# Simple Structures in Semantic Modelling

Sponsored by:

CAPSICUM Business Architects

# Purpose of Semantic Data Model

- Semantic data model, other than providing a logical structure for data, provide more meaning of the data. It helps to provide a high level understanding of data by abstracting it further away from physical aspect of data storage.
- Data is modelled in more human readable manner
- Real world concepts are captured through a knowledge graph

# Semantic Web and RDF

# Semantic Web

- Web characteristics
  - Huge amounts of data
  - No central data model
  - Hard to interpret/combine data
- Role of semantic modelling
  - RDF helps manage distributed data
  - Other semantic web standards build on this foundation
  - Each source of data = set of triples
  - Collection of RDF triples from different sources constitute a *knowledge graph*
  - Information from different sources can be easily merged

# Semantic web standards

- Semantic models can in the form of several layers of expressivity
- RDF → RDFS → OWL (increasing levels of semantic expressivity)
- Relationships → Classes (hierarchy) → Reasoning (new knowledge)

# RDF

- **R**esource:
  - A resource can be anything
  - Should be uniquely identifiable and referenced by a URI
- **D**escription:
  - Describes the resources
  - By properties and relationships that link resources
- **F**ramework:
  - A formal (machine readable) semantic model
  - Uses a combination of web based protocols
  - Is domain neutral

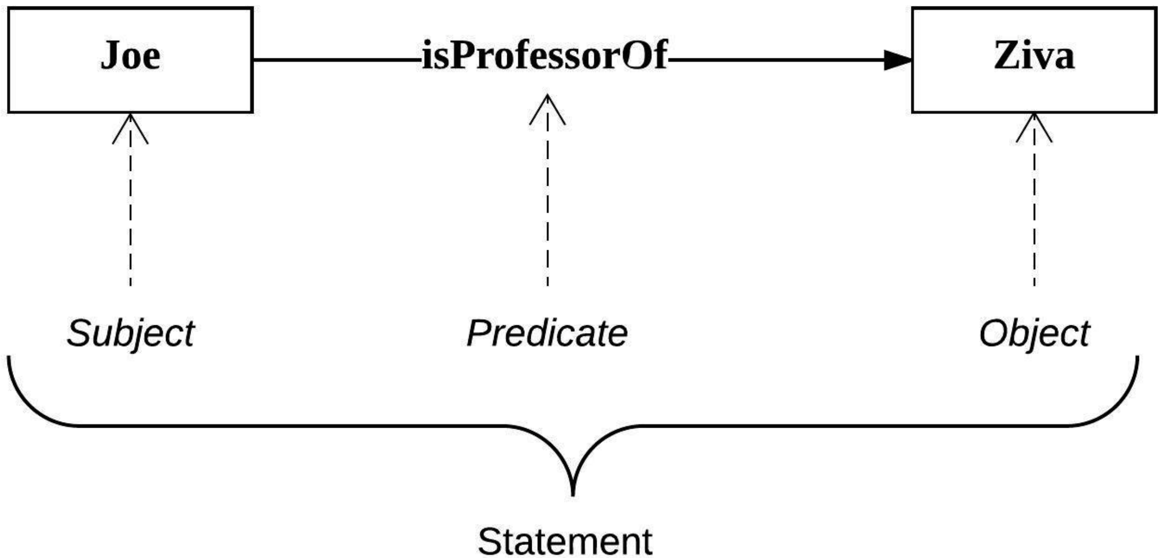
# Basic RDF triple

Example statement: Doctors treat patients

Can be represented by several triples:

- Triple: <Doctor> <treats> <Patient> .
- Triple: <Patient> <hasName> “Jim” .
- Triple: <Appointment> <hasStartTime> <xsd:time> .

- Statement: Joe is Ziva's professor.
- Can be modelled in RDF as a triple: <Joe> <isProfessorOf> <Ziva>



- Based on the context, there are other ways of modelling the same statement such as:

<Joe> <hasStudent> <Ziva>

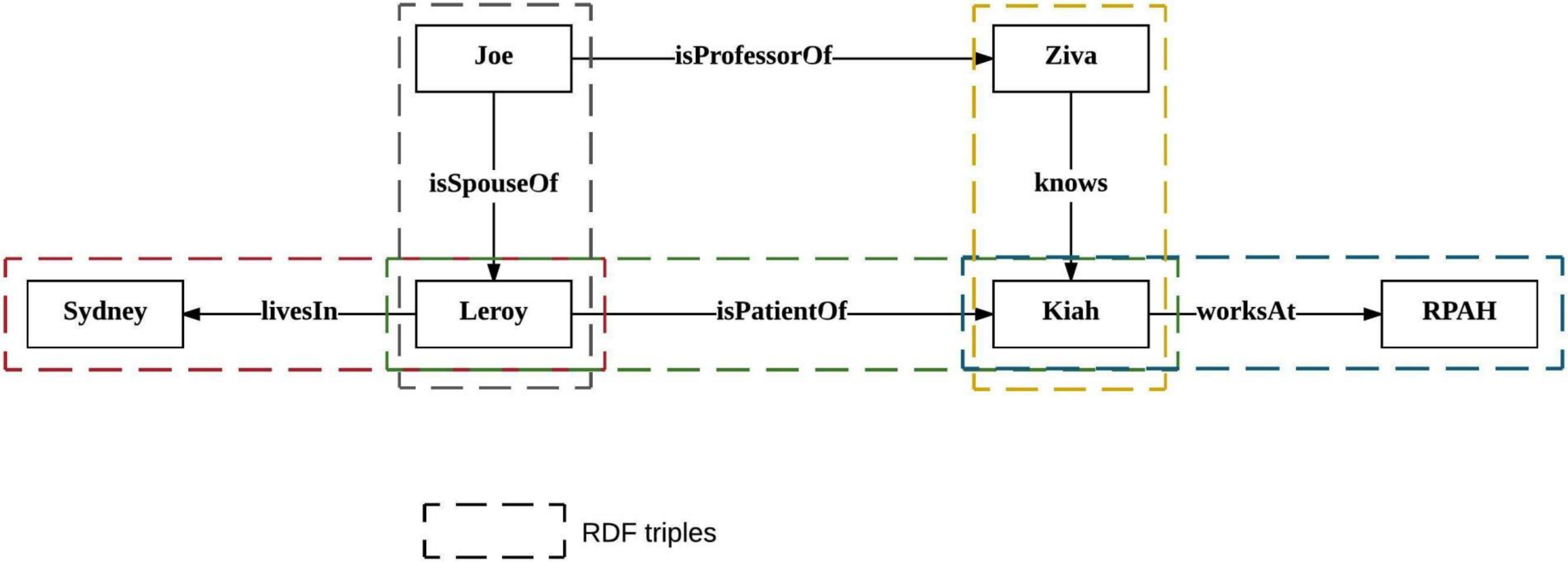
OR

<Ziva> <isStudentOf> <Joe>

OR

<Ziva> <hasProfessor> <Joe>

# Simple RDF Structures



# Example of a knowledge graph

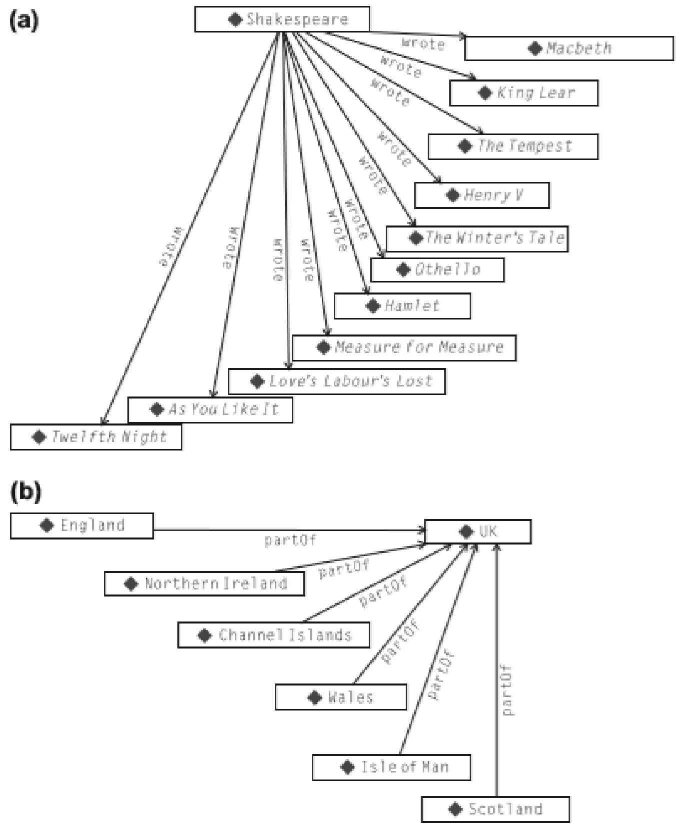



**FIGURE 3.5**

Graphic representation of triples describing (a) Shakespeare's plays and (b) parts of the United Kingdom.

