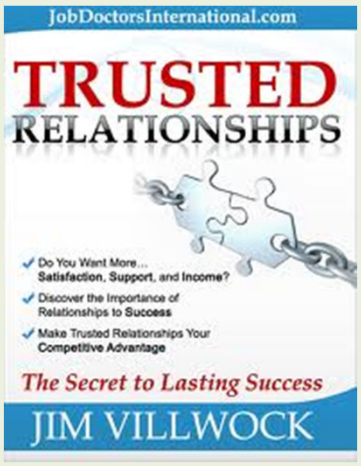
Relationships Matter

Semantic Web Conference

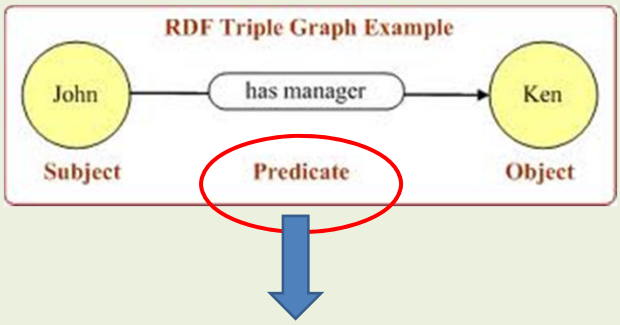
Berlin, 2011

San Francisco, 2012

We will NOT address….

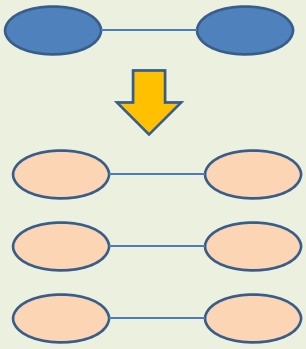


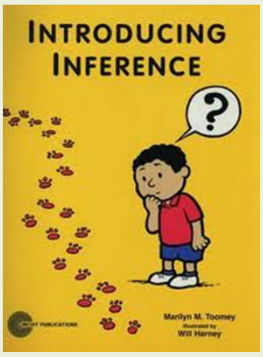
The RDF Triple



The Relationship that matters!

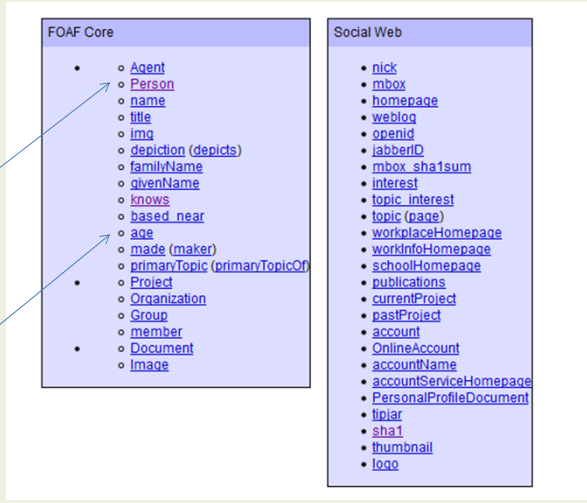
The power of RDF properties to infer new RDF statements?





foaf:
Friend of a Friend Ontology
http://xmlns.com/foaf/spec/

- Designed to provide a standard way of saying things about people and social networks
- Capitalized terms refer to classes/categories of things (foaf:Person)
- Lowercase terms are properties of things (foaf:age, foaf:knows, foaf:name, foaf:nick)



300th Anniversary of Frederick the Great

Frederick II (Frederick The Great)
(1712-1786)

Frederick II

Der Alte Fritz

We will refer to him as :Fred2

Frederick The Great

Ontology Language Stack

OWL

RDFS

N3/RDF

Let's use N3/RDF to say some things about :Fred2 using the foaf: vocabulary

Ancestors of Frederick the Great

Frederick I
(1657-1713)

```
:Fred1   foaf:name "Frederick II".
:Fred1   foaf:gender  "male".
```

Frederick William I
(1688-1740)

```
:FredWilliam   foaf:name
                  "Frederick William 1".
:FredWilliam   foaf:gender  "male".
```

Frederick II (Frederick The Great)
(1712-1786)

```
:Fred2   foaf:name "Frederick II".
:Fred2   foaf:nick  "Frederick The Great".
:Fred2   foaf:gender  "male".
```

Ontology Language Hierarchy

OWL

RDFS

RDF

The rdfs: vocabulary adds properties that we can apply to properties. Remember in RDF world properties are 'things' and we can say things about them. For example, we can talk about properties the same way we talk about functions that have a domain and a range.

