "Stephens".` ➡ `:Karen rdf:type :MarriedWoman.`



Συνδυασμός domain/range με subClassOf

- Τι άλλο μπορεί να εξαχθεί;

:Karen rdf:type :Woman.

- Οπότε γενικά καταλαβαίνουμε ότι:

?X :hasMaidenName ?Y -> ?X rdf:type :Woman

- Το πάνω δεν είναι ο ορισμός του domain; Οπότε έχουμε και:

:hasMaidenName rdfs:domain :Woman.

**?P rdfs:domain ?D .
?D rdfs:subClassof ?C ➡ ?P rdfs:domain ?C .**



Συνδυασμός domain/range με subClassOf

- Τι άλλο πρέπει να εξαχθεί;

- Οπότε

?X :hasMaidenName

Η σημασιολογία μπορεί να εφαρμοστεί και σε
επίπεδο σχήματος, όχι μόνο αντικειμένων

- Το πάνω δεν είναι ο ορισμός του domain; Οπότε έχουμε και:

:hasMaidenName rdfs:domain :Woman.

```
?P rdfs:domain ?D .  
?D rdfs:subClassof ?C
```



```
?P rdfs:domain ?C .
```

| Προσοχή

- Τι περίεργο είδατε στο προηγούμενο παράδειγμα;
- Μήπως κάτι που δεν «ταιριάζει» στον αντικειμενοστραφή προγραμματισμό;

`:hasMaidenName` `rdfs:domain` `:MarriedWoman`

(μετά από συμπερασμό)

`:hasMaidenName` `rdfs:domain` `:Woman`

- Στην ουσία ορίστηκε η αρχική σχέση σε μια κλάση ψηλότερα στην ιεραρχία

! Προσοχή

- Τι περίεργο είδατε στο προηγούμενο παράδειγμα;
- Μήπως κάτι που δεν «ταιριάζει» στον αντικειμενοστραφή προγραμ

:hasMa

(μετά α

Οι ιδιότητες δεν ορίζονται σε κλάσεις. Υπάρχουν από μόνες τους. Αυτά που εξάγονται είναι συσχετίσεις.

:hasMaidenName rdfs:domain :Woman

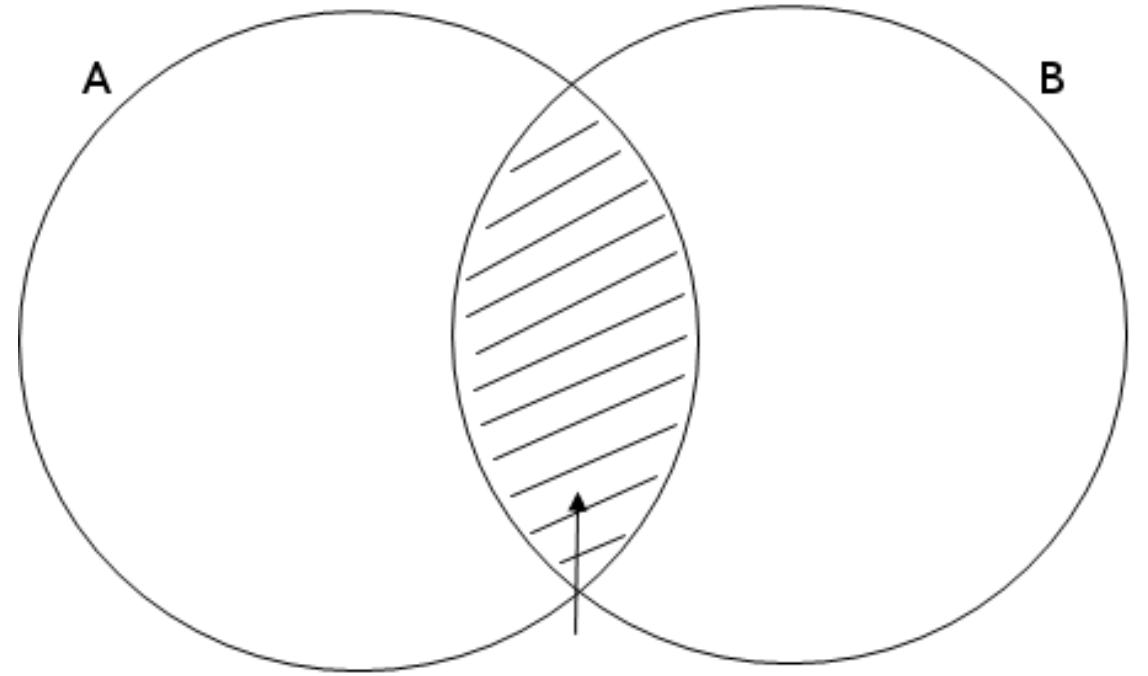
- Στην ουσία ορίστηκε η αρχική σχέση σε μια κλάση ψηλότερα στην ιεραρχία

