COMP318
Ontologies and Semantic Web
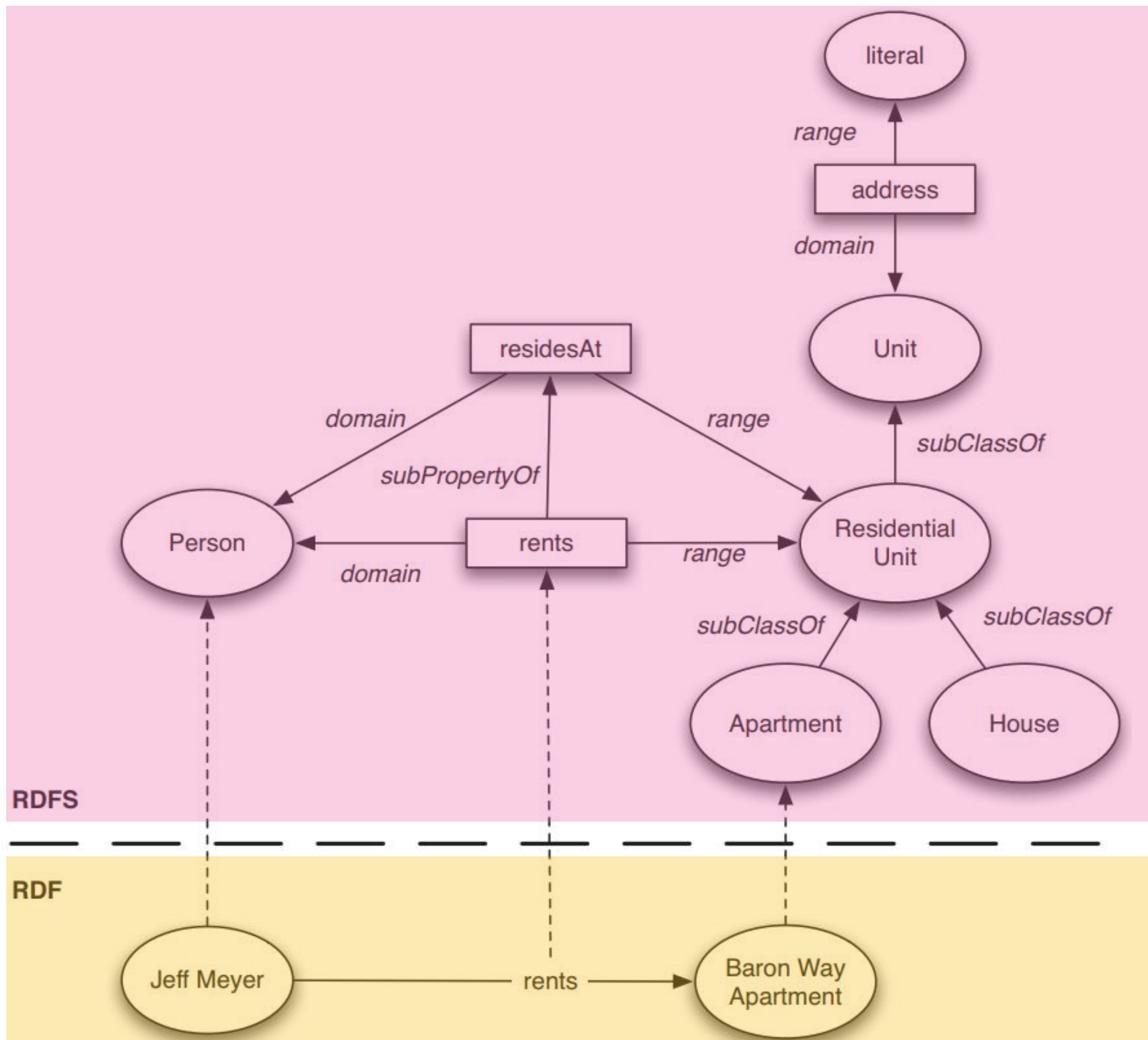
RDF - Part 8

Dr Valentina Tamma
V.Tamma@liverpool.ac.uk

How the Word Cloud Generator Works

The layout algorithm for positioning words without overlap is available on GitHub under an open source license as d3-cloud. Note that this is the only the layout algorithm and any other for converting text into words and rendering the final output requires additional development.

