Semantic Web

The RDF Schema Standard

W3C Recommendation 25 February 2014

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

RDF Vocabulary Description Language 1.1: RDF Schema

Basic Ideas of RDF Schema

- RDF is a universal language that lets users describe resources in their own vocabularies
  - RDF does not assume, nor does it define semantics of any particular application domain
- The user can do so in RDF Schema using:
  - Classes and Properties
  - Class Hierarchies and Inheritance
  - Property Hierarchies

Semantic vs Object Oriented

- Like OO models, semantic models also use the term “Class” to group entities.
- In most OO systems, an object is an instance of a class, because it was built from the definition of that class. That is, the class is an object model.
- Semantic data is based on relationships between entities. Therefore, semantic models are oriented to property, rather than to the entity.

Semantic vs Object Oriented

- Semantic entities are not "born" from classes.
- Instead, they are perceived as members of a class, because of their properties.
- Because of this distinction, semantic models have much more flexibility.

Semantic vs Object Oriented

- In OO systems, properties are defined as part of the class. Knowing what class an object belongs to, you know what properties it has.
- In semantic systems, properties are defined independently.
- The property definition can, optionally, indicate to what kind of entities it is associated, as it can also indicate what kind of values it can take.

Classes and their Instances

- We must distinguish between:
  - Concrete “things” (individual objects) in the domain: Mathematics, John Seth, etc.

  and
  - Sets of individuals sharing common properties called classes: Lecturer, Student, Course, etc.
- Individual objects that belong to a class are referred to as instances of that class.
- The relationship between instances and classes in RDF is through **rdf:type**.

Usefulness of Classes

- Impose restrictions on what can be stated in an RDF document using the schema
  - As in programming languages
  - Disallow nonsense from being stated
- And, it allows to express what is common between different entities.

Usefulness of Classes

- The use of Classes disallow nonsensical statements
- Examples:
  - Discrete Maths is taught by Concrete Maths
    - We want courses to be taught by lecturers only
    - Restriction on values of the property “is taught by” (range restriction)
  - Room 208 is taught by David Billington
    - Only courses can be taught
    - This imposes a restriction on the objects to which the property can be applied (domain restriction)

Class Hierarchies

- Classes can be organized in hierarchies
  - B is a subclass of A if every instance of B is also an instance of A
  - Then A is a superclass of B
- A subclass graph need not be a tree
- A class may have multiple superclasses

Class Hierarchy Example

Inheritance in Class Hierarchies

- Range restriction: Courses must be taught by academic staff members only
  - John is a professor
- He inherits the ability to teach from the class of academic staff members
- This is done in RDF Schema by fixing the semantics of “is a subclass of”
  - It is not up to an application (RDF processing software) to interpret “is a subclass of”