Τομή κλάσεων

- Αναφέραμε ότι οι κλάσεις στην RDFS είναι στην ουσία σύνολα
- Κάποιος θα ρωτούσε τότε: Μπορώ να μοντελοποιήσω την τομή;
- Απάντηση: ΌΧΙ
- Δεν υπάρχει κάποιο δομικό στοιχείο στην γλώσσα (σε αντίθεση με την OWL)
- Όμως μπορεί να το κάνει έμμεσα

Τομή κλάσεων

- $C \subseteq A \cap B$
- Σε επίπεδο RDF: αν κάτι ανήκει στην κλάση C, πρέπει να ανήκει και στις A και B.
- Κάποιος σας ζητάει να αναπαραστήσετε αυτήν την σχέση με αυτά που γνωρίζετε. Πώς θα το κάνατε;



```
:C rdfs:subClassOf :A.  
:C rdfs:subClassOf :B.
```


Τομή κλάσεων

```
:C rdfs:subClassOf :A.  
:C rdfs:subClassOf :B.
```

- Πώς οι προηγούμενες σχέσεις υποστηρίζουν την τομή;
- Παράδειγμα

?x rdf:type C



?

Τομή κλάσεων

```
:C rdfs:subClassOf :A.  
:C rdfs:subClassOf :B.
```

- Πώς οι προηγούμενες σχέσεις υποστηρίζουν την τομή;
- Παράδειγμα

```
?x rdf:type C
```



```
?x rdf:type :B.  
?x rdf:type :A.
```



Δεξιότητες Νοσοκομείου

- Νοσοκόμοι / ες, γιατροί, χειρουργοί, διοικητικοί, εθελοντές, κτλ.
- Χειρουργοί (Surgeon)
 - Μέλη του προσωπικού (Staff)
 - Εξειδικευμένοι γιατροί (Physician)

$$Surgeon \subseteq Staff \cap Physician$$



Δεξιότητες Νοσοκομείου

```
:Surgeon rdfs:subClassOf :Staff.  
:Surgeon rdfs:subClassOf :Physician.  
:Kildare rdf:type :Surgeon.
```



```
:Kildare rdf:type :Staff.  
:Kildare rdf:type :Physician.
```

- Προσοχή! Το ανάποδο δεν ισχύει (αν κάποιος είναι μέλος του προσωπικού και γιατρός, δεν σημαίνει ότι είναι και χειρουργός)

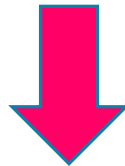
Τομή Ιδιοτήτων

- Και οι ιδιότητες θεωρούνται σύνολα!
- Παρόμοια λογική με τις κλάσεις
- Ένωση και τομή (`rdfs:subPropertyOf`)
- Προσοχή: Όπως και στις κλάσεις, η ένωση και η τομή προσεγγίζονται μέσω της `rdfs:subPropertyOf`

Τομή Ιδιοτήτων

- $P \subseteq R \cap S$
- Σε επίπεδο RDF: αν δύο πόροι σχετίζονται μέσω της ιδιότητας P , τότε σχετίζονται και μέσω των ιδιοτήτων R και S .