# Merging graphs

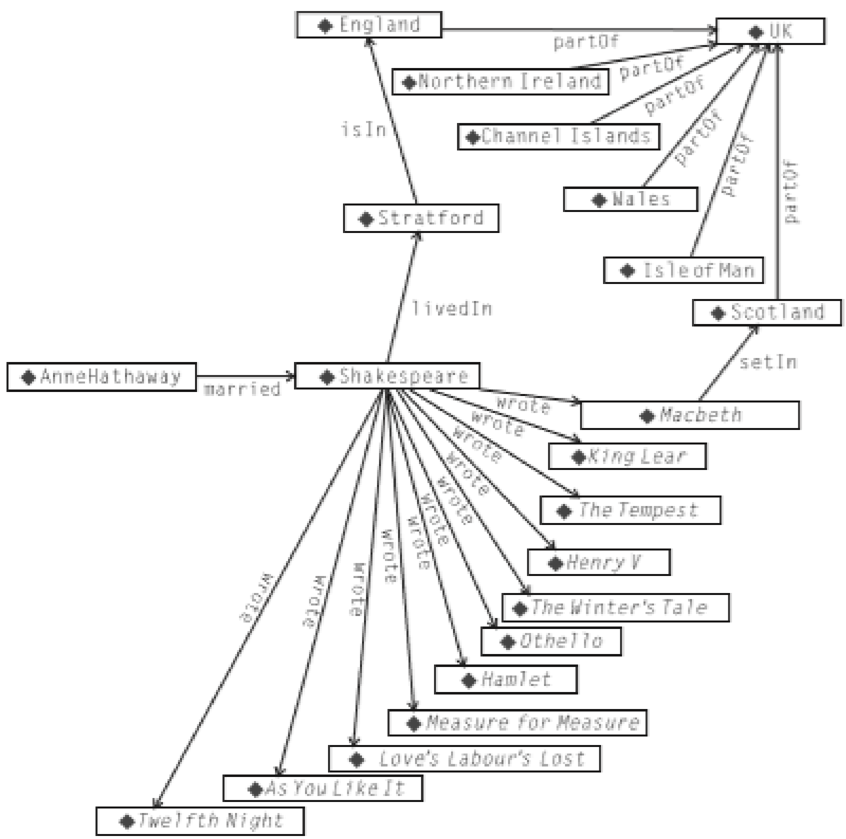


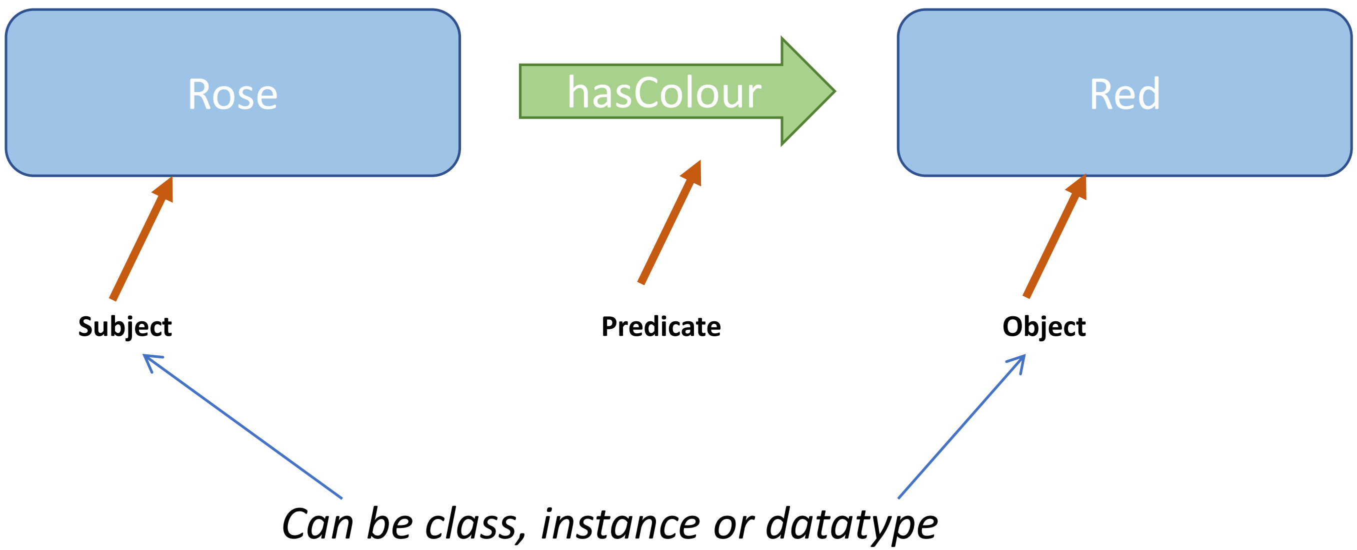

**FIGURE 3.6**

Combined graph of all triples about Shakespeare and the United Kingdom.

# Adding more expressivity (RDFS)

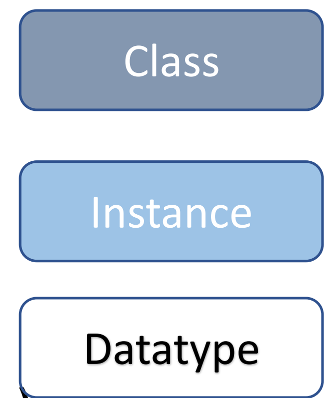
# Triples can be more specialised

- Example



# Subject–Predicate–Object expressions

- Previous example links 2 instances (Rose and Red)
- The subject and object can be of three categories:
  - Class
  - Datatype (i.e.: string, Integer, Boolean)
  - Instance (of class)
- The predicate can be
  - Datatype property (linking 2 instances)
  - Object property (linking an instance and a datatype)
- Predicates can be user defined (e.g. “hasColour”) or predefined (e.g. RDF and RDFS predicates)

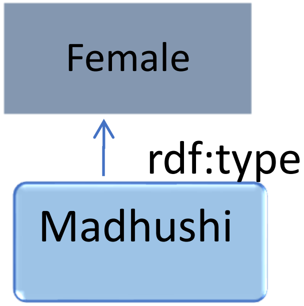


# Class Definition

- “A collection of individuals or sets of individuals that can be defined by their common properties”
  - Open World Assumption
    - A Class “Y” is the set of things that:
      - Have some common property(ies) - Intentional
      - Are designated to be a member of the Class – Extensional

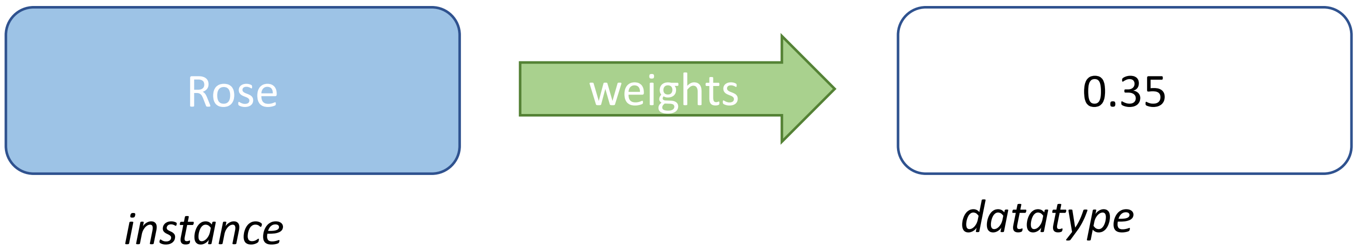
- Classes can be arranged into hierarchies
- An instance of a class or subclass is a *member or individual*
- Relationships between classes and instances are

defined with RDF predicate “type”



