



# CONCEPTUAL MODELS VERSUS ONTOLOGIES

**Stefano Spaccapietra**

Honorary Professor at EPFL  
Switzerland

# Quality of ICT

## ■ Perspectives on Quality of Information Technologies

### ● Quantitative => performance

- ◆ Massive amounts of data
- ◆ Superfast accessing
- ◆ ...

### ● Qualitative => quality of services ←

- Relevance of information
- Accuracy of information
- Exhaustiveness
- ...

# Quality of Information Services

- To achieve quality services, the quality of the information/ knowledge you have at your disposal is essential
- What is relevant?
- Which consistency rules apply?
- Which business rules govern the data?
- Which evolution rules constrain the possible future states?
- Which interpretations are correct?
- Which preferences have to be taken into account?
- Is it understandable?
- ...

# Knowledge (& Process) Description

- Information and knowledge description/modeling
  - One of the basic, very important domains in computer science
- A plethora of approaches, models, languages, tools, developed by several communities
  - Database, Artificial intelligence, Ontologies, ...
- Abstracting from the specific modalities, two different visions of knowledge acquisition and handling can be identified:
  - Knowledge ownership
  - Knowledge sharing

# The Knowledge Ownership Context

- Typically relying on **database** technology
- Knowledge is specifically acquired to serve some data management application
  - Traditional example: company databases
  - The company “owns” the data: company management rules apply, only authorized persons can create and access the data, every action on the data is strictly centrally controlled relying on the knowledge stored in the central database schema
- The best quality of a database schema is achieved using a conceptual modeling approach:
  - Conceptual Schema: offers semantic richness and technology independence
- This context is known as “**Closed World Assumption**”: only what is known is true

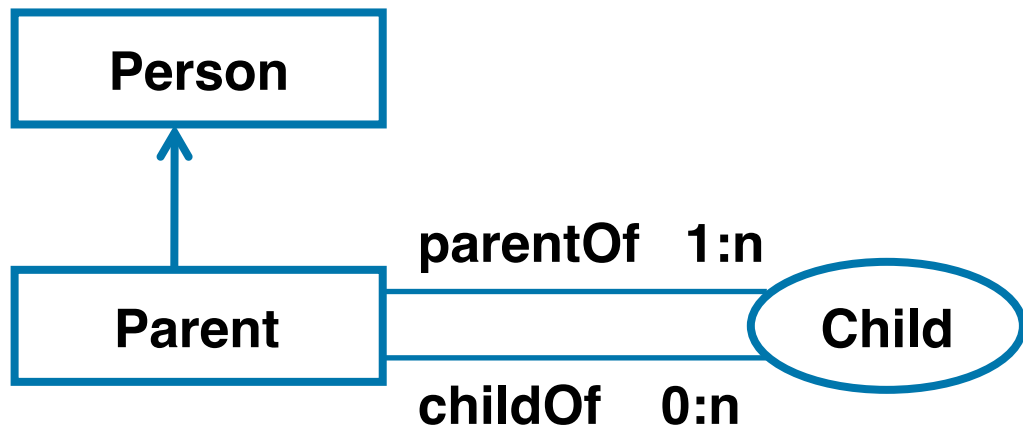
# The Knowledge Sharing Context

- Typically relying on **ontology** technology
- Knowledge is usually acquired from a variety of sources (in particular from the web) that are not necessarily consistent, to support data oriented services usually in some specific domain
- Well-suited for web-based environments, where huge amounts of knowledge are available without any centralized control
- Trustability is not guaranteed, but formal correctness is (thanks to reasoning capabilities)
- This context is known as “**Open World Assumption**”: we only know a subset of the world
- It is a decentralized permissive approach