$?x :P ?y .$



$?x :R ?y .$

$?x :S ?y .$



Ασθενείς σε δωμάτια νοσοκομείου

- Όταν ένας ασθενής εισάγεται σε ένα δωμάτιο, μπορούμε να συνάγουμε ένα αριθμό πραγμάτων σχετικά με τον ασθενή:
 - Ότι ο ασθενής θα βρίσκεται στο δωμάτιο αυτό
 - Ότι η ασφάλειά του θα χρεωθεί για το δωμάτιο αυτό.
- Πως μπορούμε να εκφράσουμε ότι αυτά συνάγονται από την εισαγωγή του ασθενή στο δωμάτιο;

`:lodgedIn rdfs:subPropertyOf :billedFor.`

`:lodgedIn rdfs:subPropertyOf :assignedTo.`



Ασθενείς σε δωμάτια νοσοκομείου

`:lodgedIn rdfs:subPropertyOf :billedFor.`

`:lodgedIn rdfs:subPropertyOf :assignedTo.`

`:Marcus :lodgedIn :Room101.`



`:Marcus :billedFor :Room101.`

`:Marcus :assignedTo :Room101.`

- Δεν μπορούμε να συνάγουμε συμπεράσματα στην αντίθετη κατεύθυνση.
- Αν γνωρίζαμε ότι «Marcus is billedFor» και ότι «assignedTo Room101», δεν υπάρχει κανόνας που να εφαρμόζεται και επομένως δεν συνάγονται συμπεράσματα

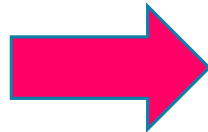
Ένωση Κλάσεων

- Χρησιμοποιώντας ένα παρόμοιο πρότυπο (pattern) με αυτό της τομής, μπορούμε να εκφράσουμε κάποια πράγματα σχετικά με την ένωση συνόλων στην RDFS.
- $A \cup B \subseteq C$
- Τι σημαίνει αυτό σε RDFS;
 - Αν κάτι ανήκει στην A ή στην B, τότε ανήκει και στην C

?x rdf:type A.

ή

?x rdf:type B.



?x rdf:type C



All Stars

- Αποφασίζοντας τους υποψήφιους σε ένα All-Stars διαγωνισμό, οι κανόνες μπορεί να δηλώνουν ότι επιλέγονται
 - μεταξύ των παικτών που έχουν χαρακτηριστεί ως οι πλέον πολύτιμοι παίκτες (Most Valuable Player - MVP)
 - αυτοί που έχουν κορυφαία επίδοση (TopScorer) στο διαγωνισμό

`:MVP rdfs:subClassOf :AllStarCandidate.`

`:TopScorer rdfs:subClassOf :AllStarCandidate.`



All Stars

- Αν γνωρίζουμε ότι ο Reilly είναι MVP και ότι ο Kaneda είναι TopScorer...

```
:Reilly rdf:type :MVP.
```

```
:Kaneda rdf:type :TopScorer.
```

- ... τότε, μπορούμε να συνάγουμε ότι και οι δύο είναι AllStarCandidate:

```
:Reilly rdf:type :AllStarCandidate.
```

```
:Kaneda rdf:type :AllStarCandidate.
```

! Προσοχή

- Όπως και στην τομή, μπορούμε να εξάγουμε συμπεράσματα μόνο προς μια κατεύθυνση. Δηλαδή μπορούμε να ελέγξουμε ότι $MVP \cup TopScorer \subseteq AllStarCandidate$, αλλά όχι την αντίθετη κατεύθυνση.
- Συμπερασματικά, μπορούμε να χρησιμοποιήσουμε την `rdfs:subClassOf` για να κάνουμε δηλώσεις σχετικά με την τομή και την ένωση ως εξής:
- $C \subseteq A \cap B, A \cup B \subseteq C$

Ένωση ιδιοτήτων

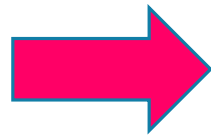
- Παρόμοιο σκεπτικό με την `rdfs:subClassOf`
- Αν δύο διαφορετικές πηγές χρησιμοποιούν ιδιότητες `P` και `Q` με παρόμοιο τρόπο, τότε μπορούμε να ορίσουμε μια νέα ιδιότητα `R` μέσω της σχέσης `rdfs:subPropertyOf` ως εξής:

`P rdfs:subPropertyOf R.`
`Q rdfs:subPropertyOf R.`

`x P y.`

ή

`x Q y.`



`x R y.`



Συνένωση Εγγραφών Βιβλιοθηκών

- Έστω ότι μια βιβλιοθήκη χρησιμοποιεί την `borrow`s για να αναπαραστήσει ότι κάποιος έχει δανειστεί ένα βιβλίο.
- Μια άλλη βιβλιοθήκη χρησιμοποιεί την ιδιότητα `checkedOut`
- Τι θα μπορούσαμε να κάνουμε για να συνενώσουμε τα δεδομένα;



Συνένωση Εγγραφών Βιβλιοθηκών

- Αν είμαστε σίγουροι ότι οι δύο ιδιότητες **έχουν ακριβώς την ίδια σημασία**:  **OWL2**

`Library1:borrows rdfs:subPropertyOf Library2:checkedOut.`
`Library2:checkedOut rdfs:subPropertyOf Library1:borrows.`

- Αν όχι:

`Library1:borrows rdfs:subPropertyOf :hasPossession.`
`Library2:checkedOut rdfs:subPropertyOf :hasPossession.`



Εναρμόνιση ορολογιών

- Έστω ότι ένα παγιωμένο βιβλιογραφικό σύστημα χρησιμοποιεί τον όρο **author** για να δηλώσει το πρόσωπο που δημιούργησε ένα βιβλίο

■ Dublin Core

- Σύνολο από χαρακτηριστικά που χρησιμοποιούνται για την περιγραφή βιβλιογραφικής πληροφορίας για συστήματα βιβλιοθηκών.
- Ένας από τους πλέον χρησιμοποιούμενους όρους είναι ο **dc:creator**
- Πώς μπορούμε να εναρμονίσουμε τα δεδομένα με το Dublin Core;

Index of Terms

Properties in the /terms/ namespace:	abstract , accessRights , accrualMethod , accrualPeriodicity , accrualPolicy , alternative , audience , available , bibliographicCitation , conformsTo , contributor , coverage , created , creator , date , dateAccepted , dateCopyrighted , dateSubmitted , description , educationLevel , extent , format , hasFormat , hasPart , hasVersion , identifier , instructionalMethod , isFormatOf , isPartOf , isReferencedBy , isReplacedBy , isRequiredBy , issued , isVersionOf , language , license , mediator , medium , modified , provenance , publisher , references , relation , replaces , requires , rights , rightsHolder , source , spatial , subject , tableOfContents , temporal , title , type , valid
Properties in the /elements/1.1/ namespace:	contributor , coverage , creator , date , description , format , identifier , language , publisher , relation , rights , source , subject , title , type
Vocabulary Encoding Schemes:	DCMIType , DDC , IMT , LCC , LCSH , MESH , NLM , TGN , UDC
Syntax Encoding Schemes:	Box , ISO3166 , ISO639-2 , ISO639-3 , Period , Point , RFC1766 , RFC3066 , RFC4646 , RFC5646 , URI , W3CDTF
Classes:	Agent , AgentClass , BibliographicResource , FileFormat , Frequency , Jurisdiction , LicenseDocument , LinguisticSystem , Location , LocationPeriodOrJurisdiction , MediaType , MediaTypeOrExtent , MethodOfAccrual , MethodOfInstruction , PeriodOfTime , PhysicalMedium , PhysicalResource , Policy , ProvenanceStatement , RightsStatement , SizeOrDuration , Standard
DCMI Type Vocabulary:	Collection , Dataset , Event , Image , InteractiveResource , MovingImage , PhysicalObject , Service , Software , Sound , StillImage , Text
Terms for vocabulary description:	domainIncludes , memberOf , rangeIncludes , VocabularyEncodingScheme



Εναρμόνιση ορολογιών

`:author rdfs:subPropertyOf dc:creator.`

- Κάθε οντότητα (στιγμιότυπο) για το οποίο ορίζεται η ιδιότητα `author` θα έχει την ίδια τιμή και για την ιδιότητα `dc:creator`.
- Η όλη δουλειά γίνεται από το μηχανισμό συμπερασμού της RDFS
- Παλαιές βάσεις που χρησιμοποιούν το `author` μπορούν να συνεχίσουν να λειτουργούν χωρίς καμία αλλαγή, ενώ νεότερες εφαρμογές μπορούν να χρησιμοποιήσουν τα συναχθέντα δεδομένα για να λειτουργήσουν σύμφωνα με τα πρότυπα.