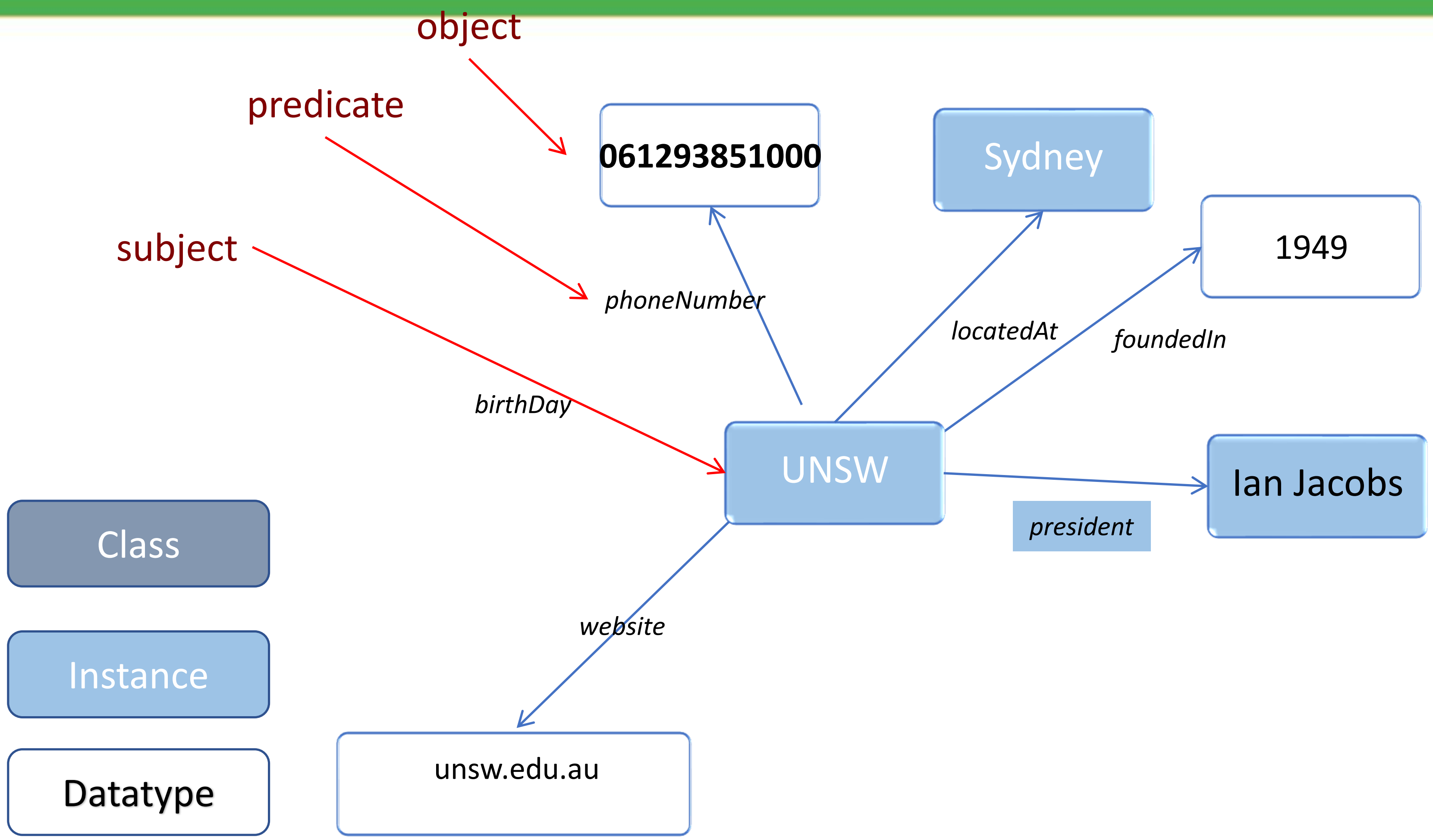
# Properties

## Datatype property



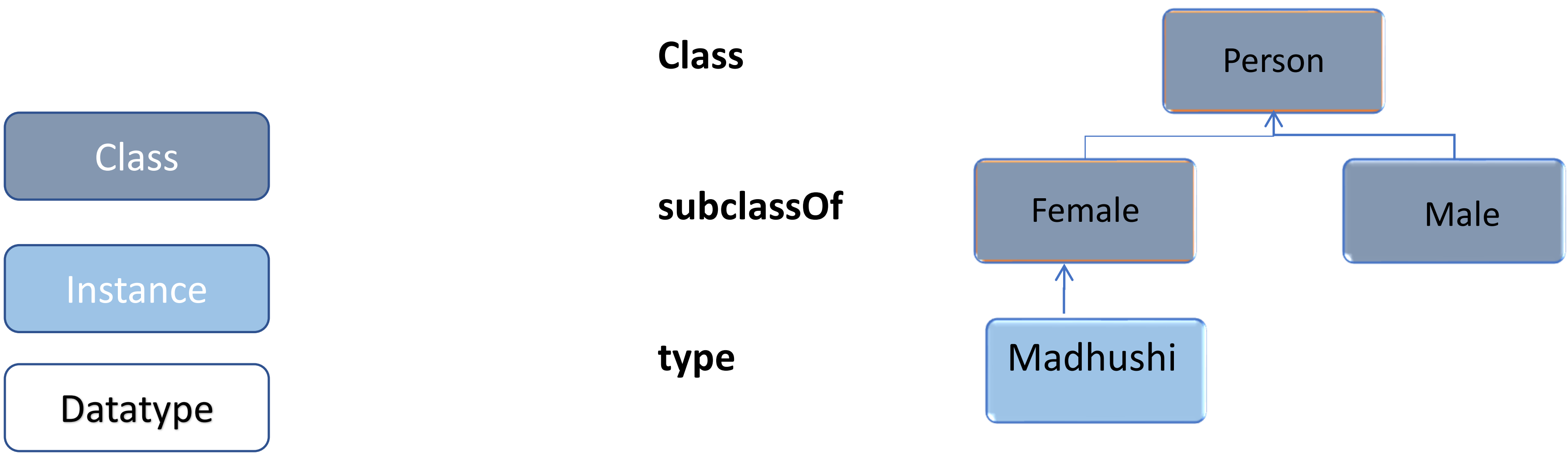

## Object property

# Example Graph

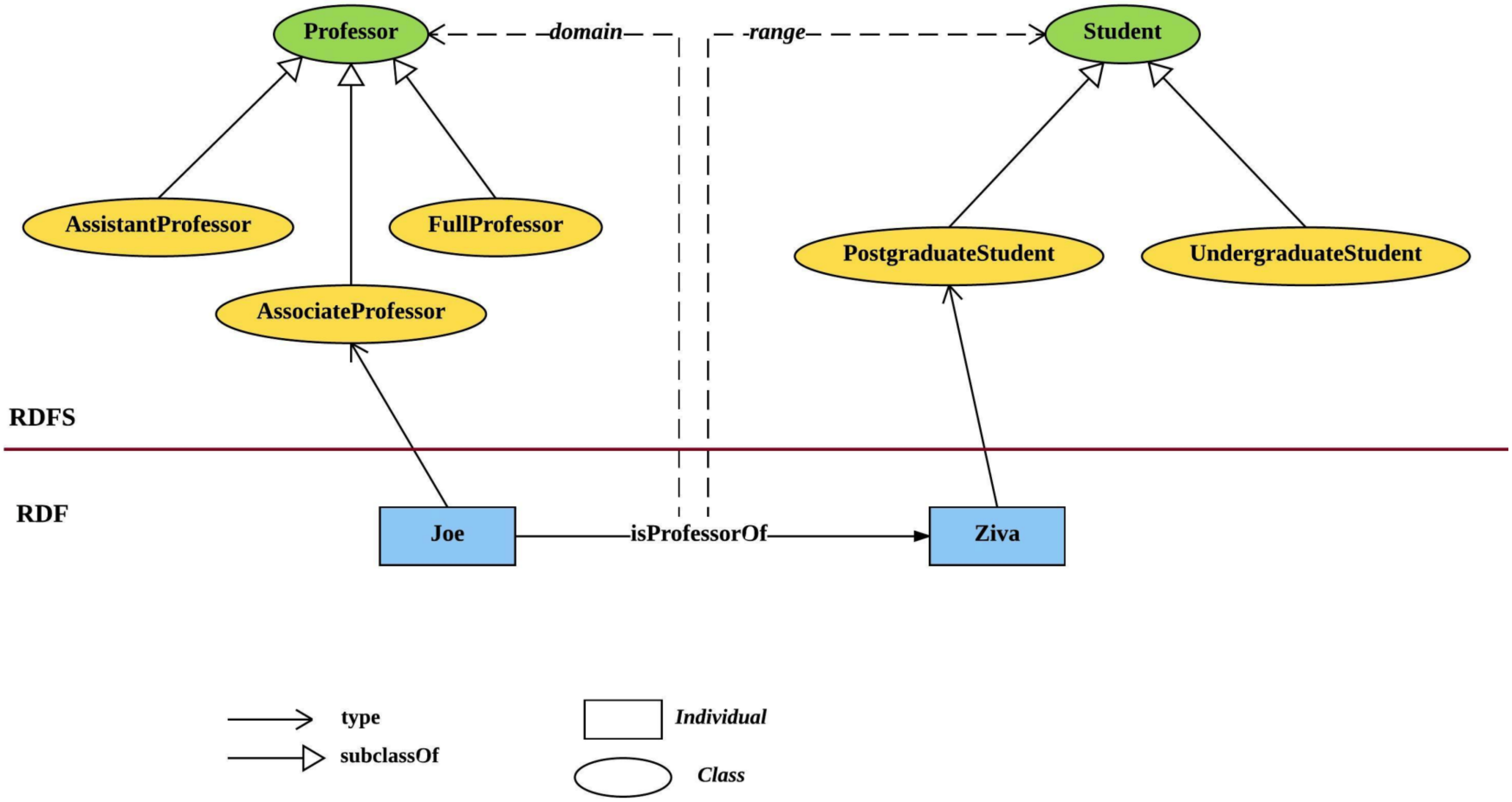


# Subclasses

- <u>Definition:</u> A classification schema for a categorisation of the concepts in a domain in a hierarchical structure
- Uses the subClass relationship, (parent-child), defined in RDFS