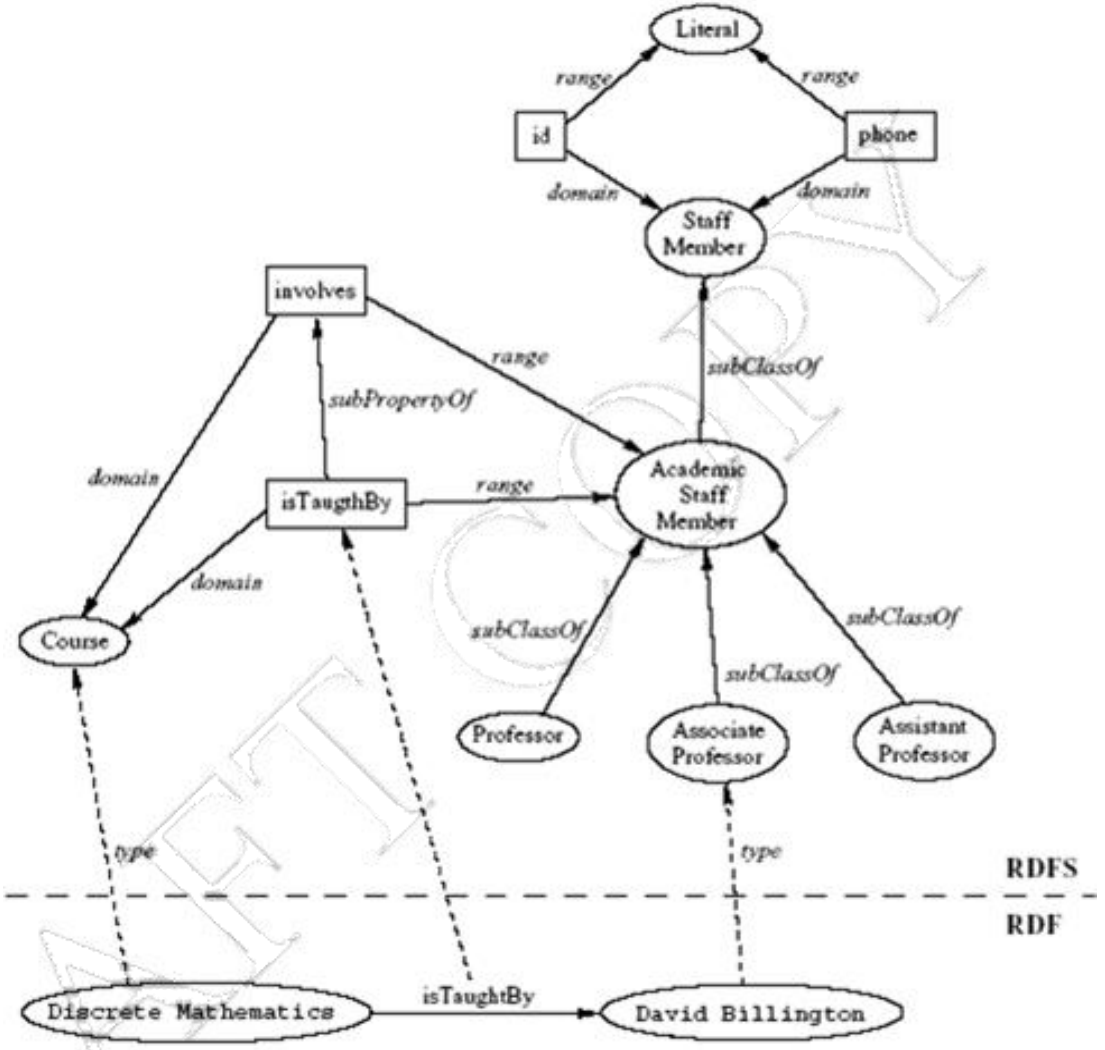
Property Hierarchies

- Hierarchical relationships for properties
  - E.g., “is taught by” is a subproperty of “involves”
  - If a course C “is taught by” an academic staff member A, then C also “involves” A
- The converse is not necessarily true
  - E.g., A may be the teacher of the course C, or
  - a tutor who marks student homework but does not teach C
- P is a subproperty of Q, if Q(x,y) is true whenever P(x,y) is true

RDF Layer vs RDF Schema Layer

- Discrete Mathematics is taught by David Billington
- The schema is itself written in a formal language, RDF Schema, that can express its ingredients:
  - Class, subClassOf, Property, subPropertyOf, Resource, etc.

RDF Layer vs RDF Schema Layer



RDF Schema in RDF

- The modeling primitives of RDF Schema are defined using resources and properties (RDF itself is used!)
- To declare that “Lecturer” is a subclass of “Academic Staff Member”
  - Define resources Lecturer, AcademicStaffMember, and subClassOf
  - Define property subClassOf
  - Write triple
    (lecturer, subClassOf, academicStaffMember)

Core Classes

- **rdfs:Resource**, the class of all resources
- **rdfs:Class**, the class of all classes
- **rdfs:Literal**, the class of all literals (strings)
- **rdf:Property**, the class of all properties.
- **rdf:Statement**, the class of all reified statements

Core Properties

- **rdf:type**, relates a resource to its class
  - The resource is declared to be an instance of that class
- **rdfs:subClassOf**, relates a class to one of its superclasses
  - All instances of a class are instances of its superclass
- **rdfs:subPropertyOf**, relates a property to one of its superproperties

Core Properties

- **rdfs:domain**, specifies the domain of a property P
  - The class of those resources that may appear as subjects in a triple with predicate P
  - If the domain is not specified, then any resource can be the subject
- **rdfs:range**, specifies the range of a property P
  - The class of those resources that may appear as values in a triple with predicate P

Examples

- Using the N3 based syntax of RDF:

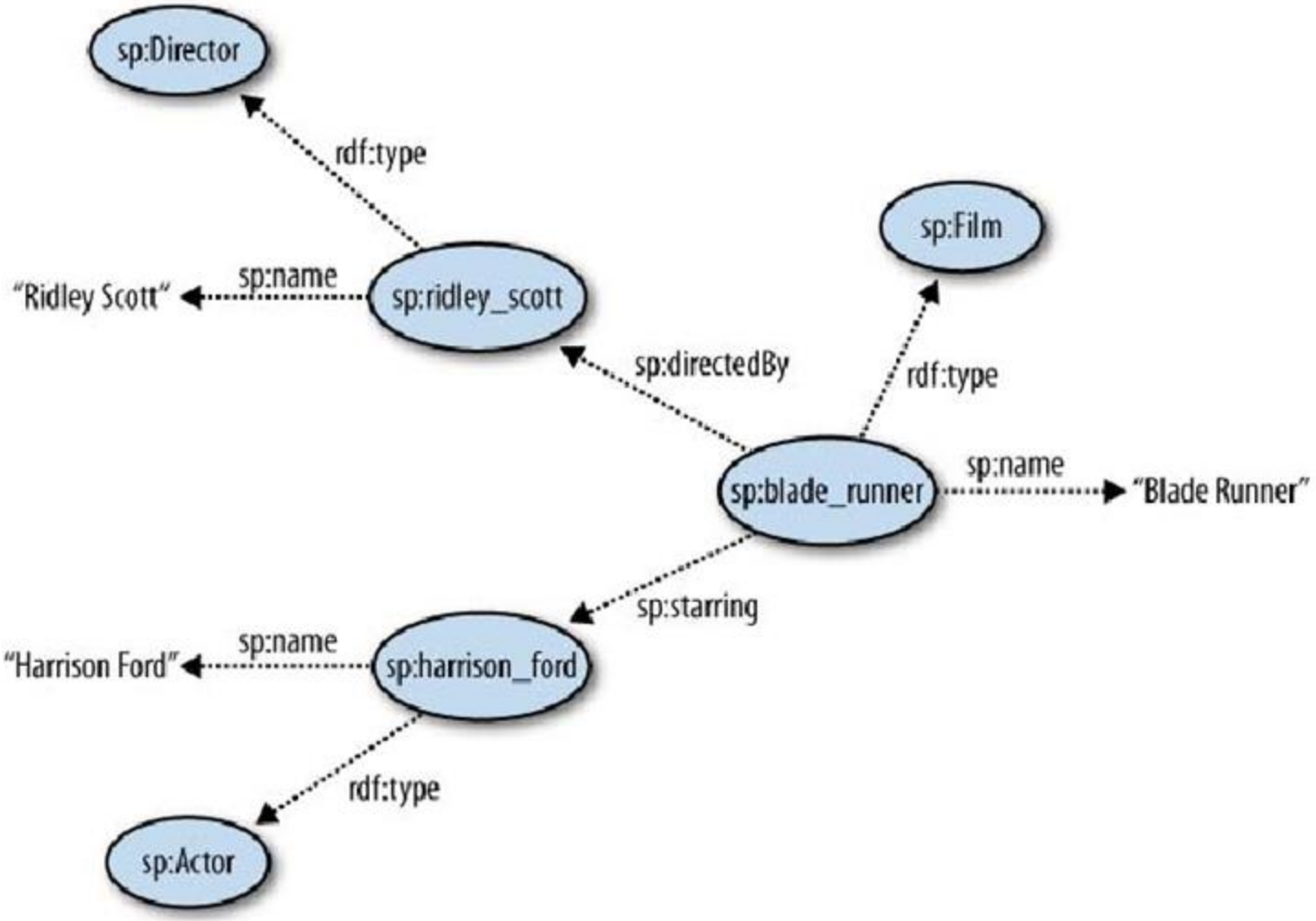
```
prefix rdf:<http://www.w3.org/1999/02/22-rdf-syntax-ns#>
prefix rdfs:<http://www.w3.org/2000/01/rdf-schema#>
prefix ex:<http://example.org#>

ex:Lecturer rdf:type rdfs:Class ;
            rdfs:subClassOf ex:StaffMember .

ex:phone rdf:type rdf:Property ;
         rdfs:domain ex:StaffMember;
         rdfs:range rdfs:Literal .
```