Παράδειγμα

- <http://rdfplayground.dcc.uchile.cl/>

Quiz

- Πώς μπορούμε να «επιβάλουμε» τη δήλωση ότι **οποιοδήποτε μέλος μιας κλάσης θα χρησιμοποιηθεί και ως μέλος μια άλλης**;
 - Π.χ. ο όρος «ερευνητής» (**Researcher**) είναι μια ιδιαίτερη περίπτωση «αναλυτή» (**Analyst**).
 - Πώς μπορούμε να αναπαραστήσουμε αυτό το γεγονός στην RDFS;
- Υπάρχει ένας αριθμός προσεγγίσεων στο θέμα αυτό, που εξαρτώνται από τις λεπτομέρειες της κάθε κατάστασης.
- Όλες οι λύσεις μπορούν να προσεγγιστούν χρησιμοποιώντας τα πρότυπα που έχουμε αναγνωρίσει έως τώρα.

| Quiz

- Αν ο ερευνητής είναι ιδιαίτερη περίπτωση αναλυτή:

```
:Researcher rdfs:subClassOf :Analyst.  
:Wenger rdf:type :Researcher.  
:Wenger rdf:type :Analyst.
```

- Αν η σχέση ισχύει στην αντίστροφη κατεύθυνση:

```
:Analyst rdfs:subClassOf :Researcher.
```

Quiz

- Αν όμως κάποιοι αναλυτές δεν είναι ερευνητές και το ανάποδο, τι γίνεται;
- Χρήση του προτύπου της ένωσης που είδαμε
- $Analyst \cup Researcher \subseteq Investigator$

`:Analyst rdfs:subClassOf :Investigator.`

`:Researcher rdfs:subClassOf :Investigator.`

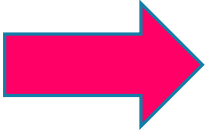
- Έτσι δεν έχουμε δεσμευτεί για ευθεία σχέση μεταξύ αναλυτών και ερευνητών

Quiz

- Έστω ότι αποφασίζουμε ότι οι δύο κλάσεις είναι ισοδύναμες
 - Θα θέλαμε κάθε μέλος της μιας κλάσης να είναι πάντα μέλος της άλλης και το αντίθετο. Πώς;

```
:Analyst rdfs:subClassOf :Researcher.  
:Researcher rdfs:subClassOf :Analyst.
```

```
:Reilly rdf:type :Researcher.  
:Kaneda rdf:type :Analyst.
```



```
:Reilly rdf:type :Analyst.  
:Kaneda rdf:type :Researcher.
```

Ισοδυναμία κλάσεων (class equivalence)

rdfs:label

- Οι πόροι στο σημασιολογικό ιστό καθορίζονται από τα URIs
- Αλλά τα URIs δεν είναι πολύ φιλικά για τον άνθρωπο.
- rdfs:label
 - Στοχεύει στο να δώσει ένα όνομα σε κάθε πόρο.

`msc:courseKM rdfs:label “Διαχείριση και Εξόρυξη Γνώσης” .`



- Υπάρχουν δυο κλάσεις Person και Movie
- Για την κλάση Person μια ιδιότητα καλείται `personName`
- Για την κλάση Movie υπάρχει η ιδιότητα `movieTitle`

```
:Person1 :personName "James Dean".  
:Person2 :personName "Elizabeth Taylor".  
:Person3 :personName "Rock Hudson".  
:Movie1 :movieTitle "Rebel Without a Cause".  
:Movie2 :movieTitle "Giant".
```




