Knowledge-Based Systems

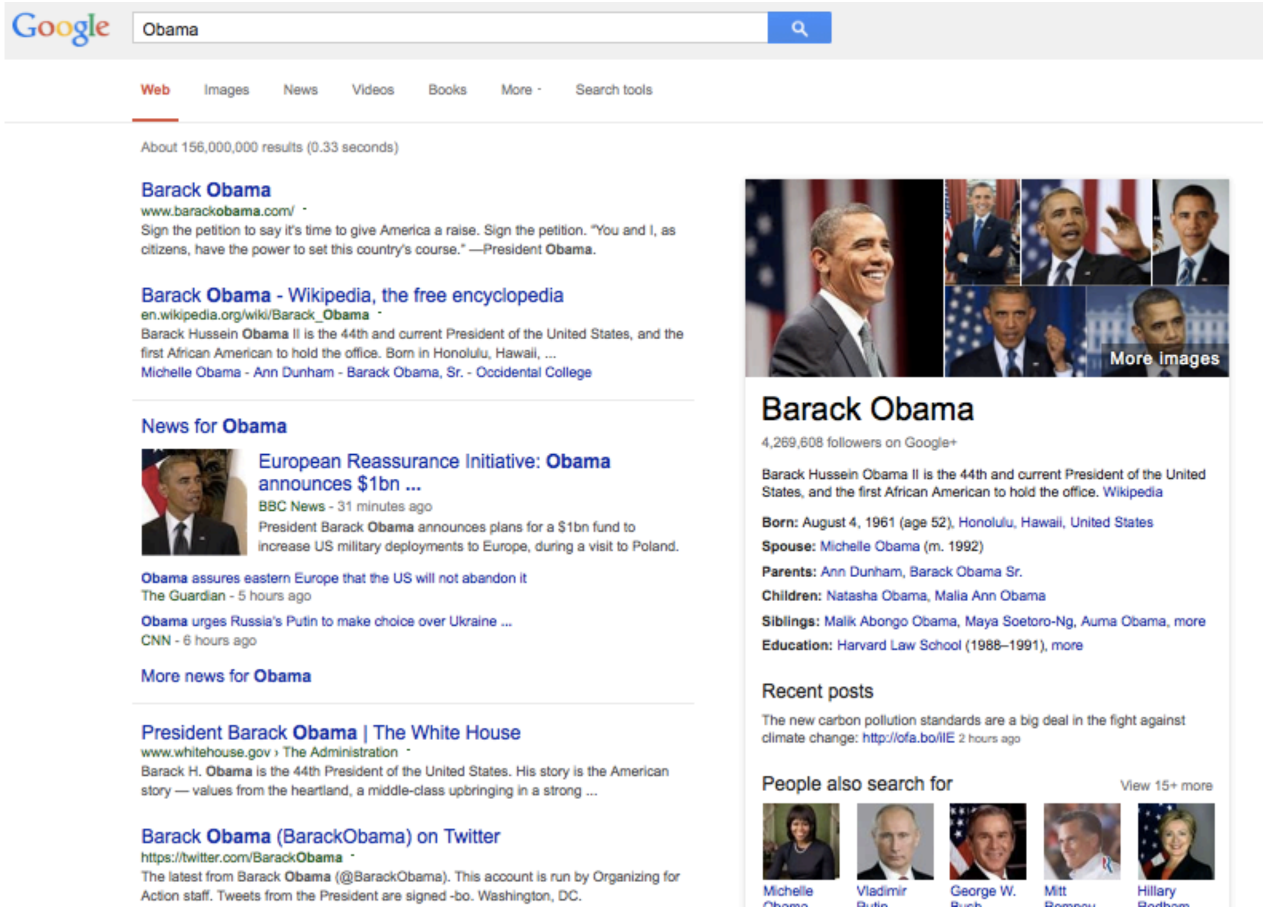
Ontology Modelling: From Semantic Networks to Ontologies