Core Classes and Properties

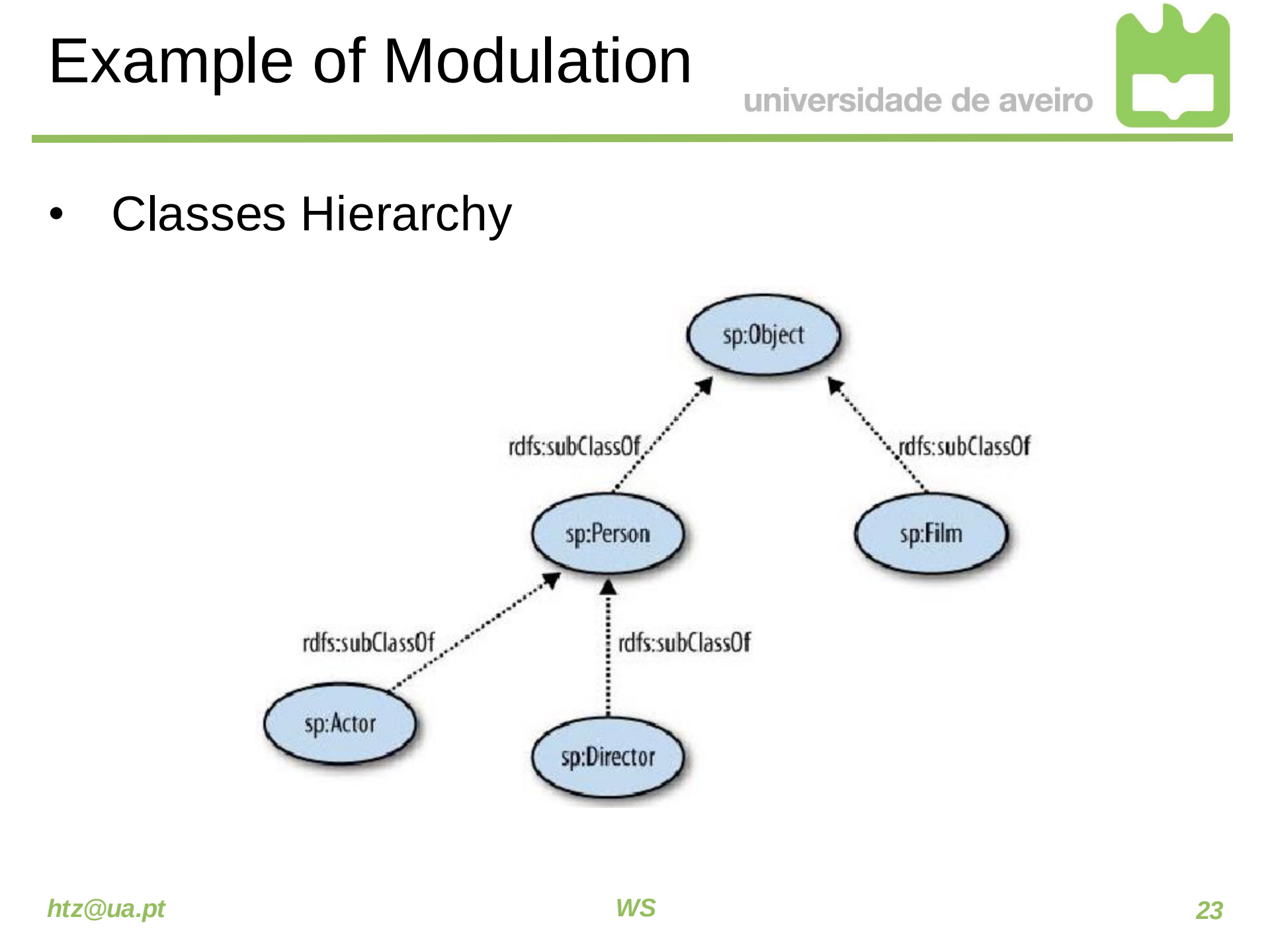
- Relationships between:
- **rdfs:subClassOf** and **rdfs:subPropertyOf** are transitive, by definition
- **rdfs:Resource** is an instance of **rdfs:Class**
  - Because **rdfs:Resource** is of type **rdfs:Class**
    - (rdfs:Resource **rdf:type** rdfs:Class)
- **rdfs:Class** is a subclass of **rdfs:Resource**
  - Because every class is a resource
    - (rdfs:Class **rdfs:subClassOf** rdfs:Resource)
- Every class is an instance of **rdfs:Class**
  - Because every class is of type **rdfs:Class**