- Θα θέλαμε να χρησιμοποιήσουμε ένα μηχανισμό παρουσίασης που χρησιμοποιεί την προηγούμενη τυποποίηση για τη παρουσίαση πληροφορίας σχετικά με ανθρώπους και ταινίες. Πώς μπορούμε να χρησιμοποιήσουμε την RDFS για το σκοπό αυτό;
- Να ορίσουμε κάθε μια από αυτές τις ιδιότητες ως υπο-ιδιότητες της `rdfs:label`

```
:personName rdfs:subPropertyOf rdfs:label.
```

```
:movieTitle rdfs:subPropertyOf rdfs:label.
```



Πλοία

- Έστω ότι μια ναυτιλιακή εταιρεία έχει ένα στόλο από πλοία
- Ο στόλος περιέχει νέα πλοία
 - που βρίσκονται υπό επισκευή
 - που επισκευάζονται
 - που βρίσκονται σε λειτουργία και
 - που έχουν αποσυρθεί



Name	Maiden Voyage	Next Departure	Decommission Date	Destruction Date	Commander
<i>Berengaria</i>	June 16, 1913	–	1938	–	Johnson
<i>QEII</i>	May 2, 1969	March 4, 2010	–	–	Warwick
<i>Titanic</i>	April 10, 1912	–	–	April 14, 1912	Smith
<i>Constitution</i>	July 22, 1798	January 12, 2009	–	–	Preble

ship:Berengaria ship:maidenVoyage "Dec. 16, 1946".
ship:QEII ship:nextDeparture "Mar 4, 2010".



- Επιπρόσθετα της κλάσης `ship:Vessel` μπορούμε να έχουμε υποκλάσεις που αντιστοιχούν στην κατάσταση των πλοίων:

```
ship:DeployedVessel rdfs:subClassOf ship:Vessel.  
ship:InServiceVessel rdfs:subClassOf ship:Vessel.  
ship:OutOfServiceVessel rdfs:subClassOf ship:Vessel.
```

- `DeployedVessel` -> έχει χρησιμοποιηθεί κάποτε
- `InServiceVessel` -> βρίσκεται σε λειτουργία
- `OutOfServiceVessel` -> κάποιο που βρίσκεται εκτός λειτουργίας (πλοία που έχουν αποσυρθεί ή που δεν έχουν ακόμα χρησιμοποιηθεί)

Quiz

- Πώς μπορούμε να κατηγοριοποιήσουμε αυτόματα το κάθε πλοίο στις κατάλληλες υπο-κλάσεις σύμφωνα με την πληροφορία του πίνακα;
 - Αν ένα πλοίο είχε ένα ταξίδι ειδικού σκοπού, τότε αυτό ανήκει στην κλάση `ship:DeployedVessel`
 - Αν η επόμενη αναχώρησή του έχει προγραμματιστεί, τότε ανήκει στην κλάση `ship:InServiceVessel`
 - Αν έχει μια ημερομηνία απόσυρσης ή καταστροφής, τότε ανήκει στην κλάση `ship:OutOfServiceVessel`