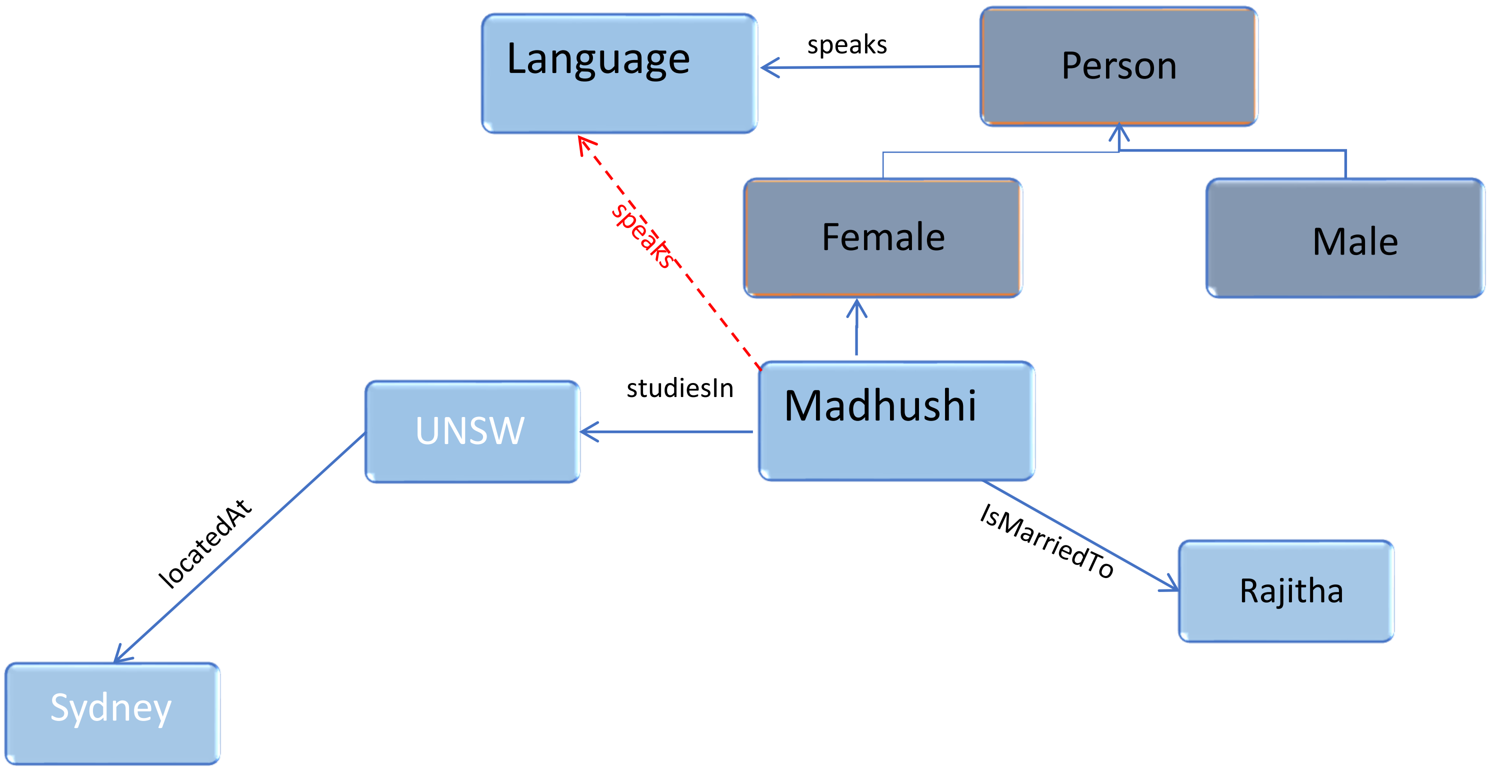


# Example


Professor
domain
range
Student
AssistantProfessor
FullProfessor
AssociateProfessor
PostgraduateStudent
UndergraduateStudent
RDFS
RDF
Joe
isProfessorOf
Ziva
type
subclassOf
Individual
Class