Jeff Z. Pan

http://homepages.abdn.ac.uk/jeff.z.pan/pages/

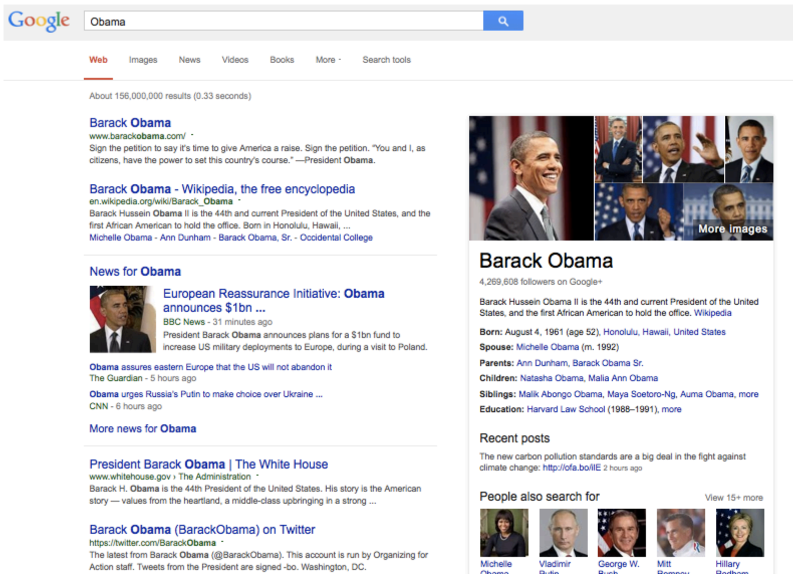
RDF: Standard of Semantic Network

- Building block of RDF: subject-property-value

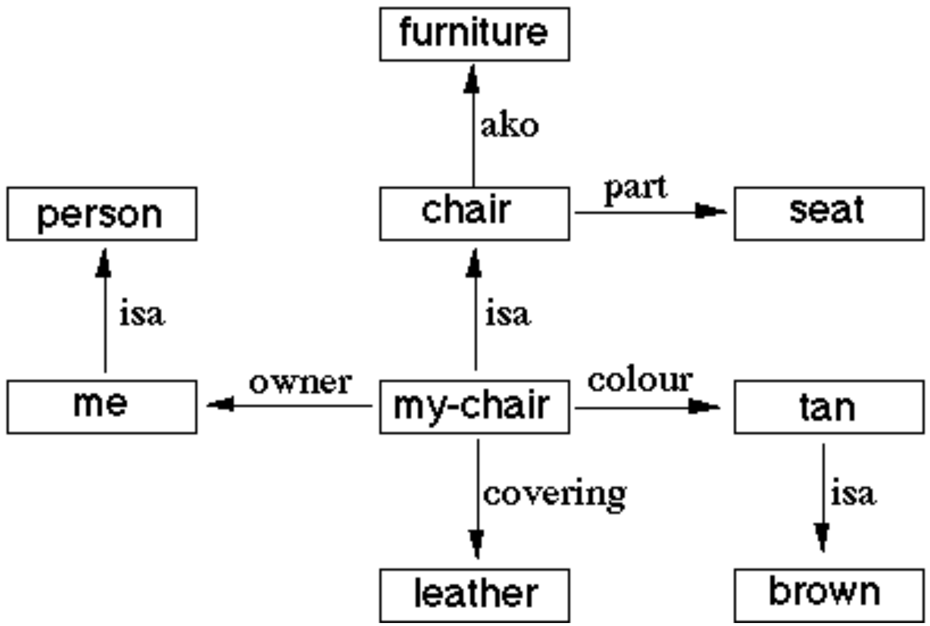


Some Example RDF Triples in N3

- [ggl:barak-obama rdf:type ggl:Politician .]
- [ggl:Politician rdfs:subClassOf ggl:Person .]
- [ggl:barak-obama ggl:born 1961-08-04 .]
- [ggl:barak-obama ggl:spouse ggl:michelle-obama .]
- [ggl:barak-obama ggl:education ggl:harvard-law-school .]



RDF and Semantic Network



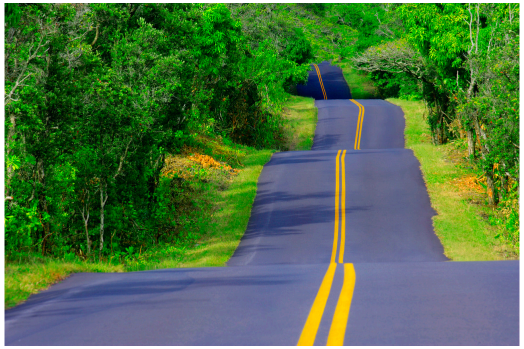
[my-chair colour tan .]

[my-chair rdf:type chair .]

[chair rdfs:subClassOf furniture .]

...

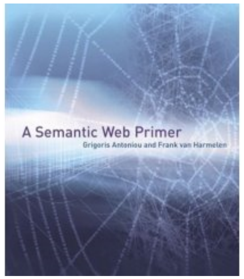
Roadmap



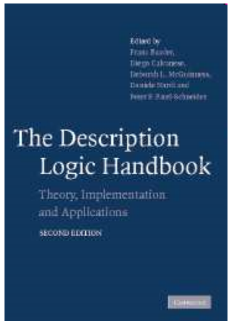
- Foundation
  - KR, ontology and rule; set theory
- Knowledge capture
- Knowledge representation
  - Ontology: Semantic Web standards RDF and OWL, Description Logics
  - Rule: Jess
- Knowledge reasoning
  - Ontology: formal semantics, tableaux algorithm
  - Rule: forward chaining, backward chaining
- Knowledge reuse and evaluation
- Meeting the real world
  - Jess and Java, Uncertainty

Lecture Outline

- Motivation
- RDF, Semantic Networks and OWL
- From Semantic Networks to OWL Ontologies
- Practical



[Section 4.1 - 4.3]



[Chapter 14]

Motivations

- Need to better understand RDF (standard of semantic network)
  - in terms of reasoning
  - in terms of relation to database
- Many human knowledge is still represented in natural language
- Ontology modelling
  - natural language -> semantic network
  - semantic network -> OWL ontologies

Lecture Outline

- Motivation
- RDF, Semantic Networks and OWL
- From Semantic Networks to OWL Ontologies
- Practical

RDF Schema Entailment Rules (1)

- [rdfs2]
  - [p rdfs:domain C .] [a p b .] => [a rdf:type C .]
- [rdfs3]
  - [p rdfs:range D .] [a p b .] => [b rdf:type D .]
- [rdfs5]
  - [p1 rdfs:subPropertyOf p2 .] [p2 rdfs:subPropertyOf p3 .] =>

    [p1 rdfs:subPropertyOf p3 .]

RDF Schema Entailment Rules (2)

- [rdfs7]
  - [p1 rdfs:subPropertyOf p2 .] [a p1 b .] => [a p2 b .]
- [rdfs9]
  - [C rdfs:subClassOf D .] [b rdf:type C .] => [b rdf:type D .]
- [rdfs11]
  - [C1 rdfs:subClassOf C2 .] [C2 rdfs:subClassOf C3 .] => [C1 rdfs:subClassOf C3 .]

Example: RDF Schema Entailment Rules

- Example entailment rule [rdfs9]:

[b rdf:type C .], [C rdfs:subClassOf D .] => [b rdf:type D .]