Example of Modulation

- Describing the film “Blade Runner”

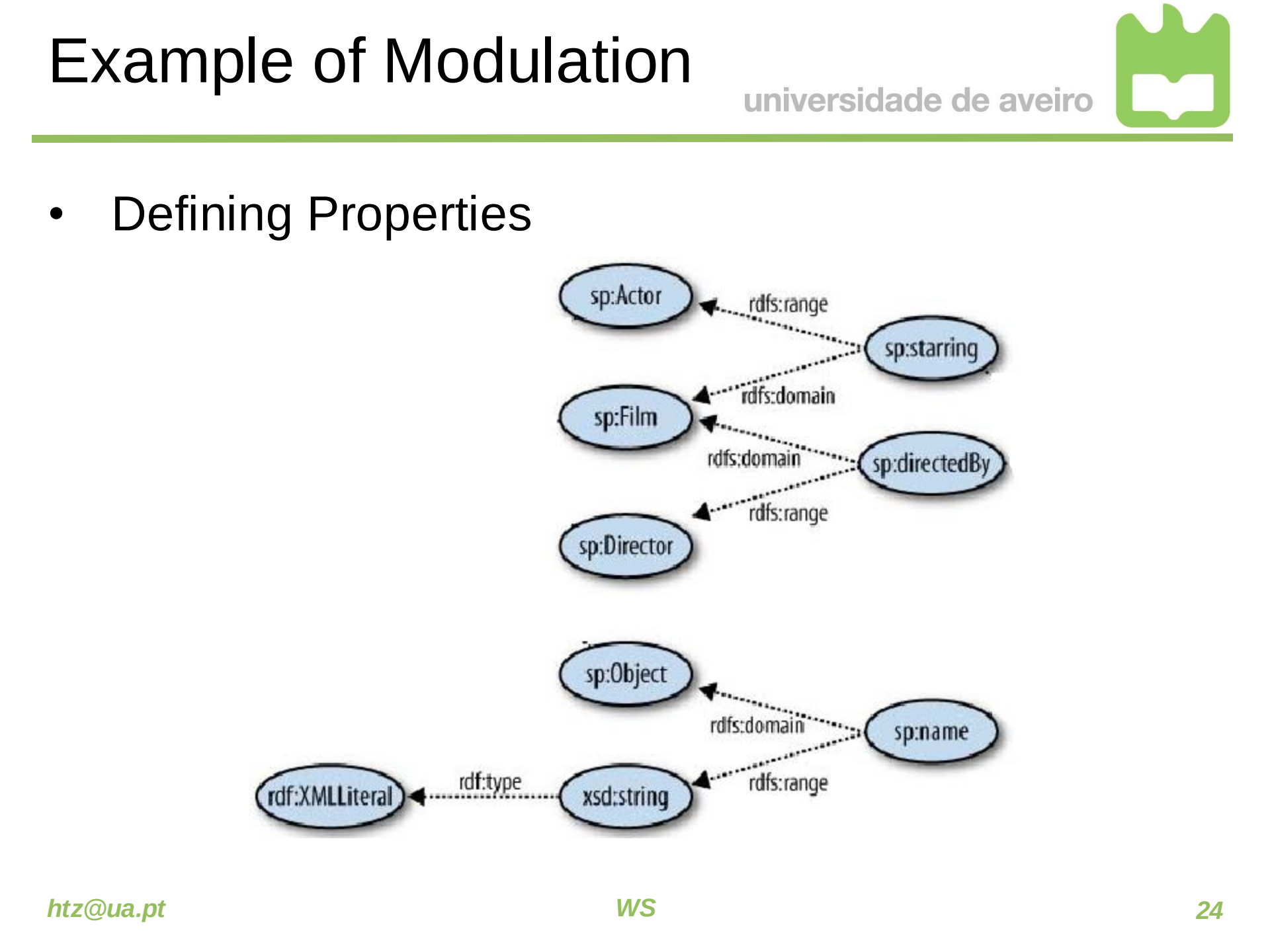


Example of Modulation



- Classes Hierarchy

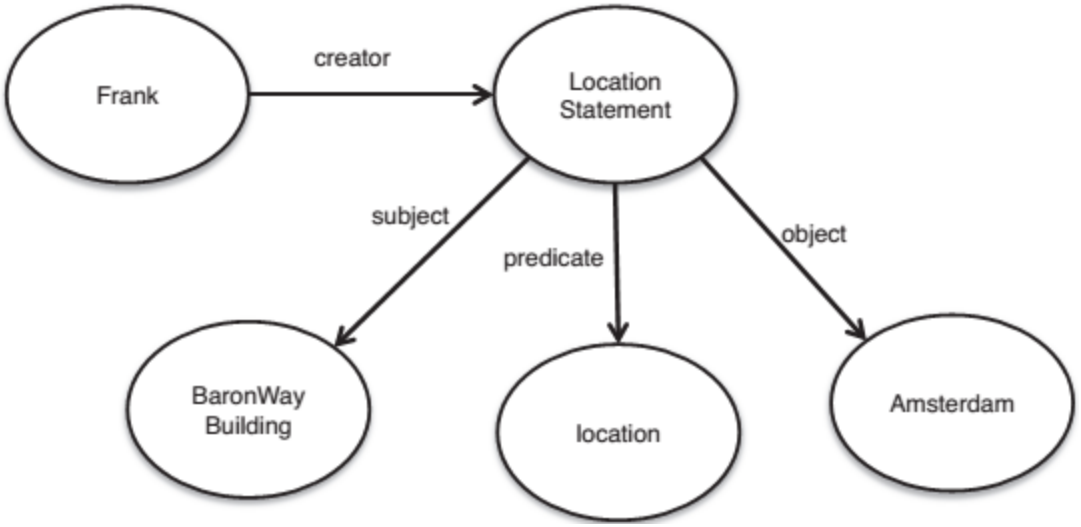
Example of Modulation



- Defining Properties

Reification

- Reification is a process that allows you to create references for statements or parts of graphs.
- An auxiliary object is introduced, representing the statement (triple) and thus it can assume the role of object in another statement.
- Example:

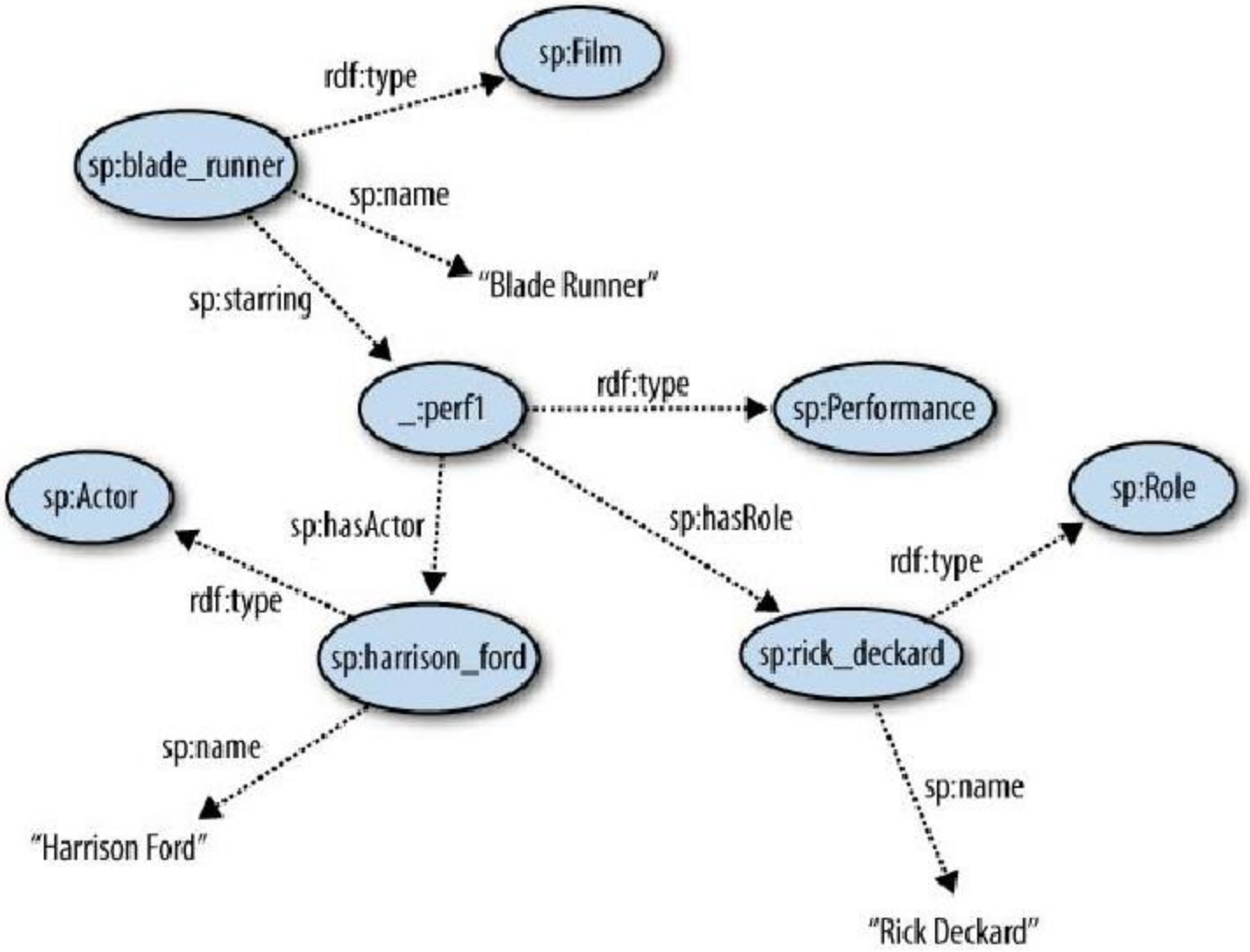


Reification

- Referencing parts of graphs:
- In the theme of films, for example, it is intended to add information about the roles played by actors
- Instead of associating this information with an actor, it can be associated with an abstract entity called **performance** that will represent the sub-graph about the actor and the role played by him.

Reification

- Use of class Performance:



Reification and Containers

- **rdf:subject**, relates a reified statement to its subject
- **rdf:predicate**, relates a reified statement to its predicate
- **rdf:object**, relates a reified statement to its object
- **rdf:Bag**, the class of bags
- **rdf:Seq**, the class of sequences
- **rdf:Alt**, the class of alternatives
- **rdfs:Container**, which is a superclass of all container classes, including the three above

Utility Properties

- **rdfs:seeAlso** relates a resource to another resource that explains it
- **rdfs:isDefinedBy** is a subproperty of **rdfs:seeAlso** and relates a resource to the place where it is defined, typically an RDF schema
- **rdfs:comment**. Comments, typically longer text, can be associated with a resource
- **rdfs:label**. A human-friendly label (name) is associated with a resource

RDF Classes (resume)

universidade de aveiro

Class name	Comment
rdfs:Resource	The class resource, everything.
rdfs:Literal	The class of literal values, e.g. textual strings and integers.
rdf:XMLLiteral	The class of XML literals values.