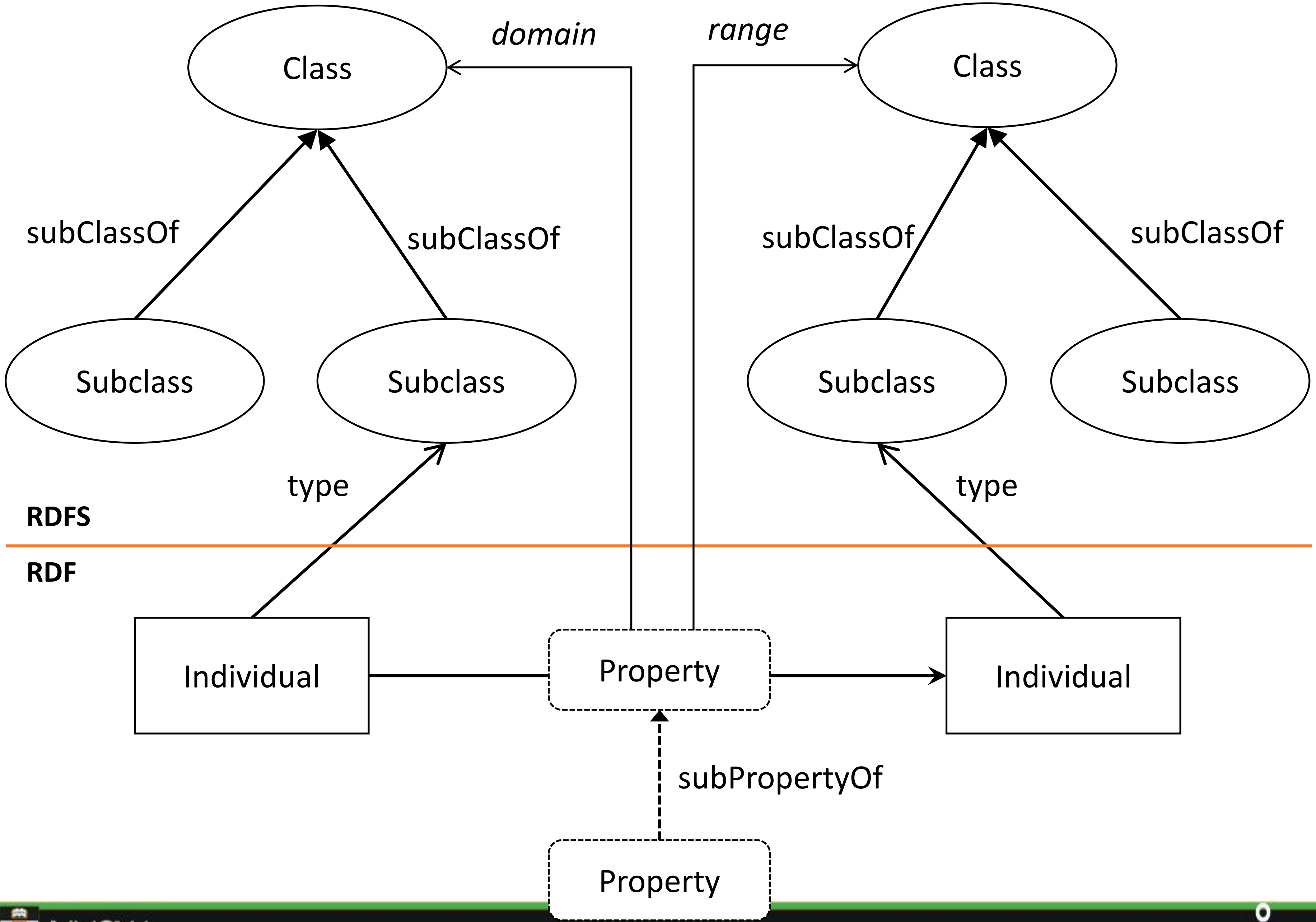
# Inheritance

- From the graph, we can see by inheritance that Madhushi speaks a Language

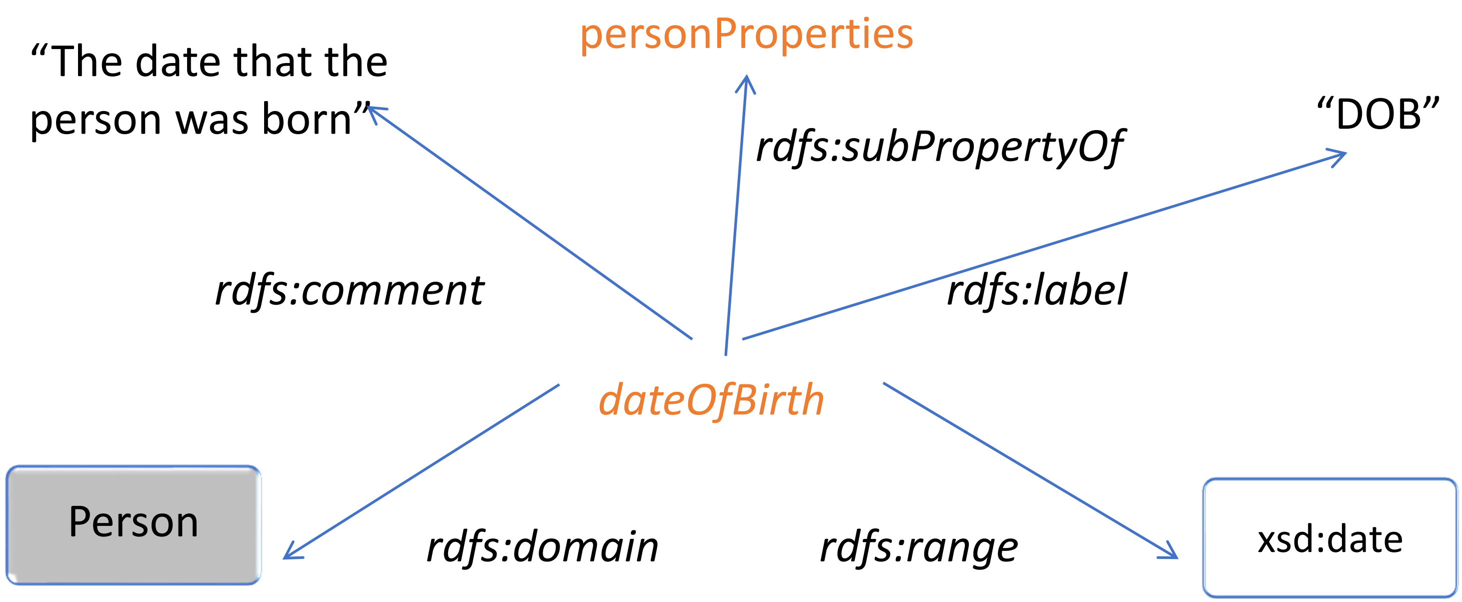


# Predefined predicates

- RDF defines:
  - rdf:type
- RDFS defines:
  - rdfs:class
  - rdfs:subClassOf
  - rdfs:domain
  - rdfs:range
  - rdfs:label
  - rdfs:comment
  - rdfs:subpropertyOf


domain
range
Class
Class
subClassOf
subClassOf
subClassOf
subClassOf
Subclass
Subclass
Subclass
Subclass
type
type
RDFS
RDF
Individual
Property
Individual
subPropertyOf
Property

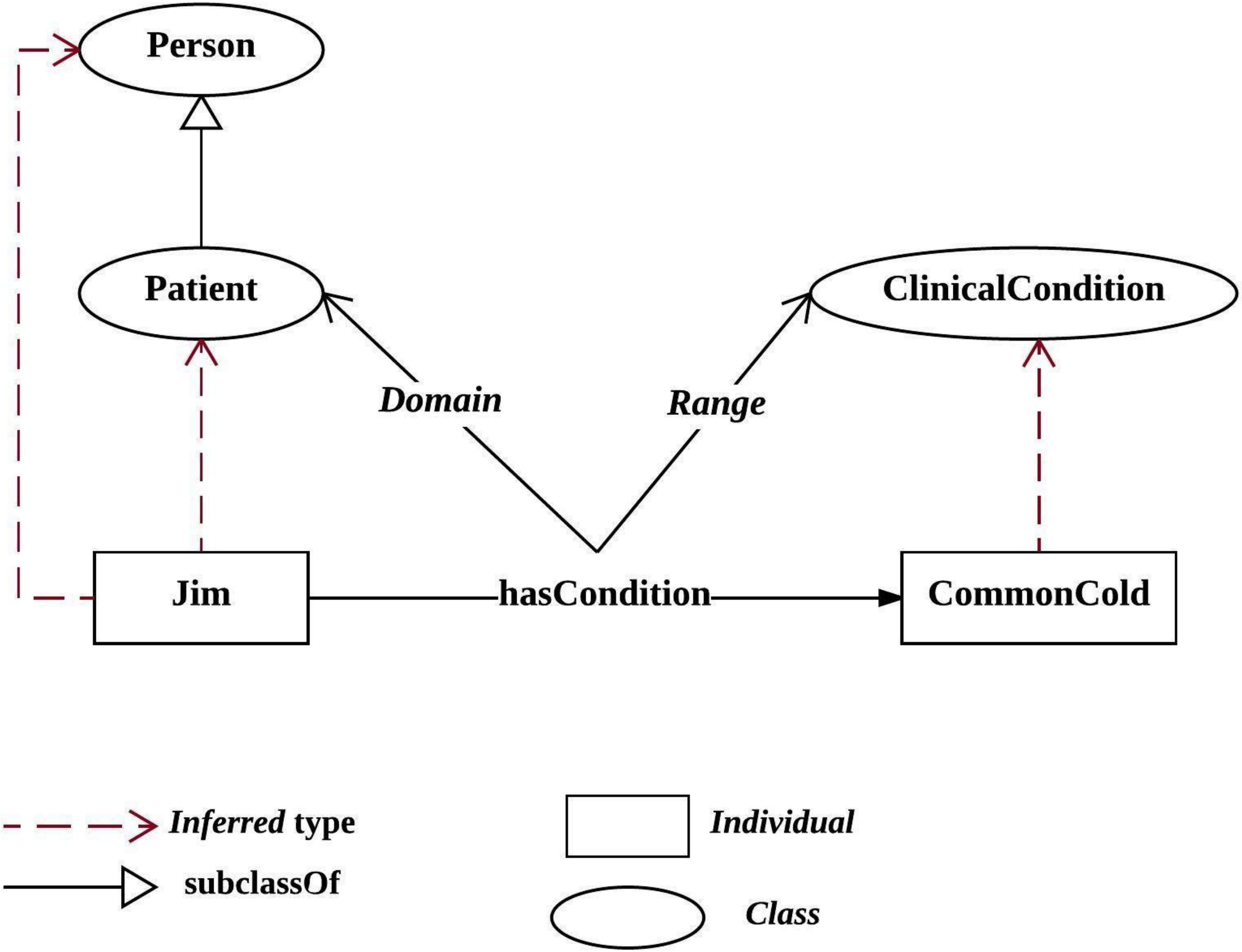
# Example of using predefined predicates




| Subject | Predicate | Object |
|---|---|---|
| dateOfBirth | *rdfs:domain* | Person |
| dateOfBirth | *rdfs:range* | xsd:date |
| dateOfBirth | *rdfs:label* | "DOB" |
| dateOfBirth | *rdfs:comment* | "The date…" |
| dateOfBirth | *rdfs:subPropertyOf* | personProperties |