- Since we have

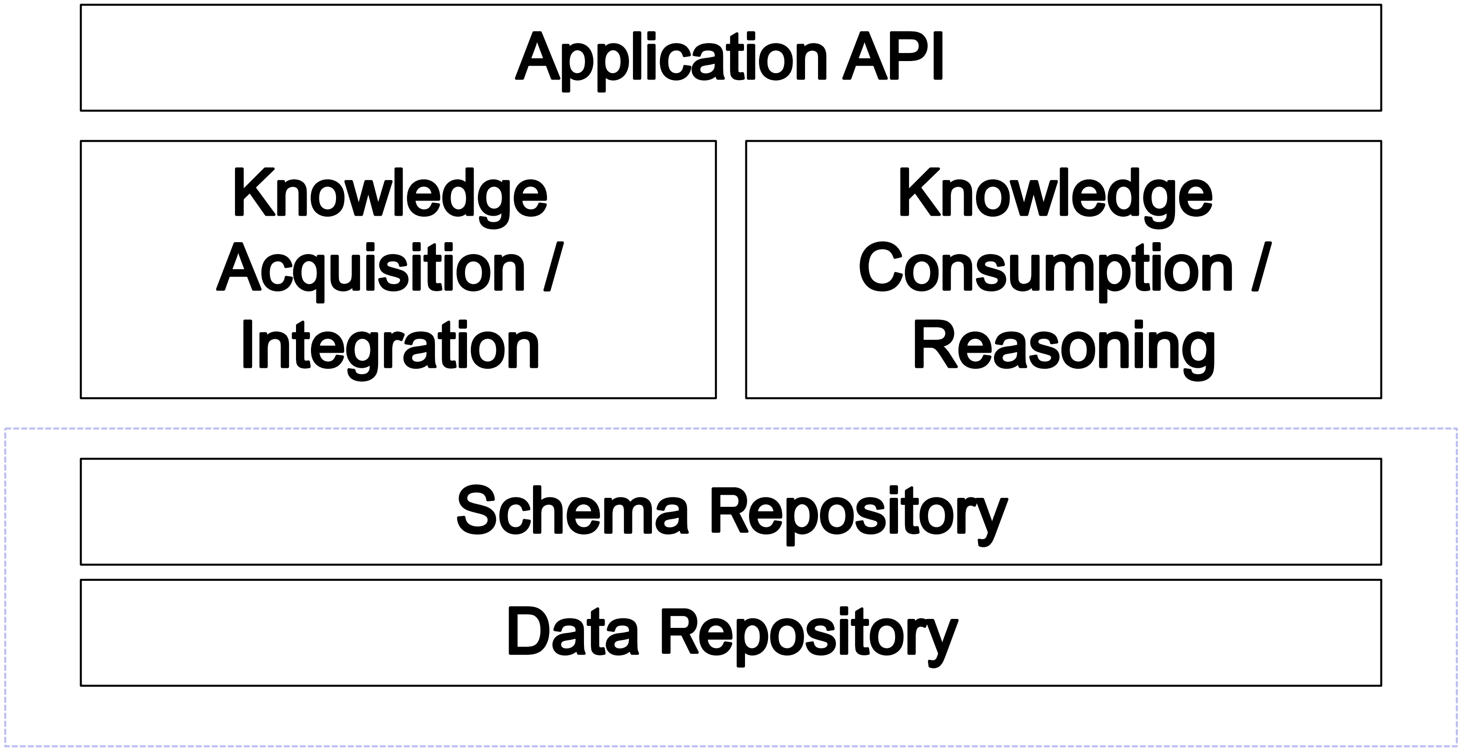
[my-chair rdf:type chair .]

[chair rdfs:subClassOf furniture .]

by applying the above rule, we have

[my-chair rdf:type furniture .]

Architecture of Knowledge Based Systems

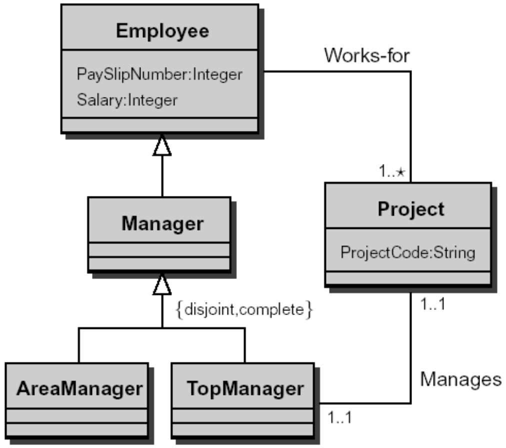


Knowledge Representation in RDF

- Start from a semantic network
- Transformed from a relational database

- From natural language sentences
- Question: can you translate the following statement into RDF?
  – Student is a person
  [Student rdfs:subClassOf Person .]
  – Student attends a course
  [attend rdfs:domain Student .]
  [attend rdfs:range Course .]
  – Professor teaches a course
  [teach rdfs:domain Professor .]
  [teach rdfs:range Course .]
  – Person has first name, last name and email address
  Not representable in RDF

Key Modelling Elements in RDF/ OWL



- Key elements: object, class (concept), property (relation)
  - in RDF, everything is a resource, including classes and properties
- A class is a set of objects
  - Employee: {E1, E2, E3, E4}
  - Project: {P1, P2}
- A property is a set to pairs (tuples) of objects
  - Works-for: {<E1,P1>, <E2,P1>, <E2,P2>, <E3,P1>, <E3,P2>, <E4,P2>}

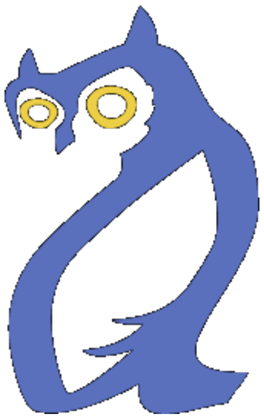
Database and RDF

- An RDF statement is a data unit with global and linkable ids for data and schema

Student ID	Name	take-course
p001	John	cs3019
p002	Tom	cs3023

  - [csd:p001 rdf:type csd:Student .]
  - [csd:p002 rdf:type csd:Student .]
  - [csd:p001 csd:name “John” .]
  - [csd:p002 csd:name “Tom” .]
  - [csd:p001 csd:take-course csd:cs3019 .]
  - [csd:p002 csd:take-course csd:cs3023 .]

Two Types of Properties in OWL



- **In OWL there are two kinds of properties**
  - **Object properties, which relate objects to other objects**
    - **E.g. take-course, supervises**
  - **Datatype properties, which relate objects to datatype values**
    - **E.g. Name, title, age, etc.**

Student ID	Name	take-course
p001	John	cs3019
p002	Tom	cs3023

UNIVERSITY OF ABERDEEN

Database and RDF

- Question: How about the foreign key to the Course table?
  - Answer??: [csd:take-course rdfs:range csd:Course .]