As word placement can be quite slow for more than a few hundred words, the layout algorithm can be run asynchronously, with a configurable time step size. This makes it possible to animate words as they are placed without stuttering. It is recommended to always use a time step even without animations as it prevents the browser's event loop from blocking while placing the words.

The layout algorithm itself is incredibly simple. For each word, starting with the most "important":

Where were we

- RDF data model

- Collections, containers, classes

RDFS Vocabulary

- **RDFS is the RDF vocabulary description language**

  - it can be used to build simple RDF vocabularies

  - it provides a data model for describing groups of related resources, and their relationships.

    - RDFS inherits RDF syntax, and thus RDFS specifications are RDF data.

    - RDFS has a simple model theoretic semantics that allows inference in the form of entailment rules.

  - RDFS vocabulary is defined in the namespace:

    - http://www.w3.org/2000/01/rdf-schema#

RDFS Primitives

- **Resource**:
  - All resources are implicitly instances of `rdfs:Resource`.

- **Class**: describe sets of resources
  - classes are resources themselves
    - e.g. Webpages, people, document types
  - Class hierarchy can be defined through `rdfs:subClassOf`
  - Every class is a member of `rdfs:Class`

- **Property**: subset of RDFS Resources that are properties
  - **domain**: class associated with property, `rdfs:domain`
  - **range**: type of the property values, `rdfs:range`
  - Property hierarchy defined through `rdfs:subPropertyOf`

- **Statements**
  - Resources that reify subject/predicate/object triples

RDFS Vocabulary Description Language

- Classes:
  - `swp:Apartment rdf:type rdfs:Class`

- Class hierarchies:
  - `swp:Apartment rdfs:subClassOf swp:ResidentialUnit`
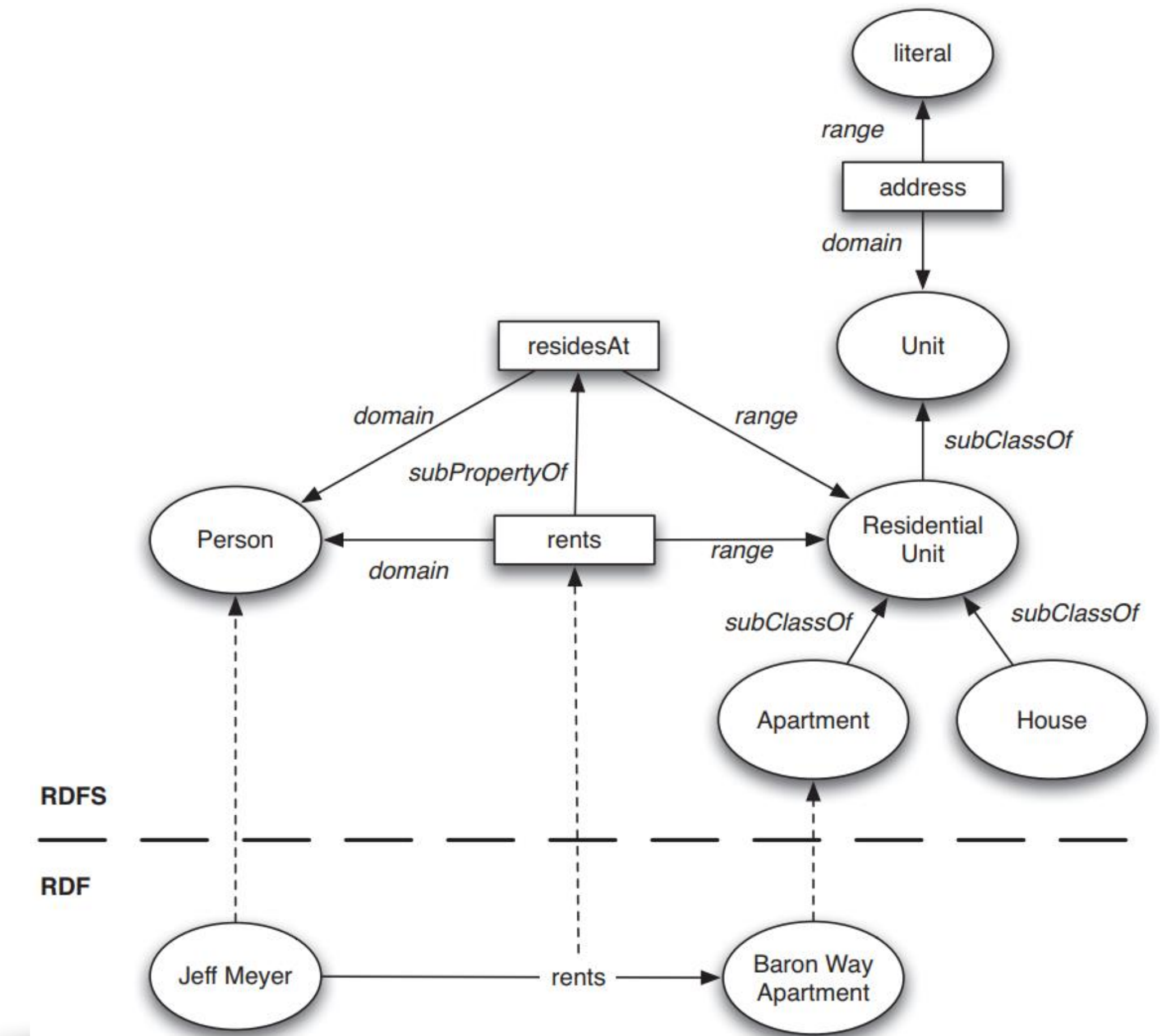
- Properties:
  - `swp:rents rdf:type rdf:Property`