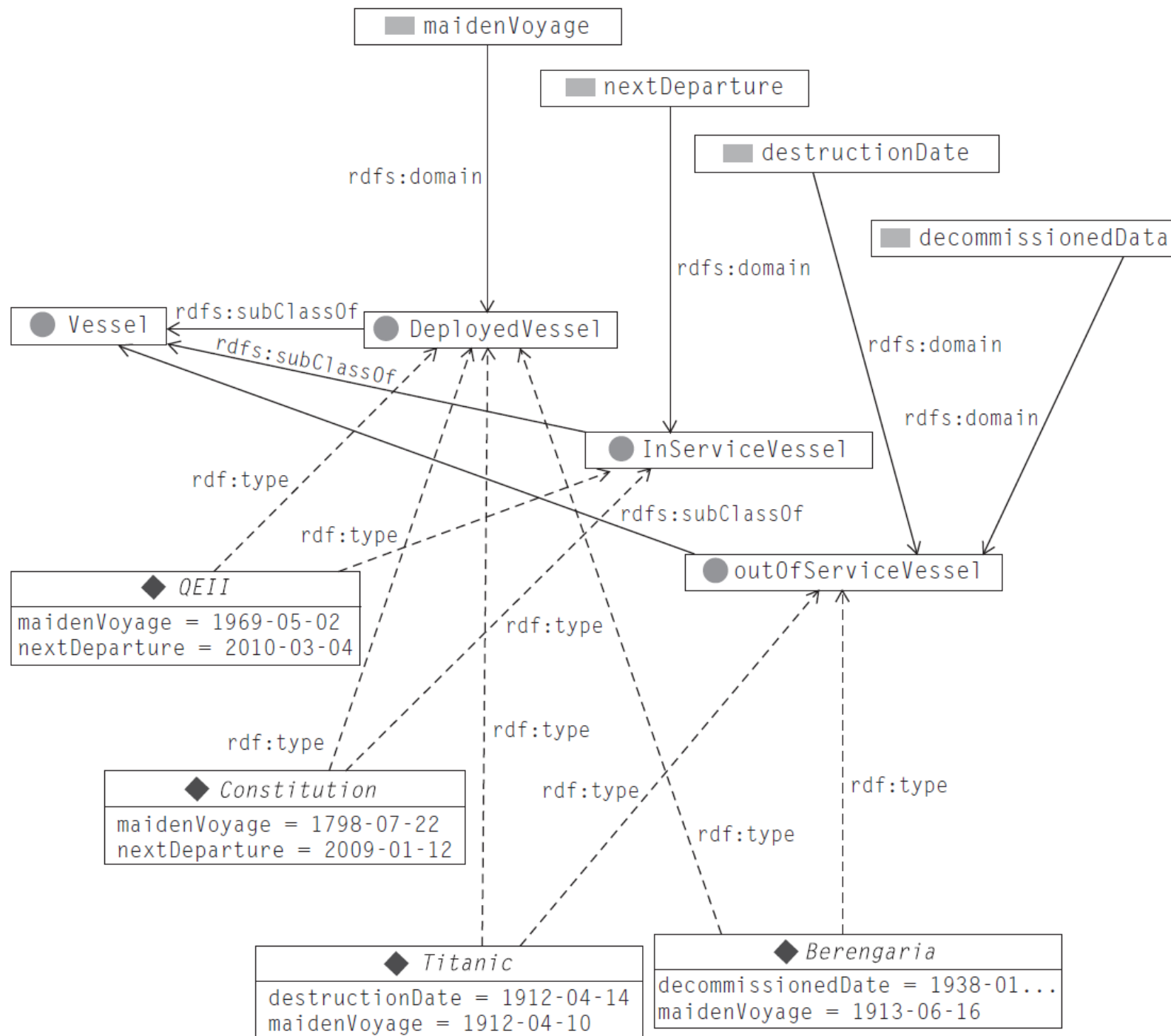
Quiz

- Μπορούμε να κάνουμε τέτοιου είδους συμπερασμούς χρησιμοποιώντας το `rdfs:domain`

```
ship:maidenVoyage rdfs:domain ship:DeployedVessel.  
ship:nextDeparture rdfs:domain ship:InServiceVessel.  
ship:decommissionedDate rdfs:domain ship:OutOfServiceVessel.  
ship:destructionDate rdfs:domain ship:OutOfServiceVessel.
```

- Μπορούν να χρησιμοποιηθούν για να φιλτράρουν πληροφορία
 - π.χ. δώσε μου όλα τα πλοία που είναι να αναχωρήσουν

```
ship:DepartingVessel rdf:type rdfs:Class.  
ship:nextDeparture rdfs:domain ship:DepartingVessel.
```



Πολλαπλά πεδία ορισμών και τιμών

- Στο παράδειγμα της ναυτιλιακής εταιρείας είχαμε δύο ορισμούς για το πεδίο ορισμού `nextDeparture`:

`ship:nextDeparture rdfs:domain DepartingVessel.`

`ship:nextDeparture rdfs:domain InServiceVessel.`

Τι σημαίνει αυτό; Ποιο είναι το πεδίο ορισμού τελικά;

Συμπεράσματα που εξάγονται

Παράδειγμα

```
ship:QEII ship:maidenVoyage "May 2, 1969".  
ship:QEII ship:nextDeparture "Mar 4, 2010".  
ship:QEII ship:hasCommander Warwick.
```



με βάση τα domains

```
ship:QEII rdf:type ship:DepartingVessel .  
ship:QEII rdf:type ship:InServiceVessel.
```

Αυτό στην ουσία είναι τομή κλάσεων!