(very close, but not quite, due to the difference between open world and closed world assumptions)

How about Primary Key

- RDF is NOT expressive enough to represent primary key
- We need OWL (Web Ontology Language) for that

- Example with Friend Of a Friend (FoaF) ontology
  - foaf:OnlineAccount owl:hasKey

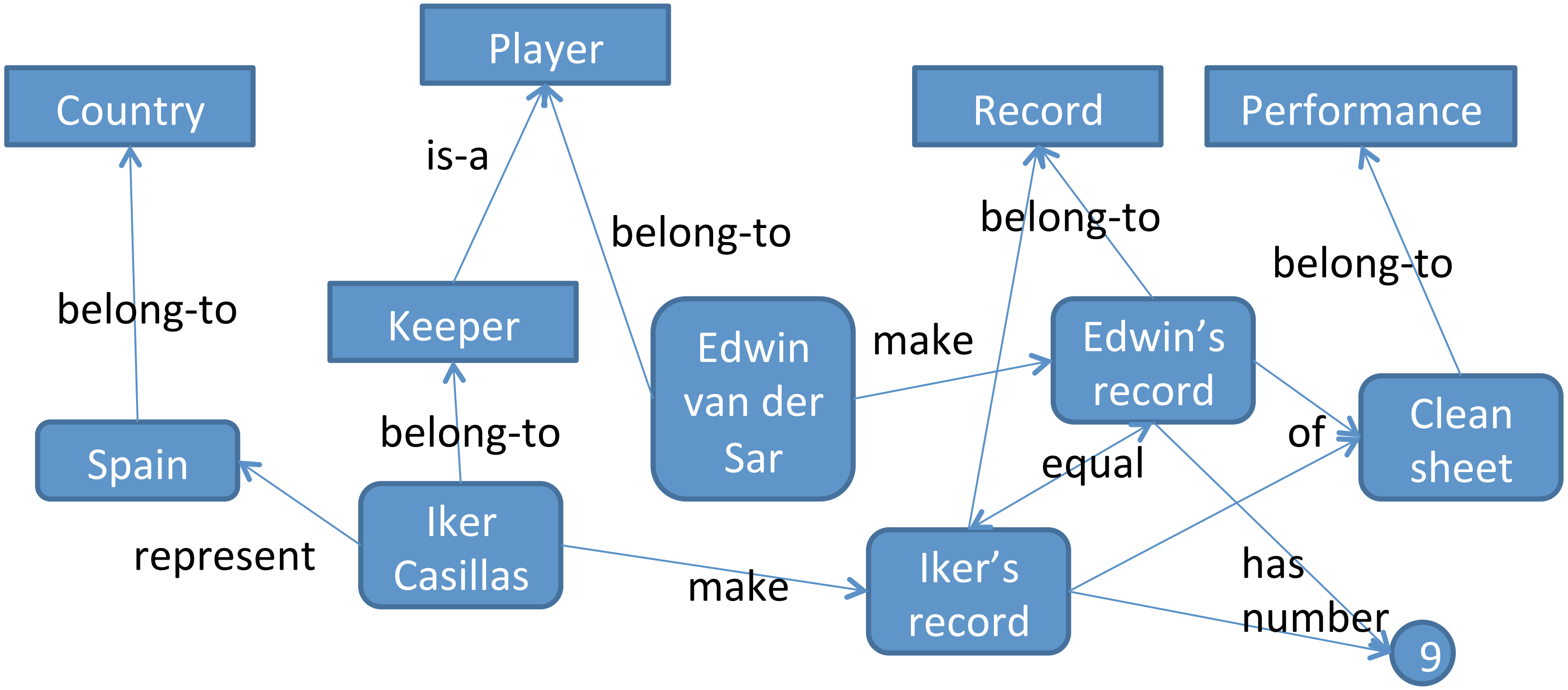
    (foaf:accountName

    foaf:accountServiceHomepage)

Lecture Outline

- Motivation
- RDF, Semantic Networkds and OWL
- From Semantic Networks to OWL Ontologies
- Practical

