RDF Schema

Mauricio Espinoza Mejía

mauricio.espinoza@ucuenca.edu.ec

RDF Schema: the language

Properties

- rdf:Property
- **A property can be defined by stating that it is an instance of the predefined class** rdf:Property.
- Example:

ex:author rdf:type rdf:Property .

- Then, property ex:author can be used as a **predicate in an RDF triple** such as the following:

ex:john ex:author ex:book123 .

RDF Schema: the language

Properties

- rdf:Property
- **Defining a property explicitly is optional**; if we write the RDF triple

S P O .

- then P is **inferred** to be a property by RDFS.
- Optionally, properties can be declared to apply to certain instances of classes by defining their **domain and range.**

RDF Schema: the language

Domain and Range: **Example**

ex:Book rdf:type rdfs:Class .
ex:Person rdf:type rdfs:Class .

ex:author rdf:type rdf:Property .

ex:author rdfs:domain ex:Book .
ex:author rdfs:range ex:Person .

RDF Schema: the language

Domain and Range:

- For a property, we can have **zero, one, or more than one** domain or range statements.

RDF Schema: the language

Domain and Range:

- **No domain or no range statement**: If no range statement has been made for property P, then **nothing has been said about the values** of this property. Similarly for no domain statement.

- **Example:** If we have only the triple

  ex:frank ex:hasMother ex:mary .

- then nothing can be inferred from it regarding resources ex:frank and ex:mary

RDF Schema: the language

Domain and Range:

- **One domain statement:** If we have

  P rdfs:domain D .

- then we can **infer** that when P is applied to some resource, this resource is an instance of class D.

- **One range statement:** If we have

  P rdfs:range R .

- then we can **infer** that when P is applied to some resource, the value of P is an instance of class R.

RDF Schema: the language

Domain and Range

- If we have

ex:author rdfs:domain ex:Book .
ex:book123 ex:author ex:frank .

then we can infer:

ex:book123 rdf:type ex:Book.

- If we have

ex:author rdfs:range ex:Person .
ex:book1 ex:author ex:frank .

then we can infer

ex:frank rdf:type ex:Person .

RDF Schema: the language

Domain and Range

- **Two domain or range statements:** If we have

  P rdfs:range C1 .

  P rdfs:range C2 .

- then we can **infer** that the values of property P are instances of both C1 and C2. Similarly, for two domain statements.
- **Example:** If we have

  ex:hasMother rdfs:range ex:Female .

  ex:hasMother rdfs:range ex:Person .

  ex:frank ex:hasMother ex:frances .

- then we can infer that ex:frances is an instance of both ex:Female and ex:Person

RDF Schema: the language

Domain and Range

- Another Example

```
ex:Professor rdf:type rdfs:Class .
ex:University rdf:type rdfs:Class .
ex:workAt rdf:type rdf:Property .
ex:workAt rdfs:domain ex:Professor .
ex:workAt rdfs:range ex:University .

ex:mauricio ex:workAt ex:ucuenca .
```

- What new triples can we infer from the above?

```
ex:mauricio rdf:type ex:Professor .
ex:ucuenca rdf:type ex:University .
```

RDF Schema: the language

Domain and Range

- Another Example

ex:Human rdf:type rdfs:Class .
ex:hasParent rdf:type rdf:Property .
ex:hasParent rdfs:domain ex:Human .
ex:hasParent rdfs:range ex:Human .

ex:tina ex:hasParent ex:john .

- What new triples can we infer from the above?

ex:tina rdf:type ex:Human .
ex:jhon rdf:type ex:Human .