- Property hierarchies:
  - `swp:rents rdfs:subPropertyOf swp:residesAt`

- Associating properties with classes:
  - `swp:address rdfs:domain swp:Unit`
    - "The property #hasName only applies to # Person:"
  - `swp:address rdfs:range xsd:string`
    - "The type of the property #hasName is # xsd:string:"

RDFS

- •RDFS Classes
  - rdfs:Resource
  - rdfs:Class
  - rdfs:Literal
  - rdfs:Datatype
  - rdfs:Container
  - rdfs:Container Membership Property

```xml
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:owl="http://www.w3.org/2002/07/owl#"
  xmlns:dc="http://purl.org/dc/elements/1.1/">

<owl:Ontology
   rdf:about="http://www.w3.org/2000/01/rdf-schema#"
   dc:title="The RDF Schema vocabulary (RDFS)"/>

<rdfs:Class rdf:about="http://www.w3.org/2000/01/rdf-schema#Resource">
  <rdfs:isDefinedBy rdf:resource="http://www.w3.org/2000/01/rdf-schema#"/>
  <rdfs:label>Resource</rdfs:label>
  <rdfs:comment>The class resource, everything.</rdfs:comment>
</rdfs:Class>

<rdfs:Class rdf:about="http://www.w3.org/2000/01/rdf-schema#Class">
  <rdfs:isDefinedBy rdf:resource="http://www.w3.org/2000/01/rdf-schema#"/>
  <rdfs:label>Class</rdfs:label>
```

RDFS

- •RDFS Properties
  - • rdfs:domain
  - • rdfs:range
  - • rdfs:subPropertyOf
  - • rdfs:subClassOf
  - • rdfs:member
  - • rdfs:seeAlso
  - • rdfs:isDefinedBy
  - • rdfs:comment
  - • rdfs:label

```
 <rdf:Property rdf:about="http://www.w3.org/2000/01/rdf-
schema#subClassOf">
 <rdfs:isDefinedBy rdf:resource="http://www.w3.org/2000/01/rdf-
schema#"/>
 <rdfs:label>subClassOf</rdfs:label>
 <rdfs:comment>The subject is a subclass of a class.</rdfs:comment>
 <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-
schema#Class"/>
 <rdfs:domain rdf:resource="http://www.w3.org/2000/01/rdf-
schema#Class"/>
</rdf:Property>

<rdf:Property rdf:about="http://www.w3.org/2000/01/rdf-
schema#subPropertyOf">
 <rdfs:isDefinedBy rdf:resource="http://www.w3.org/2000/01/rdf-
schema#"/>
 <rdfs:label>subPropertyOf</rdfs:label>
 <rdfs:comment>The subject is a subproperty of a property.</
rdfs:comment>
 <rdfs:range rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-
ns#Property"/>
 <rdfs:domain rdf:resource="http://www.w3.org/1999/02/22-rdf-
syntax-ns#Property"/>
</rdf:Property>
 ...
```