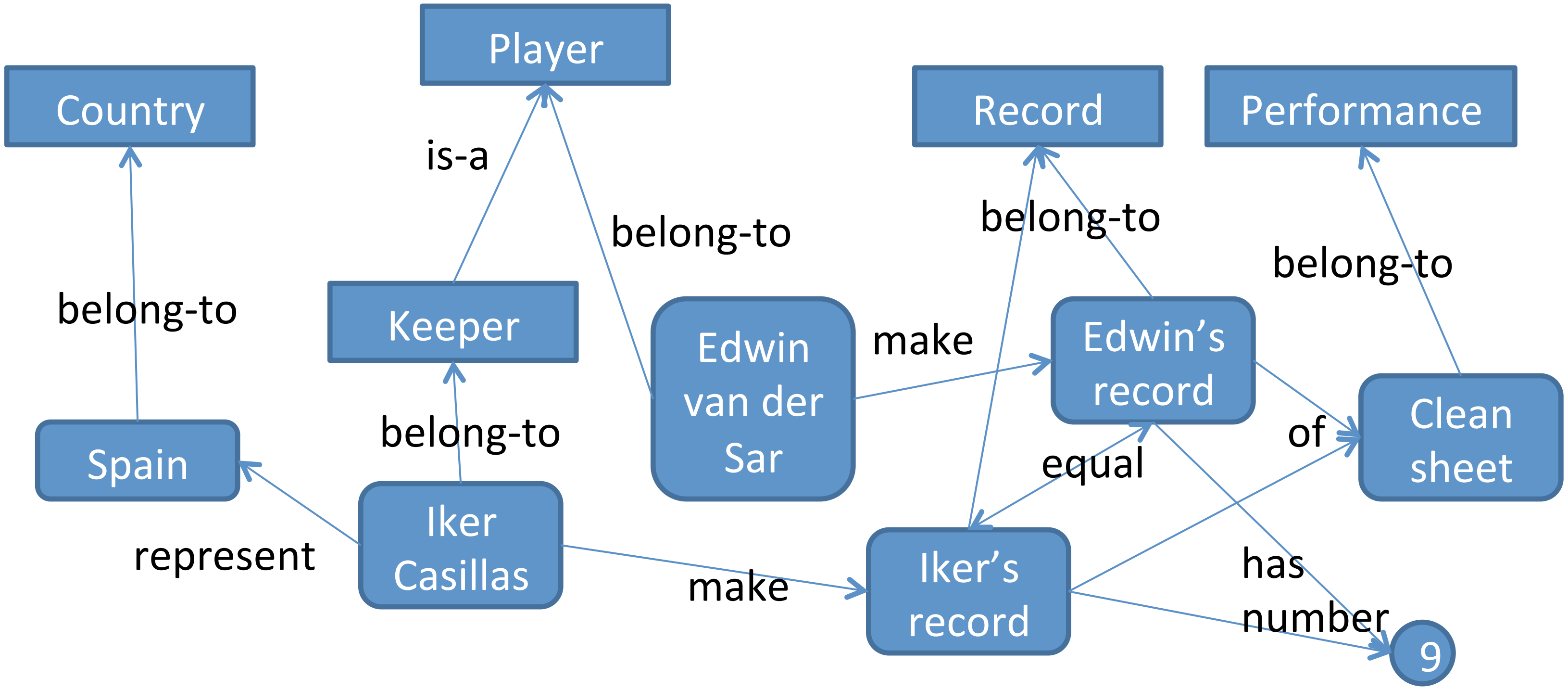
Semantic Network



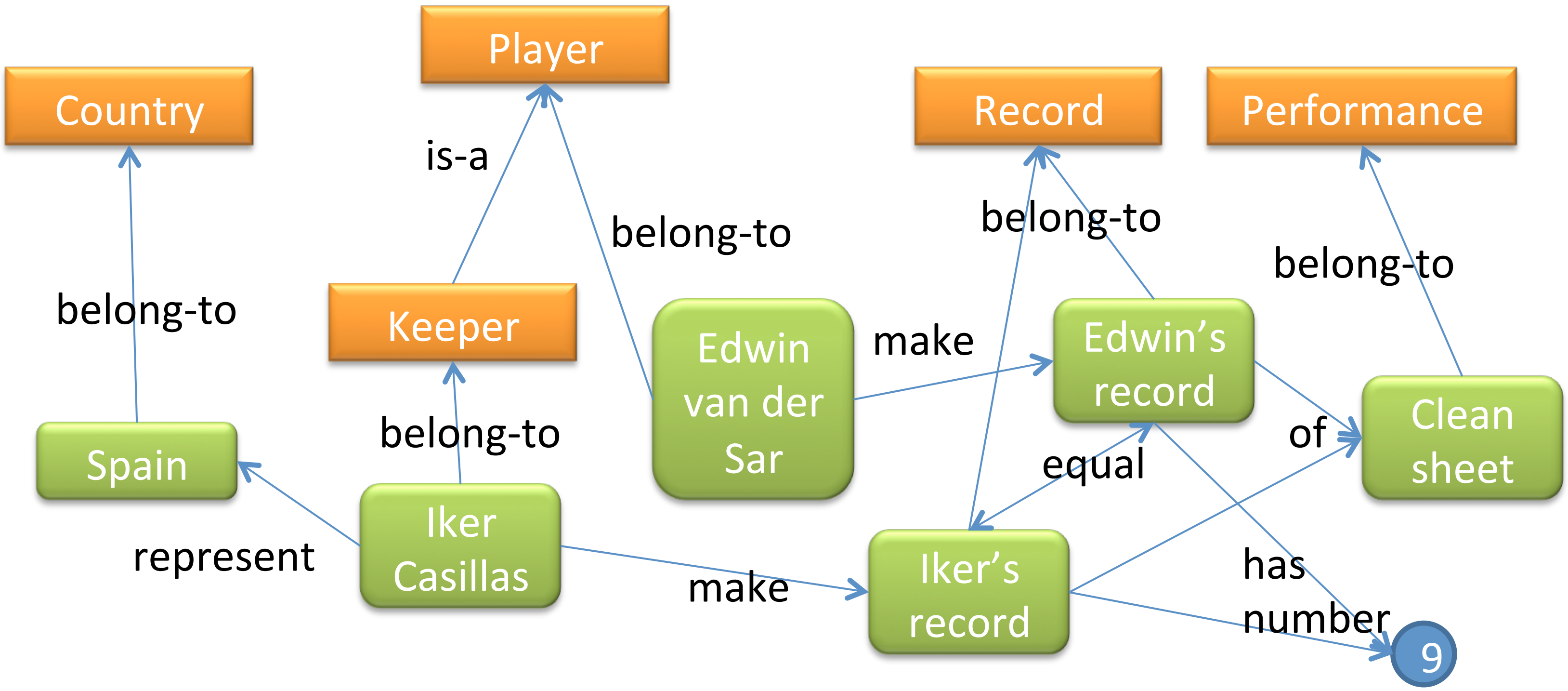
Separating Concepts and Individuals

- In semantic networks, boxes can be concepts and individuals
- In ontologies, concepts and individuals are quite different things
  - An individual is interpreted as a domain element
    - E.g. “Iker Cassilas” is an individual person in the domain
  - A concept is interpreted as a set of domain elements
    - E.g. “Keeper” is the set of all keepers in the domain, including “Iker Cassilas”
  - Of course, whether a box is a concept or individual depends on the granularity of the domain
    - E.g. “cat” can be an individual if the ontology is about different species of “animal”
    - “cat” can be a concept if the ontology is about many individual cats, your cats, my cats, etc.