Προσοχή

- Πολλαπλά πεδία ορισμού στην RDFS σημαίνει τομή των κλάσεων που αναφέρονται
- Στον αντικειμενοστραφή προγραμματισμό, ερμηνεύονται ως ένωση συνόλων
- Για κάποιον που έρχεται με αυτή την αντίληψη, η τομή της RDFS είναι έκπληξη



- Μια εταιρεία διαχειρίζεται μια ομάδα από περιοδεύοντες πωλητές
- Κάθε πωλητής έχει ένα πρόγραμμα για ταξίδια

```
sales:SalesPerson rdfs:subClassOf foaf:Person.  
sales:sells rdfs:domain sales:SalesPerson.  
sales:sells rdfs:range sales:ProductLine.  
sales:nextDeparture rdfs:domain sales:SalesPerson.
```

- Έστω ότι θέλουμε να συγχωνεύσουμε πληροφορία για τη διαχείριση του δυναμικού των πωλητών μαζί με τα προγράμματά τους και τις γραμμές συγκοινωνίας



- Ένα προφανές υποψήφιο ζεύγος για αυτό είναι το `sales:nextDeparture` και `ship:nextDeparture`

```
sales:nextDeparture rdfs:subPropertyOf ship:nextDeparture.  
ship:nextDeparture rdfs:subPropertyOf sales:nextDeparture.
```

- Για να δούμε όμως τι συμπεράσματα εξάγονται από αυτή τη συσχέτιση συγχώνευσης...



- Έστω ότι έχουμε μια τριπλέτα που περιγράφει ένα μέλος της ομάδας πώλησης:

```
sales:Johannes sales:nextDeparture "May 31, 2008".  
ship:QEII ship:nextDeparture "Mar 4, 2010".
```



```
sales:Johannes ship:nextDeparture "May 31, 2008" .  
ship:QEII sales:nextDeparture "Mar 4, 2010".  
sales:Johannes rdf:type ship:DepartingVessel.  
ship:QEII rdf:type sales:SalesPerson.  
ship:QEII rdf:type foaf:Person.
```



Ποιο ήταν το λάθος;

- Δεν θα έπρεπε να συγχωνεύσουμε τις δύο ιδιότητες `nextDeparture` με μια «ισχυρή» δήλωση ισοδυναμίας
 - Τα νοήματά τους έχουν σημαντικές διαφορές
- Θα μπορούσαμε να κάνουμε μια υπερ-ιδιότητα για να κρατά όλες τις αναχωρήσεις των πλοίων και πωλητών

```
ship:nextDeparture rdfs:subPropertyOf cal:nextDeparture.  
sales:nextDeparture rdfs:subPropertyOf cal:nextDeparture.
```



```
sales:Johannes cal:nextDeparture "May 31, 2008" .  
ship:QEII cal:nextDeparture "Mar 4, 2010".
```

Project A

- RDFS

- Περιγραφή εργασίας ([link](#))
- Δήλωση ομάδων ([link](#))

Tutorial – TopBraid Composer

- Activities
 - Running, Sleeping, Walking
- Person
 - Healthy, PersonWithAlzheimer (PwD)

Αναφορές - Πηγές



Αναφορά Δημιουργού - Μη Εμπορική
Χρήση - Παρόμοια Διανομή 4.0 Διεθνές
(CC BY-NC-SA 4.0)

- Allemang Dean, Hendler Jim, 2020. Ο Σημασιολογικός Ιστός για τους Δημιουργούς Οντολογιών. Επιμέλεια ελληνικού κειμένου: Βούρος Γεώργιος, Κώτης Κωνσταντίνος, Σαντιπαντάκης Γεώργιος. Εκδόσεις ΔΙΣΙΓΜΑ. ISBN 978-618-202-007-4.
- The Web of Data (Book), by Aidan Hogan
- Introduction to the Semantic Web – Tutorial
(<https://www.w3.org/2009/Talks/0615-SanJose-tutorial-IH/Slides.pdf>)