# CWA / OWA reasoning example1

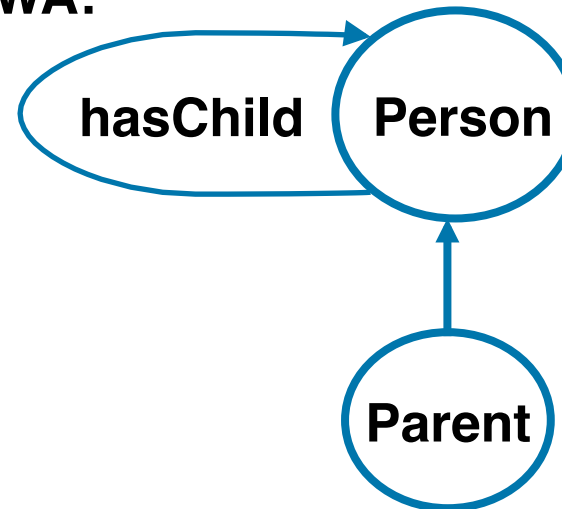
- Parent = instances of Person that have at least one child
- john: Parent

CWA (DB):



=> Creation of john is rejected

OWA:



=> john is an instance of Person  
john has at least one role  
hasChild (unknown)

# CWA / OWA reasoning example2

- john has at most 2 children
- john has a child: peter
- john has a child: mary
- john has a child: paul

## CWA(DB):

=> john has a child: peter    OK  
      john has a child: mary    OK

john has a child: paul => **rejected**

## OWA:

=> (paul should be the same as peter)  
or (paul should be the same as mary)  
or (mary should be the same as peter)

**OK**

- Open world assumption +  
   no unique name assumption
- (but implemented reasoners  
   have unique name assumption)



# Conceptual Schema / Ontology Design

## ■ Conceptual schema and database design

- = Representation of some part of the real world: the real world of interest.
- It addresses the needs of a well-defined community (e.g., an enterprise)
- It is defined and maintained by IS specialists
- It enforces a common vocabulary within the enterprise
- It constrains what people are allowed to do in terms of data management

## ■ Ontology and ontology design

- = Explicit specification of a conceptualization of a part of the real world
- Ontology design is collaborative and incremental
- ==> Needs reasoning services
  - ◆ to check consistency of the specifications
  - ◆ to accurately integrate new knowledge
  - ◆ to infer all inferable knowledge
- Formal reasoning => logic-based approaches
- Understandable by a machine (semantic web)
- Suitable for knowledge exchange (big data)

# Quality Improvement

- Whatever approach we use (Closed world versus Open world) we can improve its quality by learning from the other approach
- Can conceptual modeling benefit from ontologies?
  - Theoretical support for:
    - Formal definition of conceptual models
    - Providing the basic concepts for new conceptual models
    - Checking the consistency of a schema (structure)
    - Validating a schema (content)
- Can ontologies benefit from conceptual models?
  - Experience on useful concepts
  - Time, space, context, multirepresentation
- Ontologies can also rely on existing DBMS for
  - Scalability, query languages...