Semantic Network



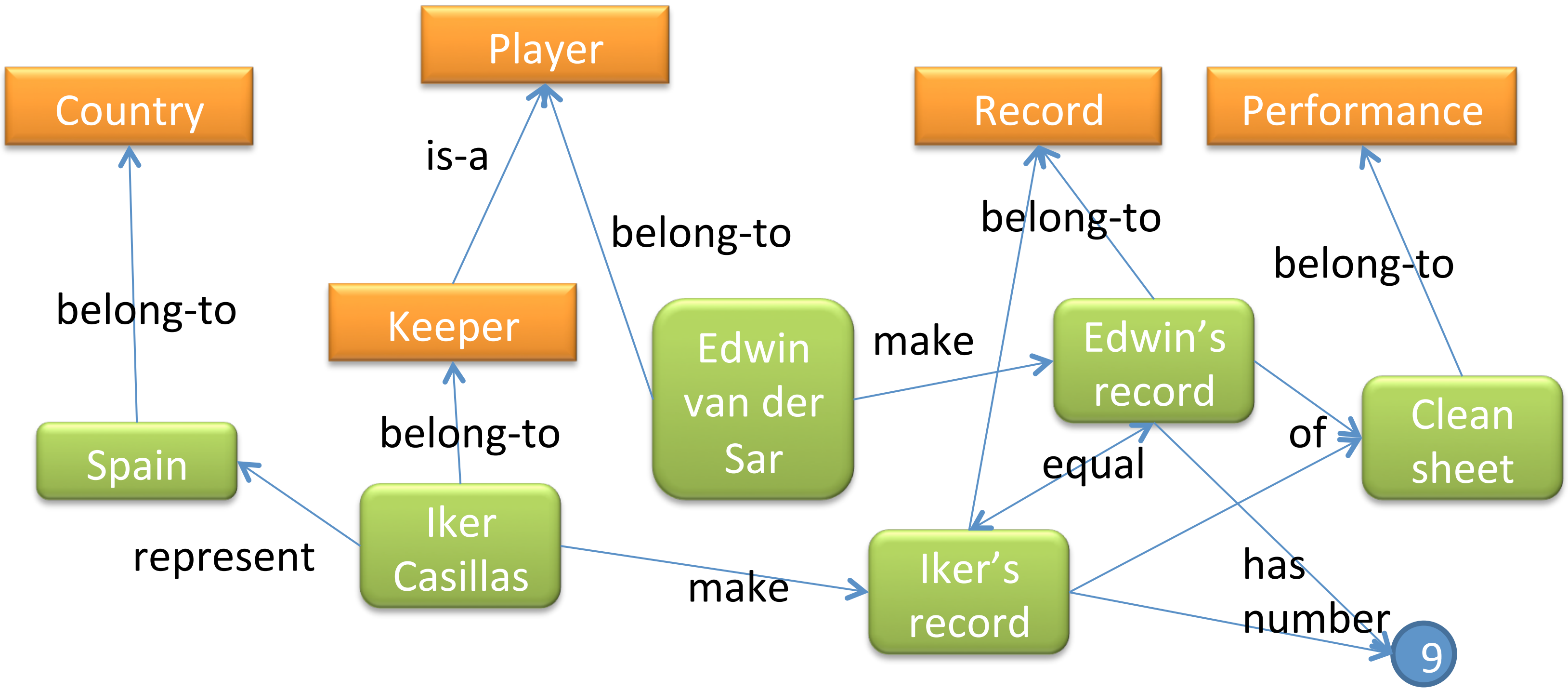
Separating Concepts and Individuals



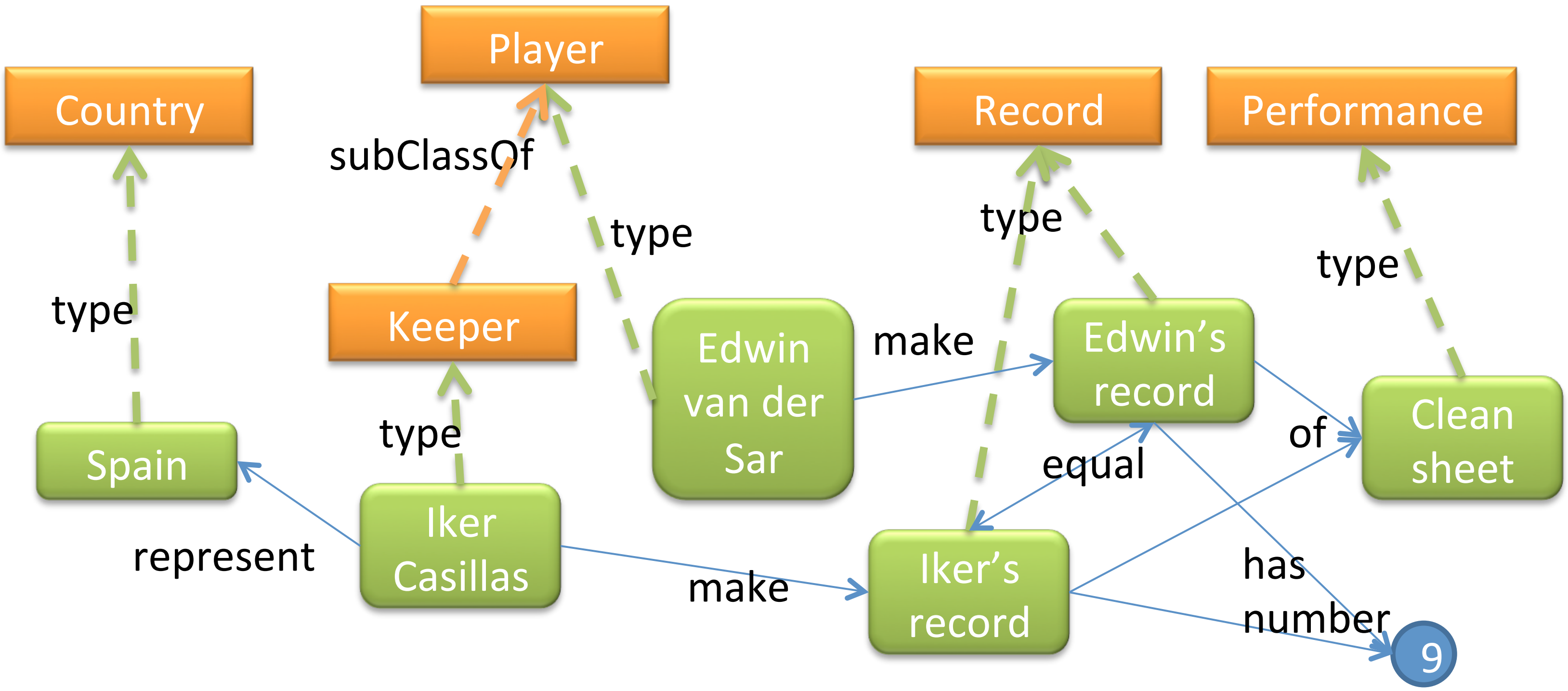
Identifying Concept/Property Subsumption and Individual Classifications

- In Semantic Networks, edges can be domain properties or predifined properties in RDF/OWL, such as
  - subClassOf, subPropertyOf and type
- An individual x is an instance of a concept C if x is interpreted as an element of the set that C is represents
- A concept A is a sub-concept of a concept B if every instance of A is an instance of B
  - E.g. every keeper is also a player, “Iker Casillas” is a “Keeper”,
  - “Keeper” is a sub-concept of “Player” implying that “Iker Casillas” is a “Player”
- A property R is a sub-property of a property S if every instance of R is an instance of S