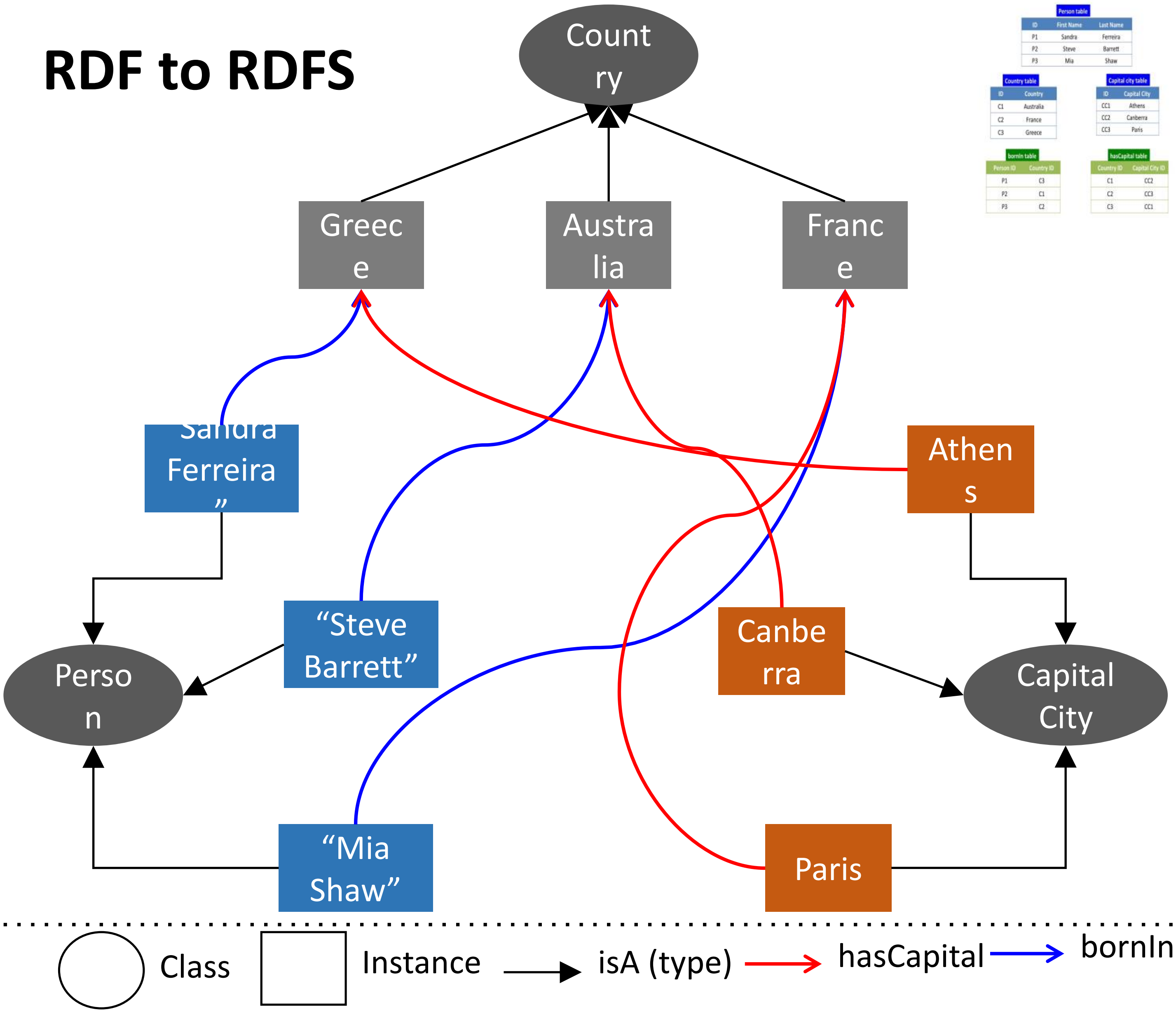
# Comparing RDF and RDFS

- Example:
  - RDF:
    - Jim has common cold
  - RDF(S):
    - `Patient` is a subclass of `Person`
    - The domain for the property `hasCondition` is the class `Patient` and range is the class `ClinicalCondition`



Person
Patient
ClinicalCondition
Domain
Range
Jim
hasCondition
CommonCold
Inferred type
subclassOf
Individual
Class

# RDF to RDFS




Person table

| ID | First Name | Last Name |
|---|---|---|
| P1 | Sandra | Ferreira |
| P2 | Steve | Barrett |
| P3 | Mia | Shaw |

Country table

| ID | Country |
|---|---|
| C1 | Australia |
| C2 | France |
| C3 | Greece |

Capital city table

| ID | Capital City |
|---|---|
| CC1 | Athens |
| CC2 | Canberra |
| CC3 | Paris |

bornIn table

| Person ID | Country ID |
|---|---|
| P1 | C3 |
| P2 | C1 |
| P3 | C2 |

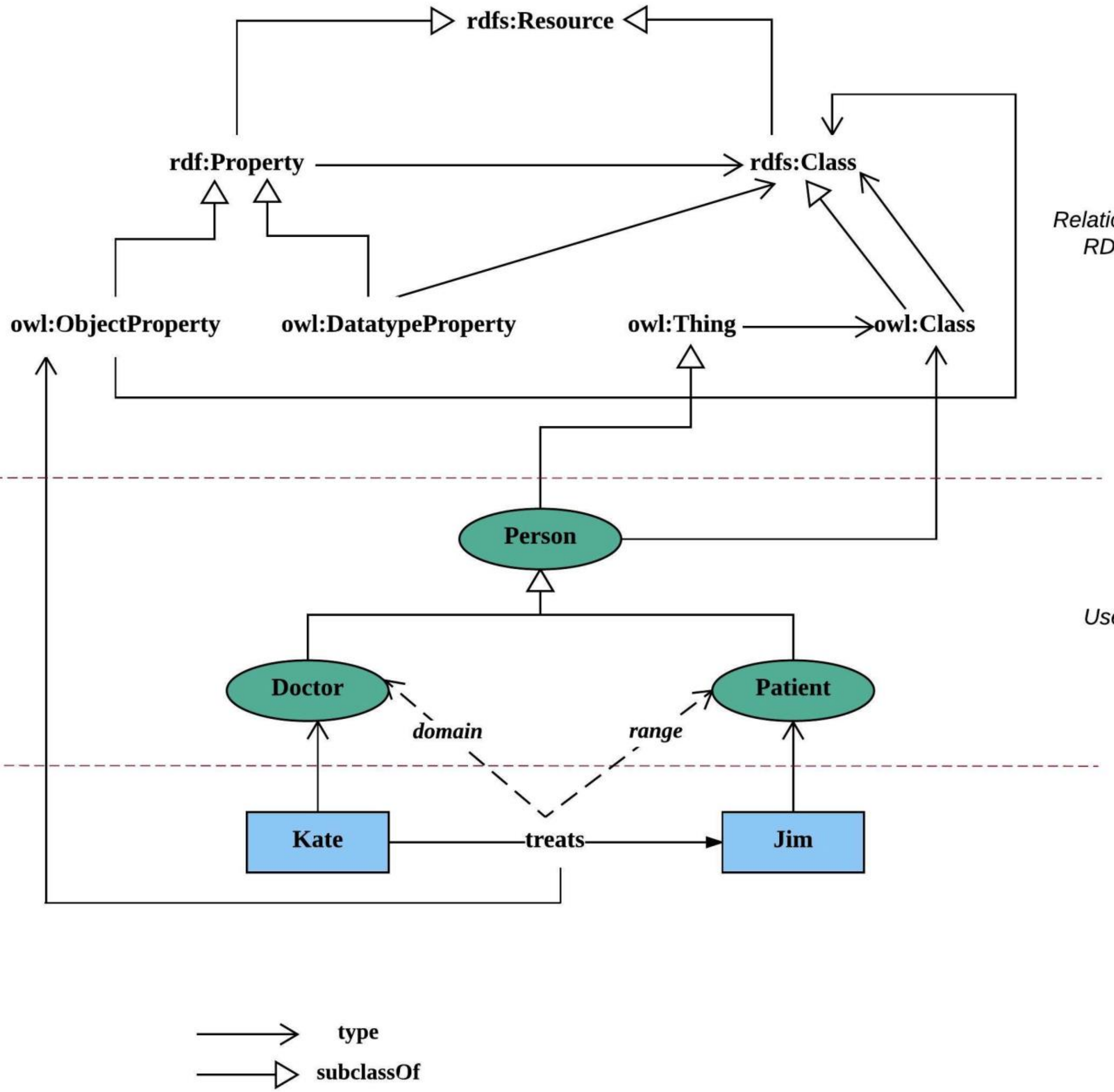
hasCapital table

| Country ID | Capital City ID |
|---|---|
| C1 | CC2 |
| C2 | CC3 |
| C3 | CC1 |

# More expressivity using OWL

# OWL

- Web **Ontology** Language (**OWL**)
- Designed for expressivity about
  - Things
  - Groups of things
  - Relations between things
- Designed to facilitate making *inferences*