Why Classes are Useful

- Impose restrictions on what can be stated in an RDF document using the schema

  - As in programming languages

    - E.g. A+1, where A is an array

    - the arguments of "+" must be numbers

  - Disallow nonsense from being stated

Disallow nonsensical statements

- **`:JeffMeyer :rents :AmsterdamCo mmercialUnit`**

  - We want a person to only rent residential units

  - Restriction on values of the property "**`:rents`**" (range restriction)

    - Can only be a **`:ResidentialUnit`** or any of its subclasses

- **`:BaronWayBuilding :rents :BaronWayApartment`**

  - Only people can rent **`:ResidentialUnits`**

  - This imposes a restriction on the objects to which the property can be applied (domain restriction)
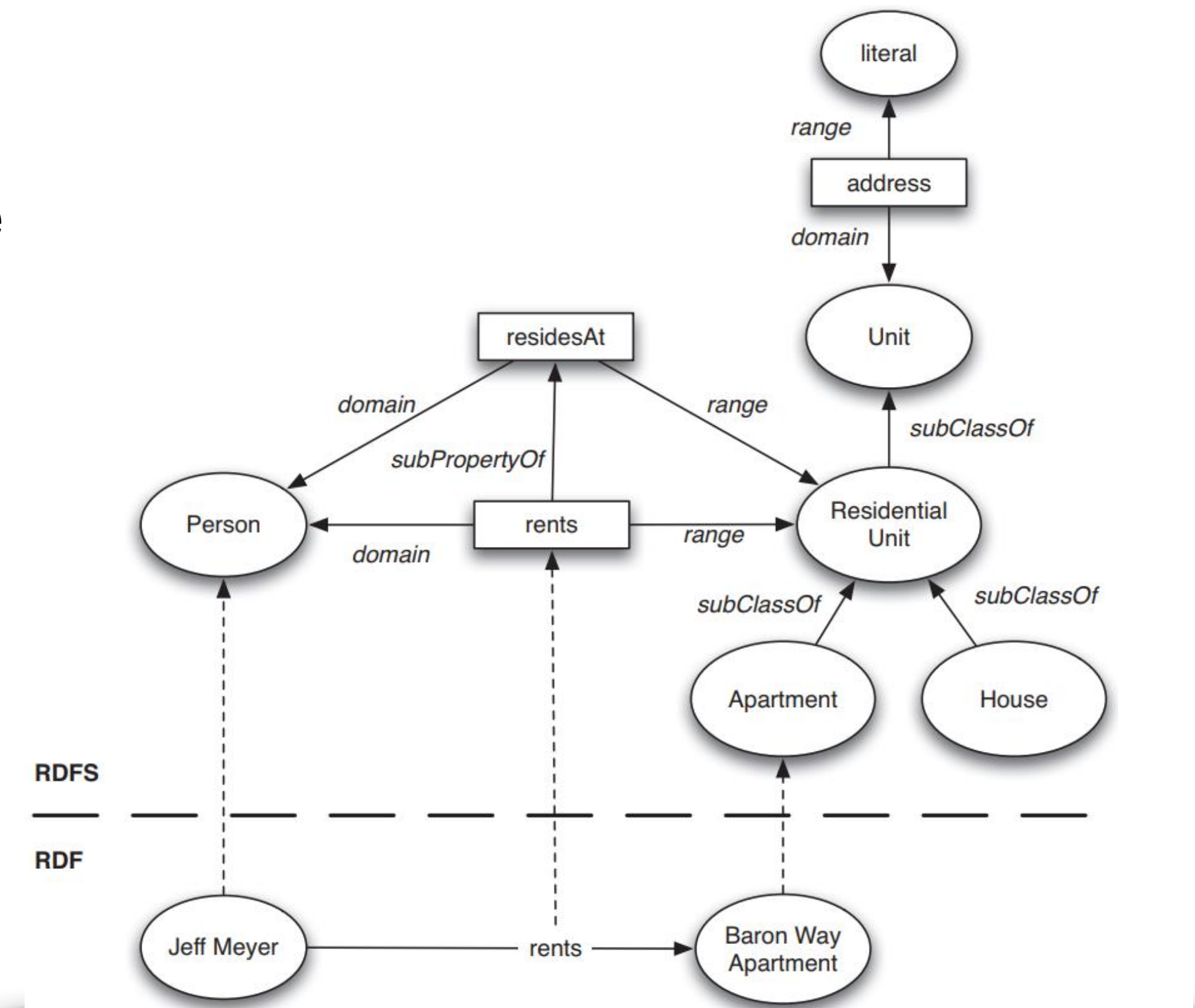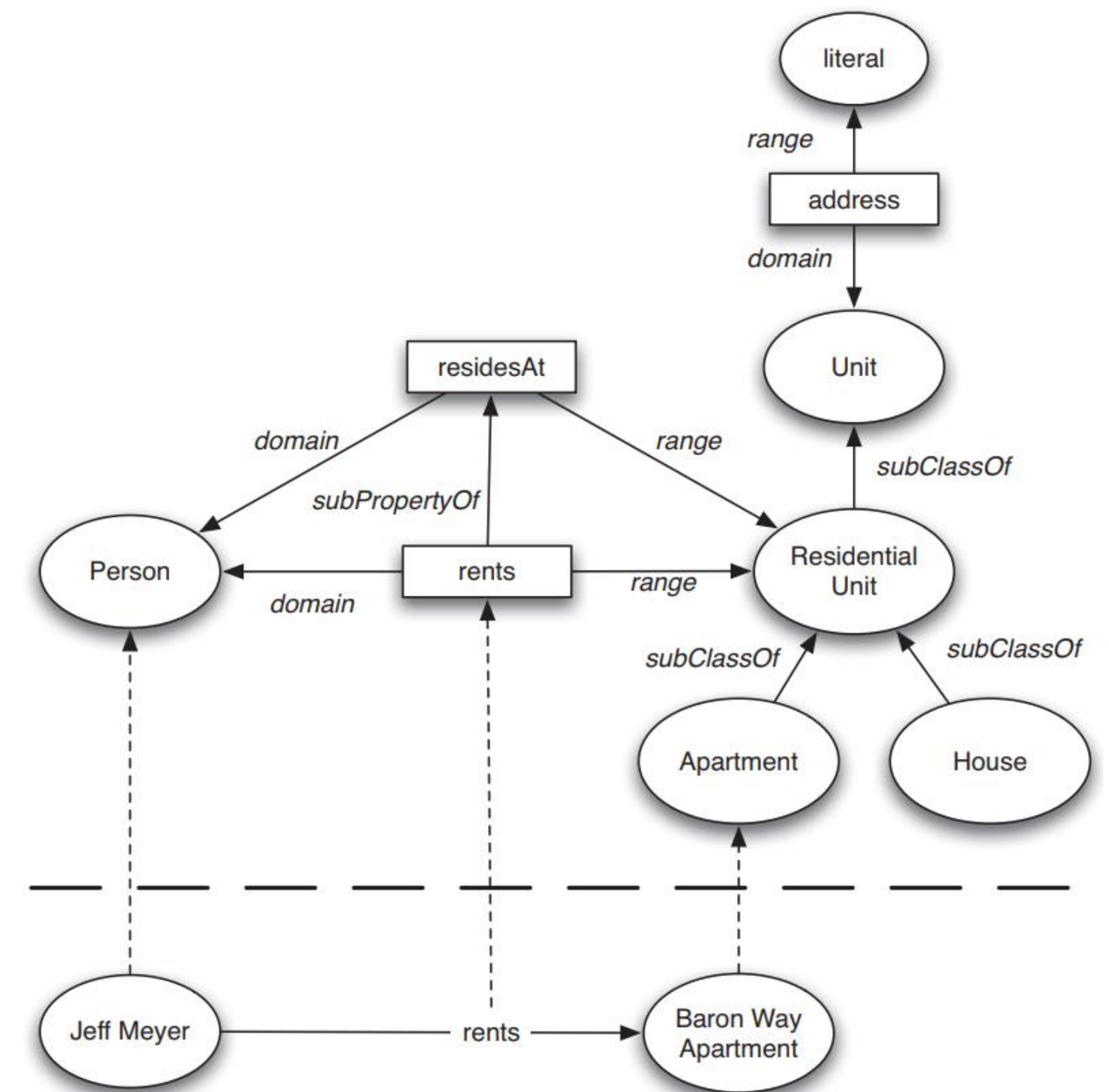
Class Hierarchies

- Classes can be organised in hierarchies

  - `:Apartment` is a subclass of `:ResidentialUnit` if every instance of `:Apartment` is also an instance of `:ResidentialUnit`