Properties defined in the rdfs: vocabulary

rdfs:domain
rdfs:range
rdfs:subPropertyOf

rdfs:domain

:age rdfs:domain foaf:Person .

says that when we have a statement
:rollo :age 22 .
The system will INFER that:
:rollo a foaf:Person .
by adding a new triple to the triple store

rdfs:range

:loves rdfs:range foaf:Person .

says that when we have a statement
:rollo :loves :Marla .
The system will INFER that:
:Marla a foaf:Person .
by adding a new triple to the triple store

Triple:
:Fred1 :entombedIn :BerlinerDom.

:entombedIn

:Fred1

:BerlinerDom

Can we say anything about "things" on the left side of our property?

Persons are entombed.

:entombed rdfs:domain foaf:Person.

:Fred1 rdf:type foaf:Person.

Triple:
:Fred1 :entombedIn :BerlinerDom.

:entombedIn

:Fred1

:BerlinerDom

Persons are entombed in Buildings (of some kind)

Can we say anything about "things" on the right side of our property?

:entombedIn rdfs:range :Building

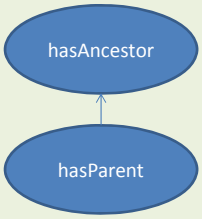
:BerlinerDom rdf:type :Building

rdfs:subPropertyOf

Frederick I
(1657-1713)

:Fred2 :hasParent :FredWilliam1
:FredWilliam1 :hasParent :Fred1

Frederick William I
(1688-1740)



Frederick II
(1712-1786)

:hasParent rdfs:subPropertyOf :hasAncestor

:Fred2 :hasAncestor :FredWilliam1
:FredWilliam1 :hasAncestor :Fred1

Ontology Language Hierarchy

OWL

RDFS

RDF

OWL is a more sophisticated ontology that lets us say even more about properties. We can define certain properties as:

- inverseProperties
- symmetricProperties
- equivalentProperties
- transitiveProperites
- .. and more..

owl:inverseOf

- :Fred2 :hasParent :FredWilliam1

:hasParent

:Fred2 :FredWilliam1

:hasParent owl:inverseOf :hasChild

SELECT ?who
WHERE (:FredWilliamI1 :hasChild ?who) → :Fred2

owl:SymmetricProperty

:**ElisabethChristine** :married :Fred2

:married

:married rdf:type owl:SymmetricProperty

:Fred2 :married :ElisabethChristine

SELECT ?who
WHERE (:Fred2 :married ?who) → : **ElisabethChristine**

owl:inverseOf

:BerlinOperaHouse :commissionedBy :Fred2

:commissionedBy

:commissionedBy owl:inverseOf :commissioned

SELECT ?who
WHERE (?who :commissioned :BerlinOperaHouse)

:Fred2

owl:TransitiveProperty

Frederick I
(1657-1713)

:FredWilliam :ancestorOf :Fred2.
:Fred1 :ancestorOf :FredWilliam.

:ancestorOf rdf:type owl:TransitiveProperty

Frederick William I
(1688-1740)

SELECT ?who
WHERE (?who :ancestorOf :Fred2)

Frederick II
(1712-1786)

FredWilliam, Fred1

owl:EquivalentProperty

Allows us to say that two relationships mean the same thing

:correspondedWith

:Fred2 :Voltaire

:Fred2 :correspondedWith :Voltaire.

:correspondedWith owl:EquivalentPropery :exchangedLetters.

Select ?who
(?who :exchangedLetters :Voltaire) → :Fred2

When Merging Triple Stores equivalence is also available for entities

:Fred2 owl:sameAs bio:*DerAlteFritz*

=

OWL Functional Properties

The square root of four is TWO

The square root of four is ZWEI

The square root of four is DEUX

TWO = ZWEI = DEUX

Functional Property

A functional property is a property that can have only one (unique) object value y for each subject, i.e. there cannot be two distinct values for the 'object'
:hasMother a owl:FunctionalProperty .

:Fred2 :hasMother :SophiaDorotheaOfHanover.
:Fred2 :hasMother :ElectressOfBrandenburg.

:ElectressOfBrandenburg owl:sameAs :SophiaDorotheaOfHanover

If subject is same then object is the same

Inverse Functional Property

If object value is same then subject the same

:SophiaDorotheaOfHanover :motherOf :Fred2.
:ElectressOfBrandenburg :motherOf :Fred2

:ElectressOfBrandenburg owl:sameAs :SophiaDorotheaOfHanover

Summary

- Relationships play a much larger role in data sets than in traditional SQL databases
- Both RDFS and OWL provide support for a rich ontology based on relationships