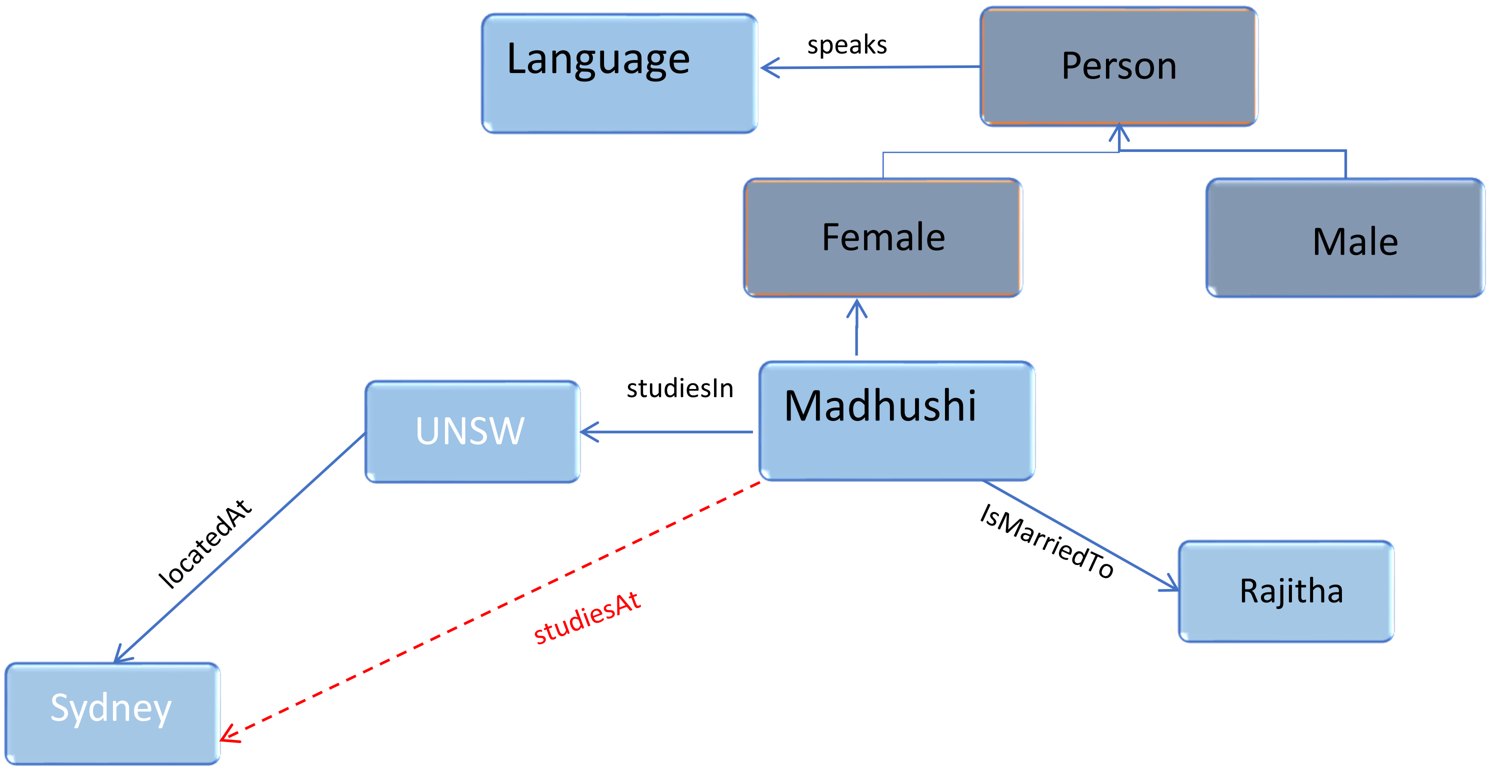

*Relation between RDF, RDFS, and OWL*

*User defined classes*

*Instances*

# Inference example

- From the graph, we can see by inference that Madhushi studies at Sydney



# OWL basics (1)

- Class:
  - owl:Class*
  - owl:Thing
  - owl:Nothing
  - owl:Restriction
  - owl:onProperty
- Property:
  - owl:ObjectProperty
  - owl:DatatypeProperty
- Equivalence:
  - owl:sameAs
  - owl:equivalentClass
  - owl:equivalentProperty
- Non-equivalence:
  - owl:differentFrom
  - owl:disjointWith
  - owl:AllDifferent
  - owl:distinctMembers

# OWL basics (2)

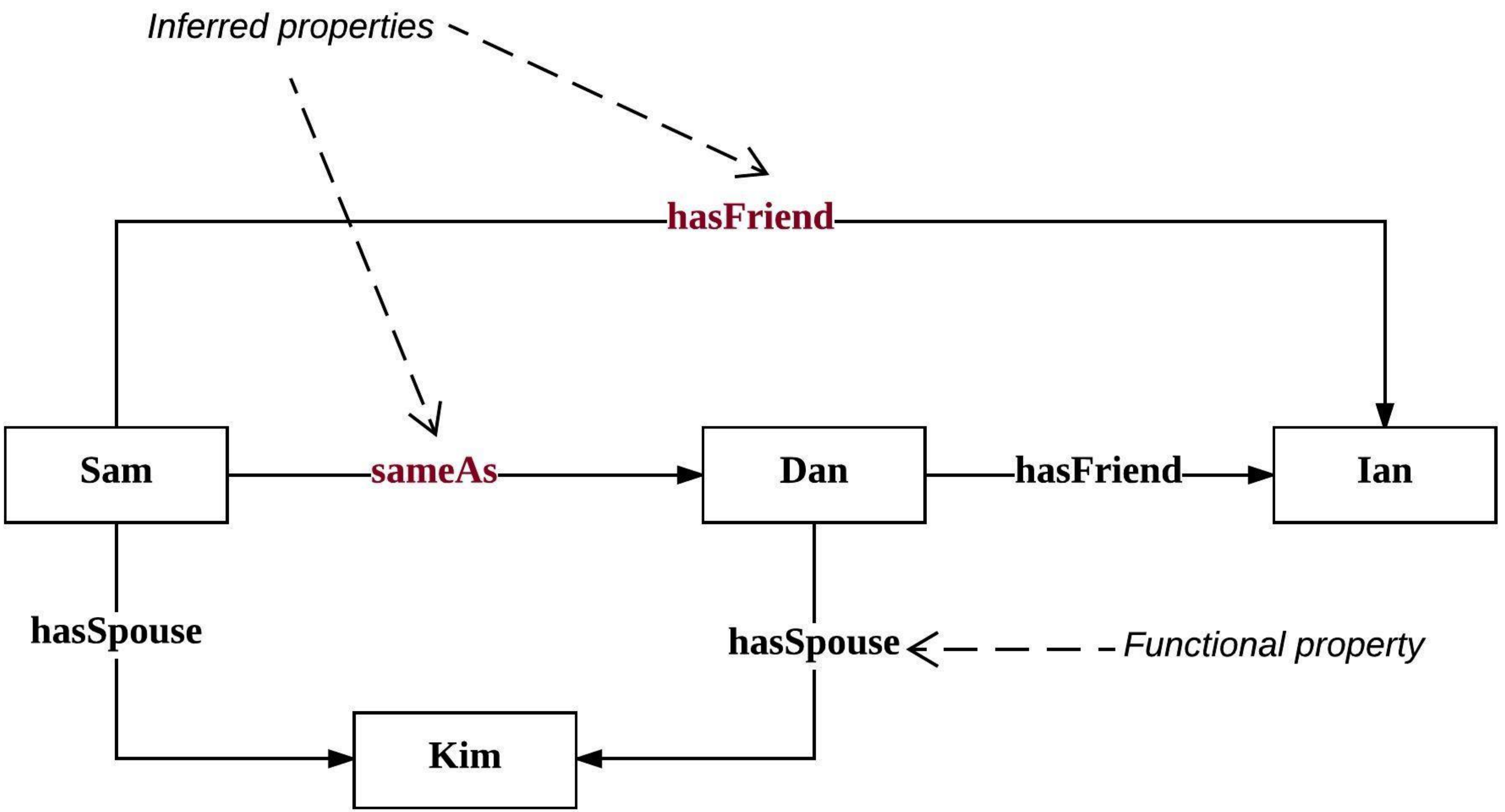
- Qualification:
  - owl:hasValue
  - owl:someValuesFrom
  - owl:allValuesFrom
- Cardinality:
  - owl:minCardinality
  - owl:maxCardinality
  - owl:cardinality
- Enumeration
  - owl:oneOf
- Boolean:
  - owl:complementOf
  - owl:unionOf
  - owl:intersectionOf