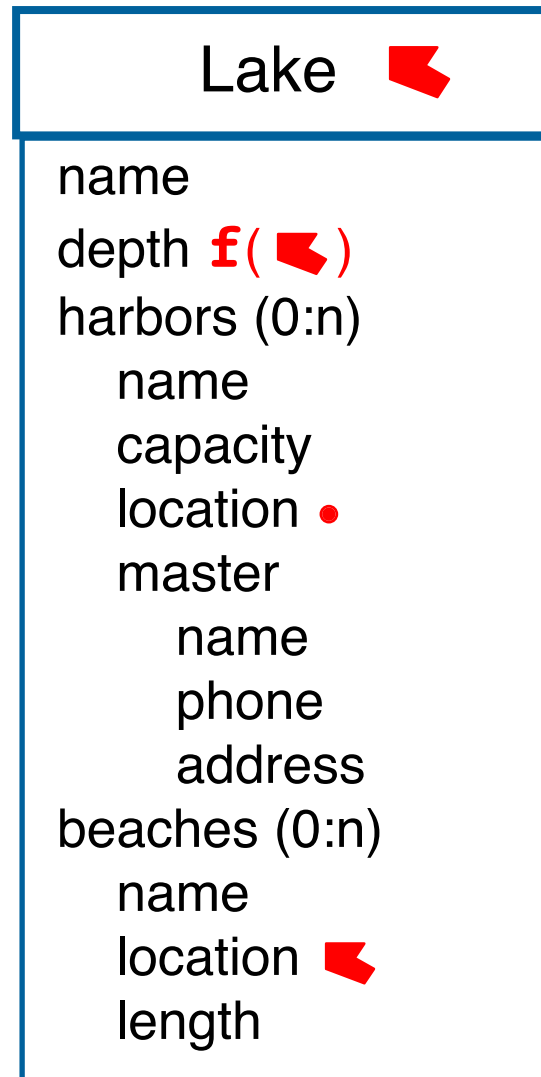
# Some Ontology Approaches

- Formalisms and tools:
  - Wordnet
  - Description Logics (OWL, Protégé, Racer, Fact...)
  - Rule languages (F-Logic...)
- KAON2: an ontology and semantic web framework allowing the design and management of ontologies.
  - An infrastructure for managing OWL-DL, SWRL, and F-Logic ontologies (<http://kaon2.semanticweb.org/>)
- DOGMA : an ontology engineering framework based on the ORM (Object-Role-Modeling) conceptual model
  - (<http://www.starlab.vub.ac.be/research/dogma.htm>)