  - Then `:ResidentialUnit` is a superclass of `:Apartment`

- A subclass graph need not be a tree

- A class may have multiple superclasses

Class Hierarchy Example
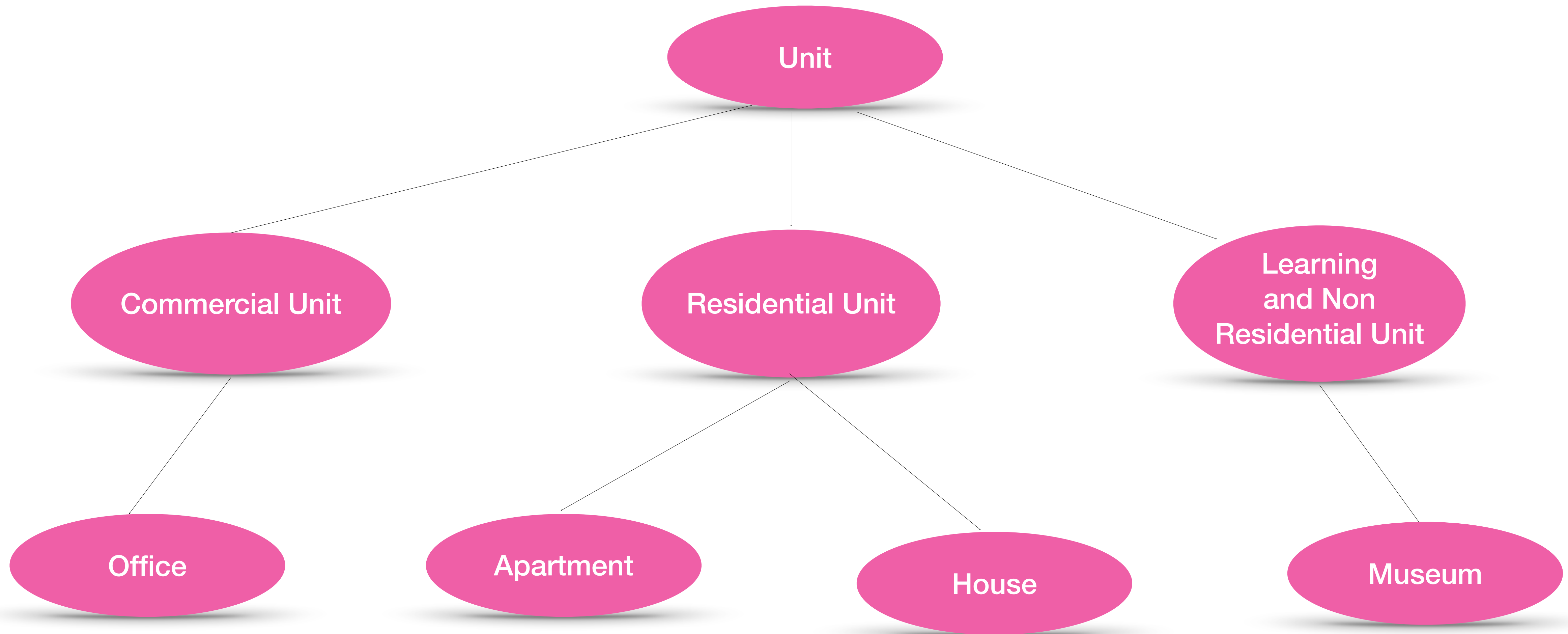
Inheritance in Class Hierarchies

- Range restriction: Apartments must have an address of type literal

  - `:BaronWayApartment` is an `:Apartment`

  - It inherits the property address from being a subclass of `:ResidentialUnit` and `:Unit`

- This is done in RDF Schema by fixing the semantics of "`rdfs:subClassOf`"

  - It is not up to an application (RDF processing software) to interpret "`rdfs:subClassOf`"

Property Hierarchies

- •Hierarchical relationships for properties
  - • E.g., **"rents"** is a sub-property of **"residesAt"**
  - • If a person *x* rents a residential unit *y* then the then the person *x* also resides at *y*