RDF Schema: the language

Domain and Range: Datatypes for Ranges

- The rdfs:range property can also be used to indicate that the value of a property is given by a **typed literal.**
- **Example:**

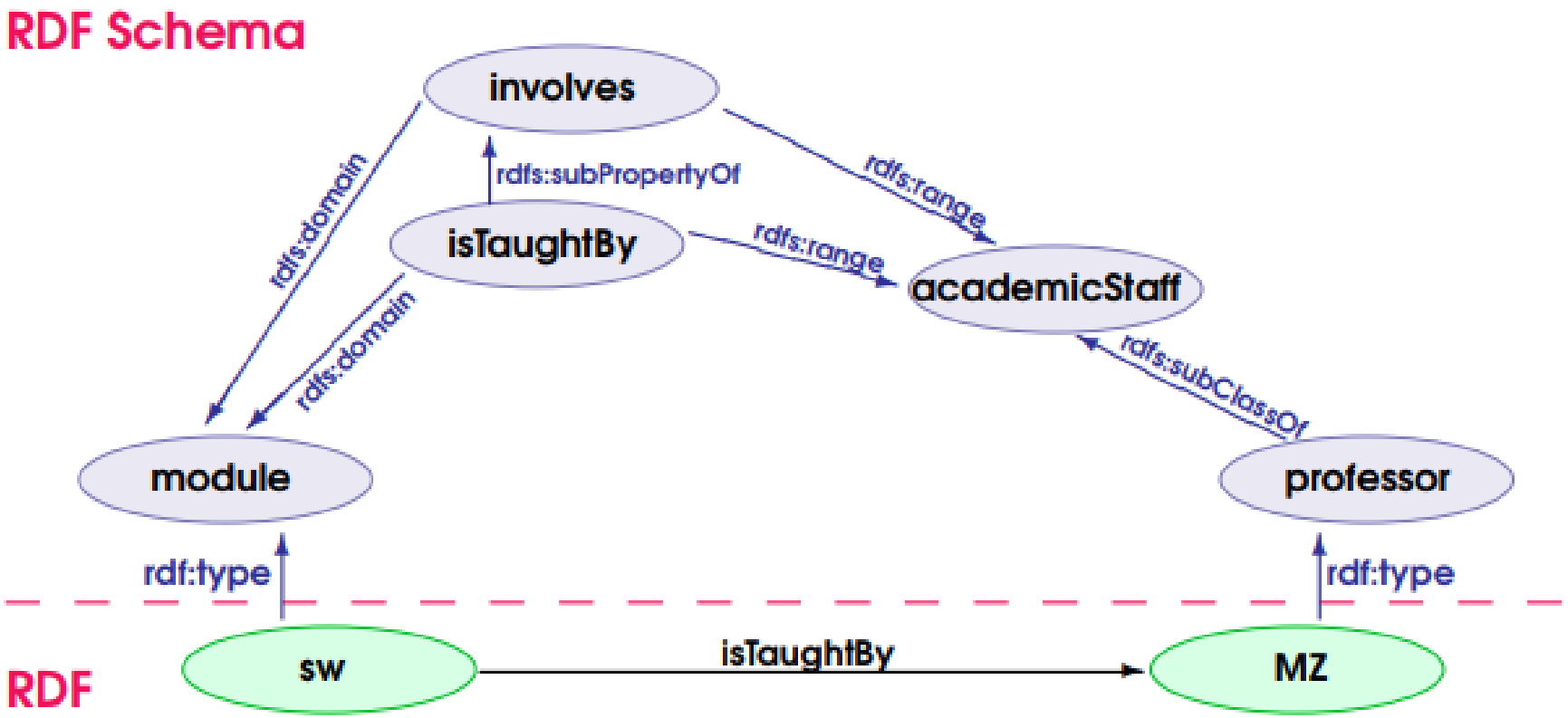
ex:age rdf:type rdf:Property .

ex:age rdfs:range xsd:integer .