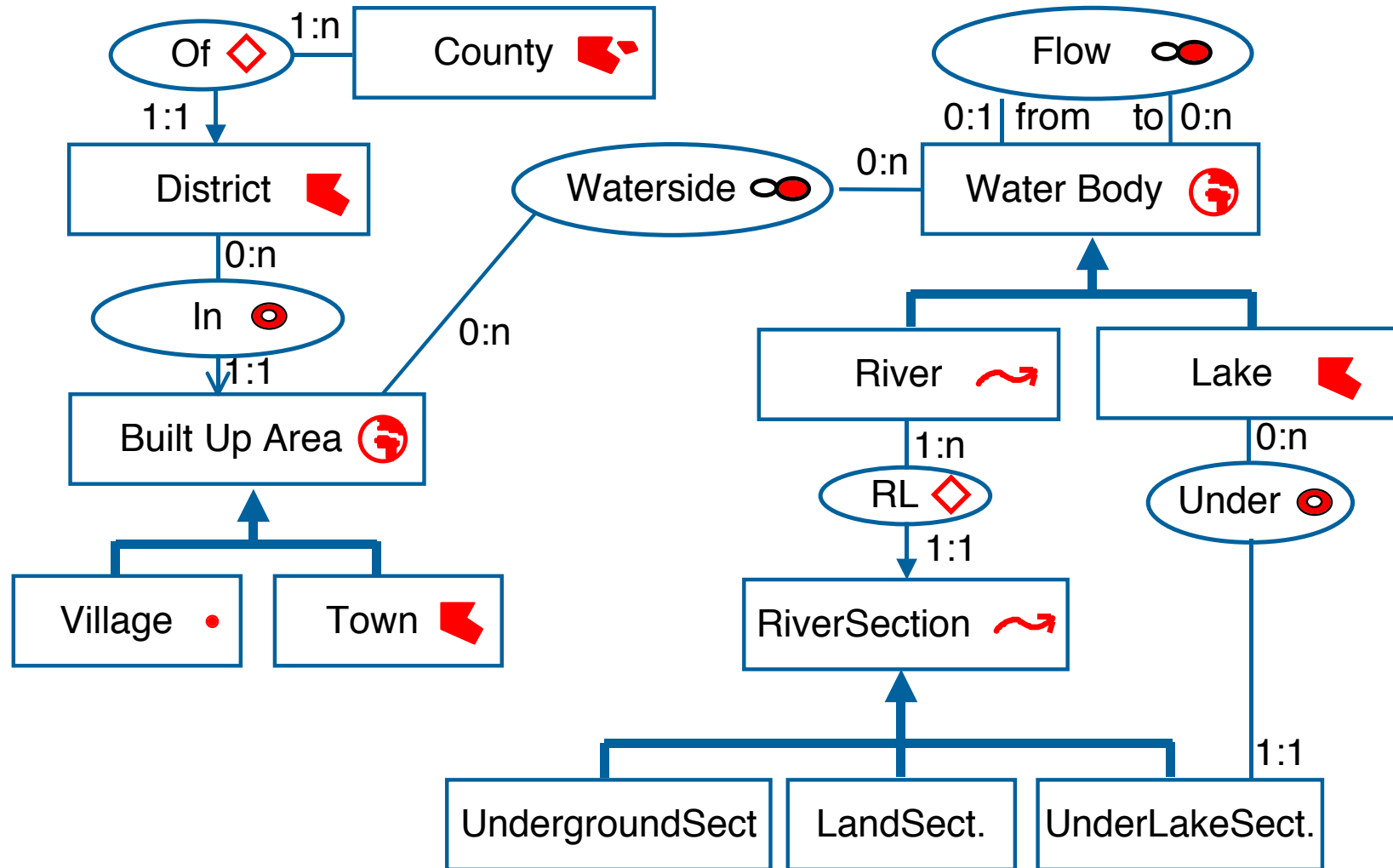
# Conceptual Models

- A representative example: MADS
- A spatio-temporal conceptual model
  - ER data structure extended with
    - Complex objects, n-ary relationships with attributes, generalization hierarchies, multi-instantiation,
    - Spatio-temporal features
    - Contextual features
    - Conceptual languages
      - ◆ Algebraic Query Language
      - ◆ Algebraic Manipulation Language
- A prototype visual schema editor and a visual query editor (MurMur EEC/IST Project)

# A MADDS Complex Object Type



# A MADs Schema



# More on Ontologies

- Ontology : a means to share information and to achieve semantic interoperability between humans and computers (borrowed from Ian Horrocks)
- **“An explicit specification of a conceptualization”** (Gruber)
- It is constituted by :
  - a specific vocabulary used to describe a certain reality, plus
  - a set of explicit assumptions regarding the intended meaning of the vocabulary
- Different kinds of ontologies
  - Taxonomic ontologies
  - Descriptive ontologies
  - Upper level ontologies