Semantic Network



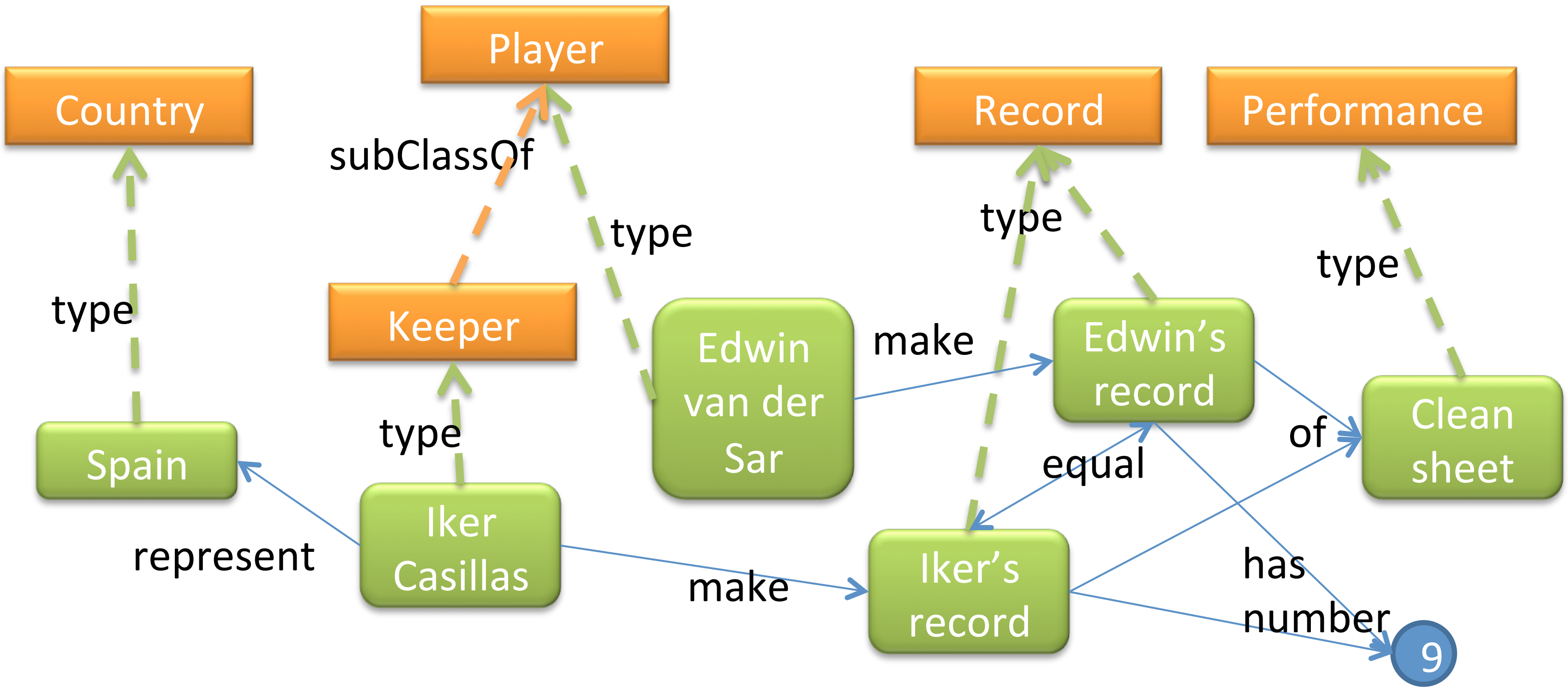
Identifying Concept/Property Subsumption and Individual Classifications



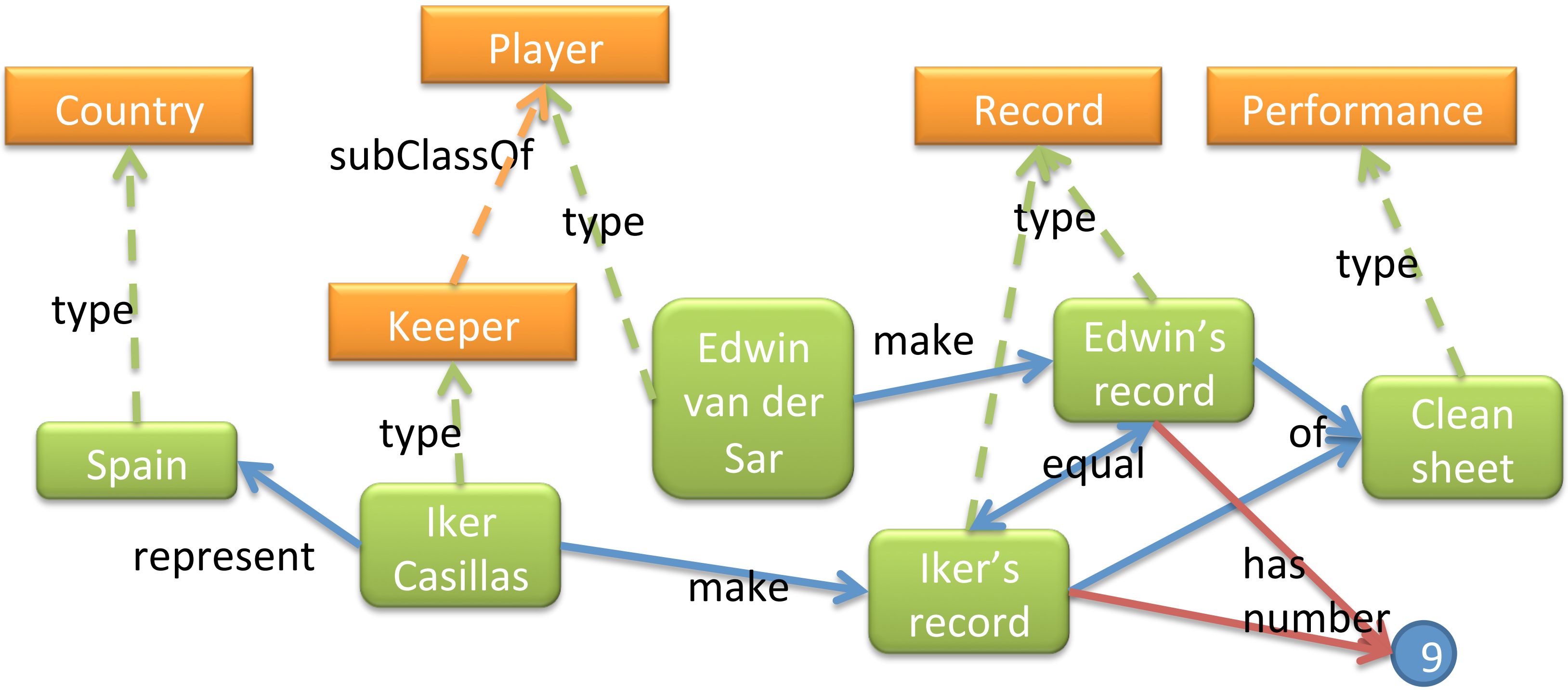
Identifying Object Properties and Datatype Properties

- Object properties represent relations between two individuals
  - E.g. “Iker Casillas” and “Spain” are both individuals, hence Iker Casillas “represents” Spain is an object property.
- Datatype properties represent relations between an individual and a datatyped value
  - Data value includes numbers, dates, strings, etc.
  - E.g. “Iker’s record” is an individual, “9” is an integer, hence Iker’s record “has number” 9 is a datatype proeprty.