- •The converse is not necessarily true
  - • E.g., Someone might be a dependant of *x,* e.g.a child, but they do not rent *y*
    - • P is a subproperty of Q, if Q(x,y) is true whenever P(x,y) is true
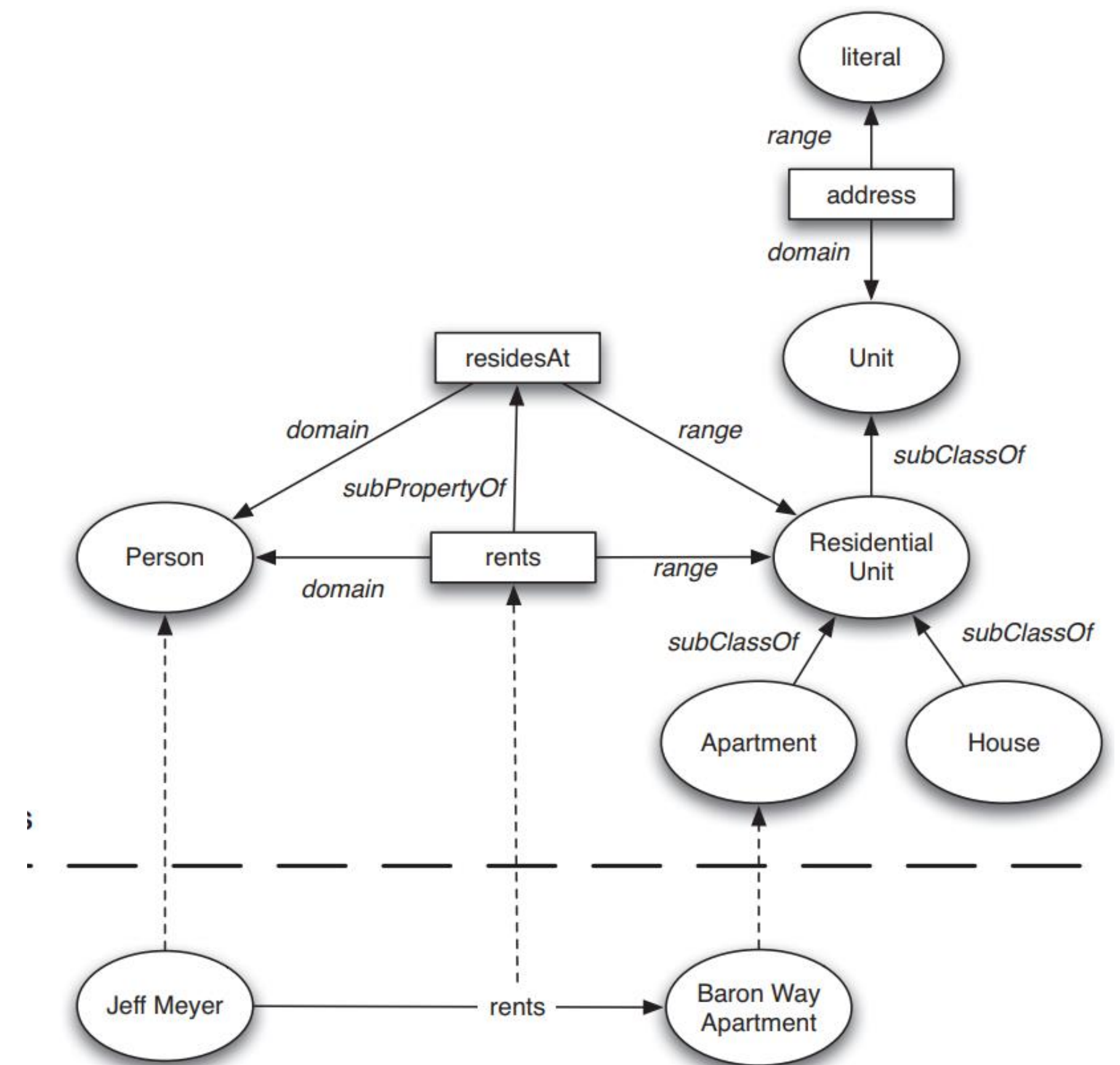
COMP318
Ontologies and Semantic Web

End of RDF - Part 8

Dr Valentina Tamma

`V.Tamma@liverpool.ac.uk`

How the Word Cloud Generator Works

The layout algorithm for positioning words without overlap is available on GitHub under an open source license as d3-cloud. Note that this is the the only the layout algorithm and any of for converting text into words and rendering the final output requires additional development.

As word placement can be quite slow for more than a few hundred words, the layout algorithm can be run asynchronously, with a configurable time step size. This makes it possible animate words as they are placed without stuttering. It is recommended to always use a time step even without animations as it prevents the browser's event loop from blocking wh placing the words.

The layout algorithm itself is incredibly simple. For each word, starting with the most "important":