rdfs:Class	The class of classes.
rdf:Property	The class of RDF properties.
rdfs:Datatype	The class of RDF datatypes.
rdf:Statement	The class of RDF statements.
rdf:Bag	The class of unordered containers.
rdf:Seq	The class of ordered containers.
rdf:Alt	The class of containers of alternatives.
rdfs:Container	The class of RDF containers.
rdfs:ContainerMembershipProperty	The class of container membership properties, rdf:_1, rdf:_2, ..., all of which are sub-properties of 'member'.
rdf:List	The class of RDF Lists.

RDF Properties (resume)

Property name	Comment
rdf:type	The subject is an instance of a class.
rdfs:subClassOf	The subject is a subclass of a class.
rdfs:subPropertyOf	The subject is a subproperty of a property.
rdfs:domain	A domain of the subject property.
rdfs:range	A range of the subject property.
rdfs:label	A human-readable name for the subject.
rdfs:comment	A description of the subject resource.
rdfs:member	A member of the subject resource.
rdf:first	The first item in the subject RDF list.
rdf:rest	The rest of the subject RDF list after the first item.
rdfs:seeAlso	Further information about the subject resource.
rdfs:isDefinedBy	The definition of the subject resource.
rdf:value	Idiomatic property used for structured values (see the RDF Primer for an example of its usage).
rdf:subject	The subject of the subject RDF statement.
rdf:predicate	The predicate of the subject RDF statement.
rdf:object	The object of the subject RDF statement.