Semantic Network



Identifying Object Properties and Datatype Properties



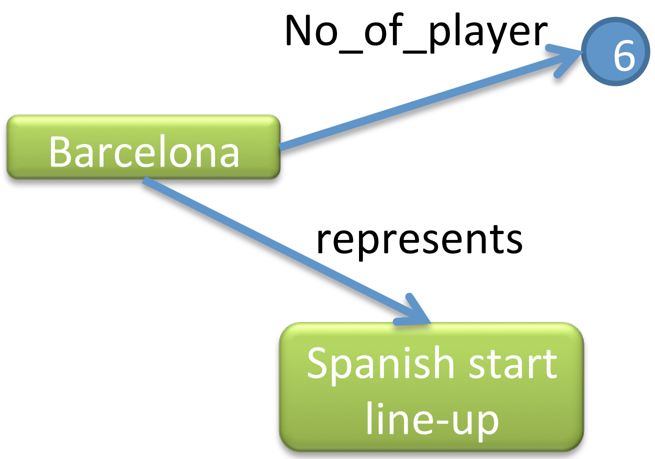
Constructing the Ontology

1. Adding concepts
   – E.g. Country, Keeper, Player, Record, Performance
2. Adding Object and datatype properties
   – E.g. represents, make, equal, etc., has_number, etc.
   – Domain and range
     • Domain of a property is a concept
       – E.g. Domain(represents) = Player
     • Range of an object property is a concept
       – E.g Range(represents) = Country
     • Range of a datatype property is a datatype
       – E.g. Range(has_number) = Integer
     • Based on the types of subjects and objects of relations
3. Adding individuals
   – E.g. spain, Iker_Casillas, Iker's_record, etc.
4. Adding individual relations and datatype property values
   – E.g. represents(Iker_Casillas, spain), make(Iker_Casillas, Iker's_record), has_number(Iker's_record, 9)

A Process of Refinement

- The procedure of constructing an ontology is also a procedure in which you examine and refine your semantic networks!

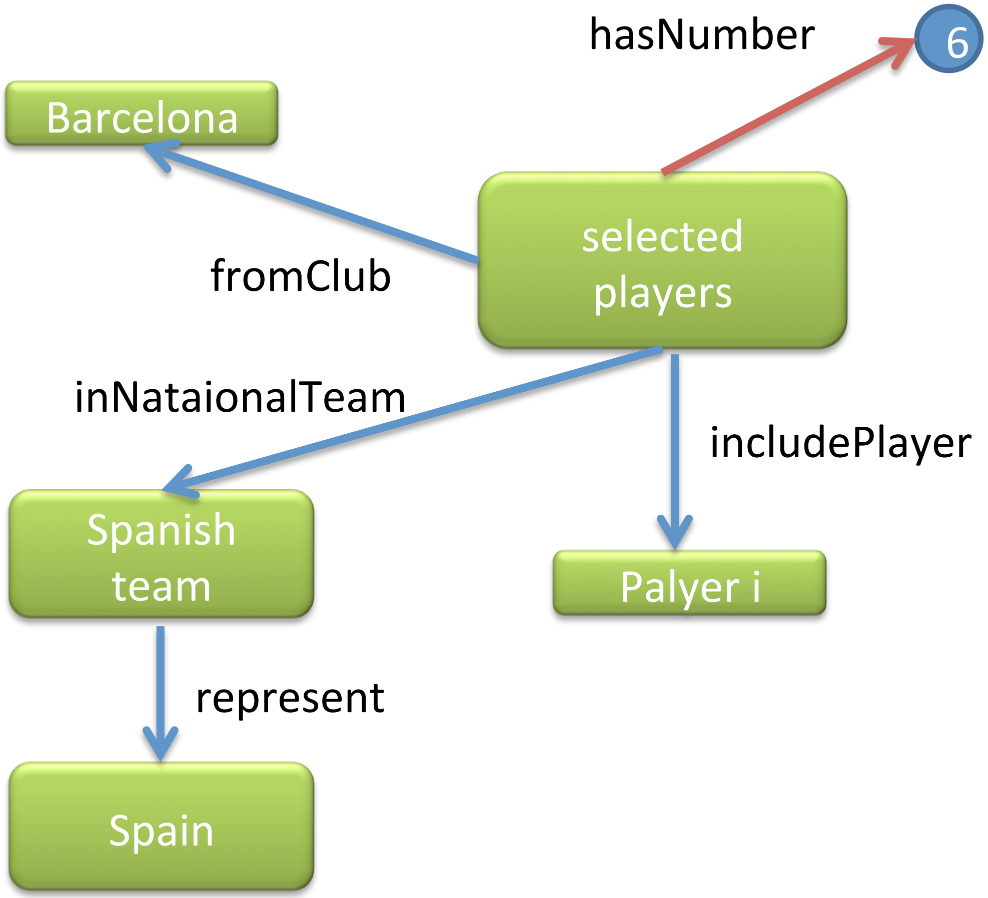
- A typical example: multi-entity relationships
  - E.g. *"Six players from Barcelona were in the Spanish starting line-up"*
    - This is a complex relationship involving a "Country's team", a "club team", and "6" players
  - In semantic networks, one may draw a figure like this or similar:



- This is not semantically precise:
  - Barcelona has 6 players?
    - No, it has 6 players in the Spanish start line-up
  - Barcelona represents Spain?
    - No, its 6 players represent Spain in the start line-up
- This relationship makes sense only when you take all these entities into account!

A Process of Refinement

- An alternative representation
  - there could be others
- Constructing ontology helps you re-evaluate and refine your semantic network.



Practical

- Revisit your semantic networks and see
  - how to construct OWL ontologies accordingly
- How to construct OWL ontologies
  - ontology editors <-> OWL abstract syntax
  - using OWL API

Summary

- RDF and other modelling mechanisms
  - Modern version of semantic network, with formal syntax and semantics
  - Can represent data and foreign keys in relational data model
    - but not primary key
- OWL provides much expressiveness that RDF does not have
- Unerstand how to build OWL ontologies from semantic networks