Example

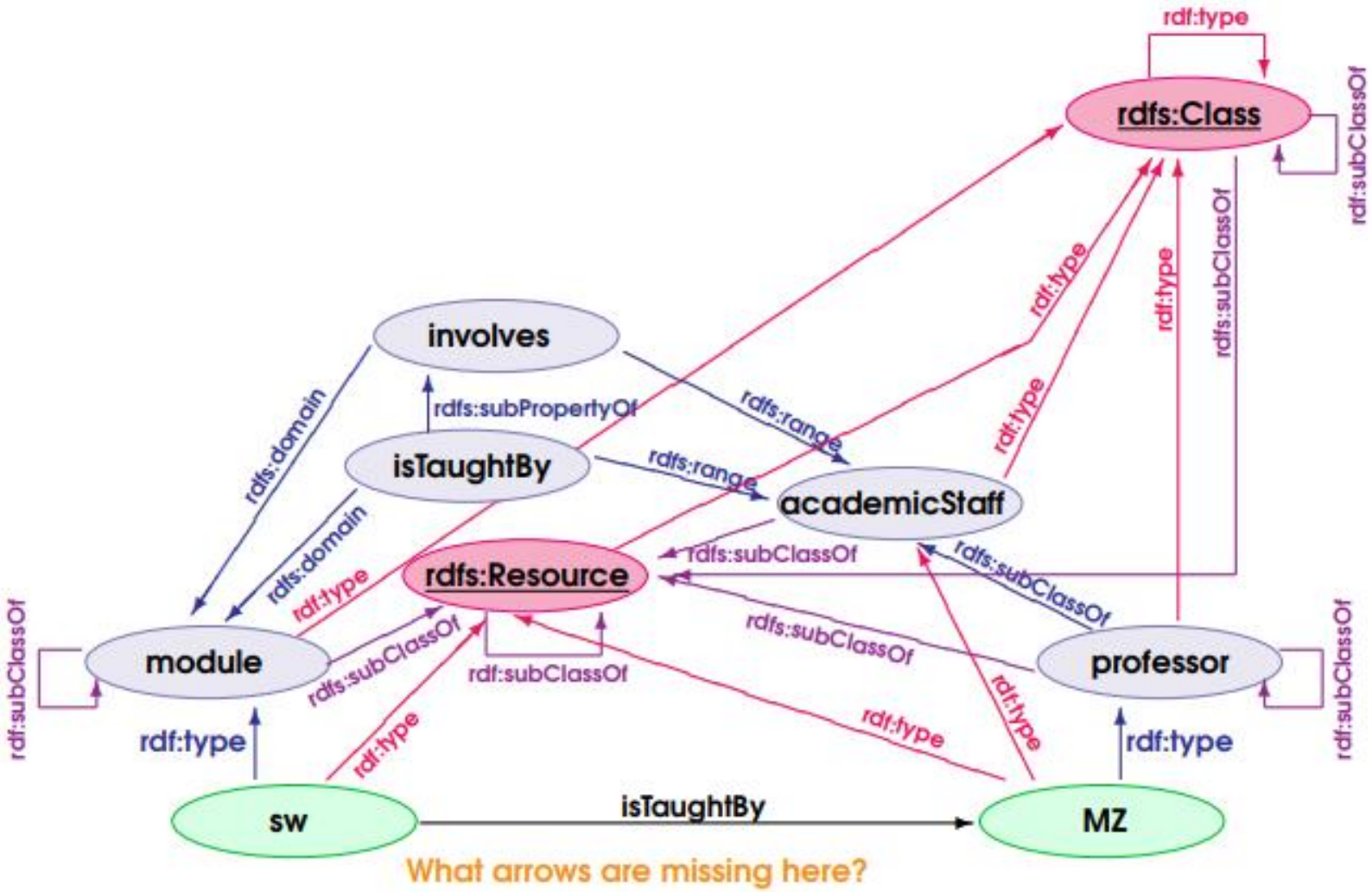
- professors are academic_staff_members
- modules are taught by academic staff members only
- Is taught by is a subproperty of involves
- Semantic Web is a module and Michael Zang is a proffesor

RDF vs. RDFS layers



What ‘implicit knowledge’ is missing in the picture

RDFS Semantics



Preguntas?