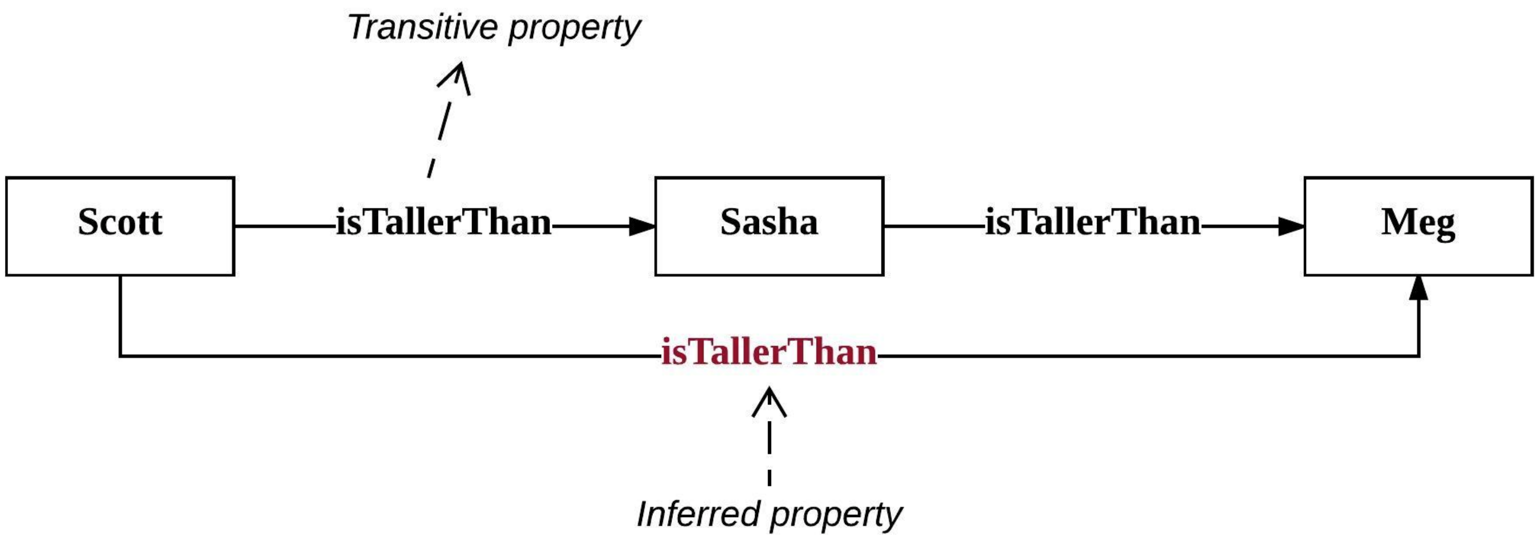
# OWL properties

- OWL properties* and corresponding modelling rules:
  - owl:inverseOf
    - If ($p_1$, owl:inverseOf, $p_2$) and (x, $p_1$, y) Then (y, $p_2$, x)
    - Example: If (hasPatient, owl:inverseOf, isPatientOf) and (Kate, hasPatient, Jim) Then (Jim, isPatientOf, Kate)
  - owl:TransitiveProperty
    - If (p, rdf:type, owl:TransitiveProperty) and (x, p, y) and (y, p, z) Then (x, p, z)
    - Example: If (isTallerThan, rdf:type, owl:TranisitveProperty) and (Scott, isTallerThan, Sasha) and (Sasha, isTallerThan, Meg) Then (Scott, isTallerThan, Meg)
  - owl:SymmetricProperty
    - If (p, rdf:type, owl:SymmetricProperty) and (x, p, y) Then (y, p, x)
    - Example: If (hasFriend, rdf:type, owl:SymmetricProperty) and (Dan, hasFriend, Ian) Then (Ian, hasFriend, Dan)

# OWL properties

- Continued..
  - owl:FunctionalProperty
    - If (p, rdf:type, owl:FunctionalProperty) and (x, p, y) and (x, p, z) Then (y, owl:sameAs, z)
    - Example: If (hasSpouse, rdf:type, owl:FunctionalProperty) and (Kim, hasSpouse, Sam) and (Kim, hasSpouse, Dan) Then (Sam, owl:sameAs, Dan)
  - owl:InverseFunctionalProperty
    - If (p, rdf:type, owl:InverseFunctionalProperty) and (x, p, y) and (z, p, y) Then (x, owl:sameAs z)
    - Example: If (hasPassportNumber, owl:InverseFunctionalProperty) and (Toni, hasPassportNumber, "A5110817") and (Sophie, hasPassportNumber, "A5110817") Then (Toni, owl:sameAs, Sophie)

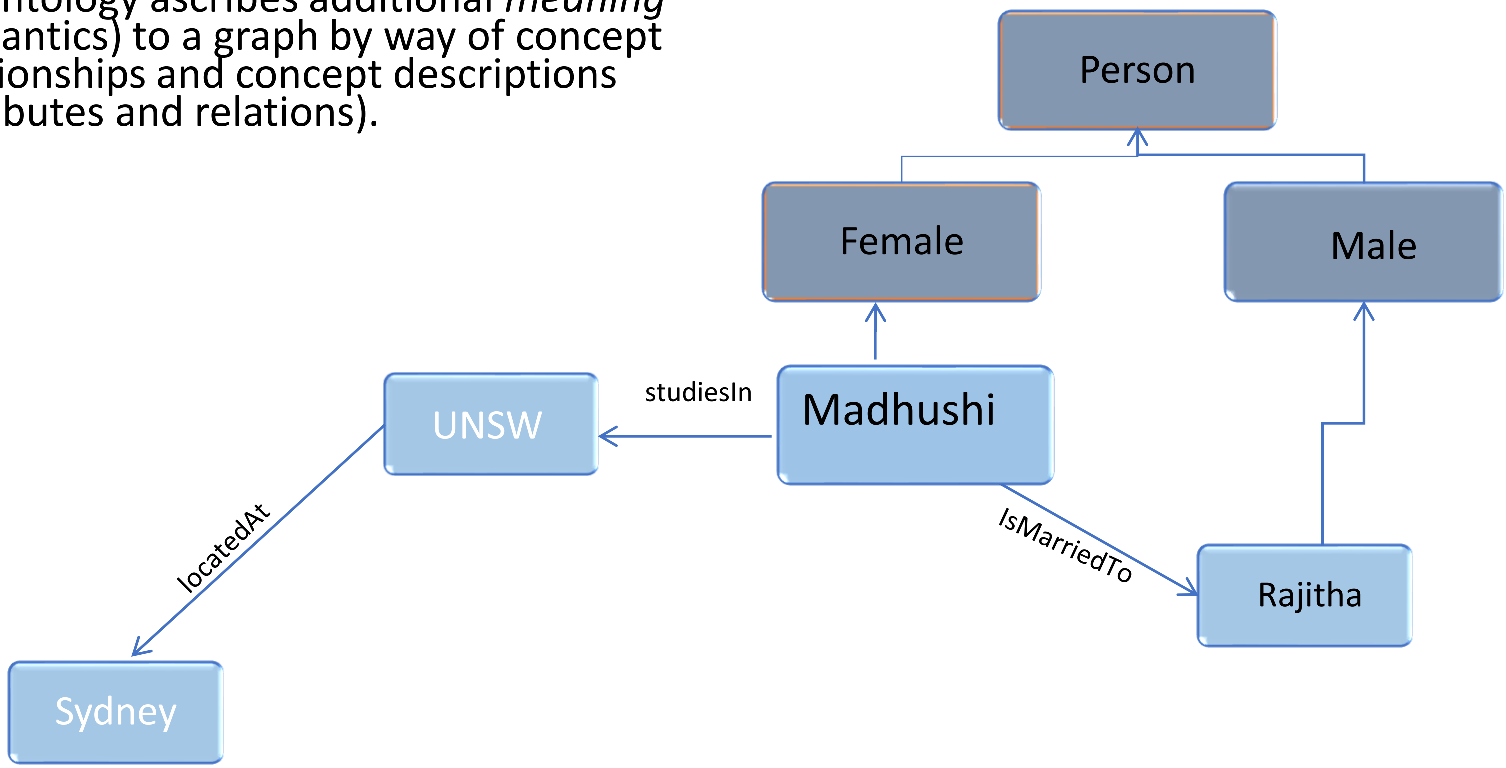

Inferred properties
hasFriend
Sam
sameAs
Dan
hasFriend
Ian
hasSpouse
hasSpouse
Functional property
Kim


Transitive property
Scott
isTallerThan
Sasha
isTallerThan
Meg
isTallerThan
Inferred property

# Summary

# Ontologies

- <u>Definition:</u> *"A formal, explicit specification of a shared conceptualisation"*
- An Ontology ascribes additional *meaning* (semantics) to a graph by way of concept relationships and concept descriptions (attributes and relations).




[1] Gruber, 1995

# Semantic modelling standards

- RDFS and OWL define special types of relationships on top of RDF
- RDFS: Define class hierarchies
- OWL: Defines different types of properties
- All knowledge can be uniquely *encoded* in an ontology
- Querying and reasoning becomes possible

# Inheritance and Inference

- Ontologies contain embedded knowledge about an entity through **Inheritance** and **Inference**
- Inheritance
  - In the structure of parent-child relationships, a subtype inherits the properties and relations of a supertype and increments one or more additional properties
    - If B is a *subClassOf* C and x is a member of B
    - By inference, we can derive that x is also a member of C
- Inference
  - Extends the concept of inheritance to all relationships of an entity (not just super/subtypes)

# Conclusion

- The basic element in a semantic model is a triple which denotes the relationship between a subject, a predicate and an object
- Simple structures can be defined on top of that
  - Subjects and objects can be classes or instances
  - Predicates connecting different instances can be properties
  - Predicates connecting different classes can be of different types, most commonly in a class and subclass relationship
  - Additional and more complex relationships can be defined
- These simple structures are used together to build ontologies which represent knowledge in a specific area
- One characteristic of semantic models is that new knowledge can inferred from the ontology