# Structure of an Ontology

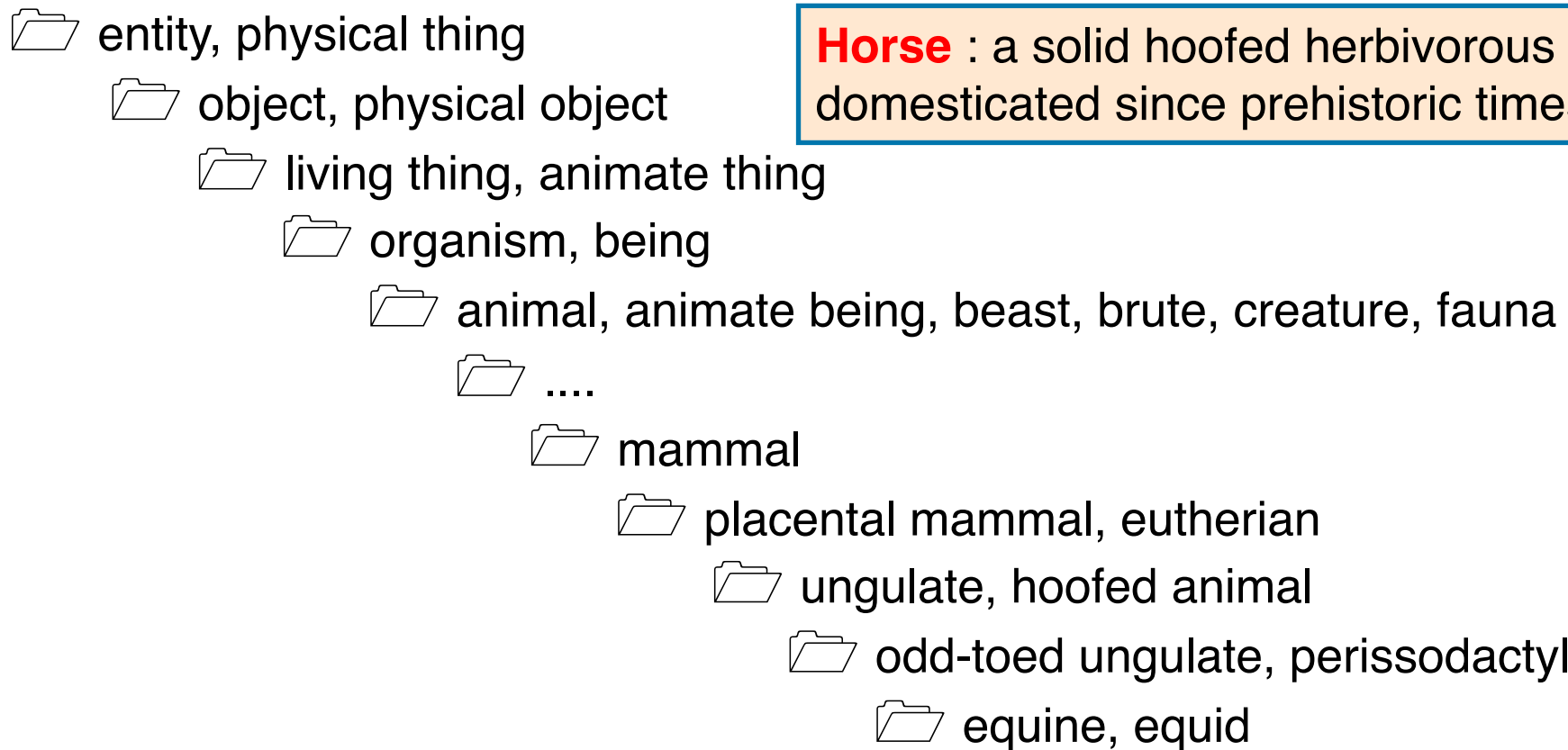
Ontologies typically have distinct components:

- Names for important concepts in the domain (vocabulary)
  - **Animal**, **Elephant**, **AdultElephant**, **Herbivore** ...
- Definitions for the concepts
  - **Elephant** is a concept whose members are a kind of animal
  - **AdultElephant** is a concept whose members are exactly those elephants whose age is greater than 20 years
  - **Herbivore** is a concept whose members are exactly those animals who eat only plants or parts of plants
- Background knowledge / constraints on the domain
  - **AdultElephants** weigh at least 2,000 kg
  - All **Elephants** are either **AfricanElephants** or **IndianElephants**
  - No individual can be both a **Herbivore** and a **Carnivore**



# Taxonomic Ontologies

- Goal : providing a reference **vocabulary**
  - Definition of concepts through **terms**, their hierarchical **organization**, and a few **relationships** such as synonymy, composition, and possibly others



**Horse** : a solid hooped herbivorous quadruped domesticated since prehistoric times

 **horse, Equus, caballus**

**Wordnet**

 saddle horse, riding horse.. 17

# Taxonomic Ontologies usage

- Taxonomic ontologies provide a shared well-defined terminology
- e.g. domain ontologies for groups of companies
  - Electricity companies in Switzerland
  - Land administrations in Switzerland
- Database integration
  - Similarity analysis based on lexical and linguistic techniques
- Database alignment
  - e.g. catalogues in a given market place

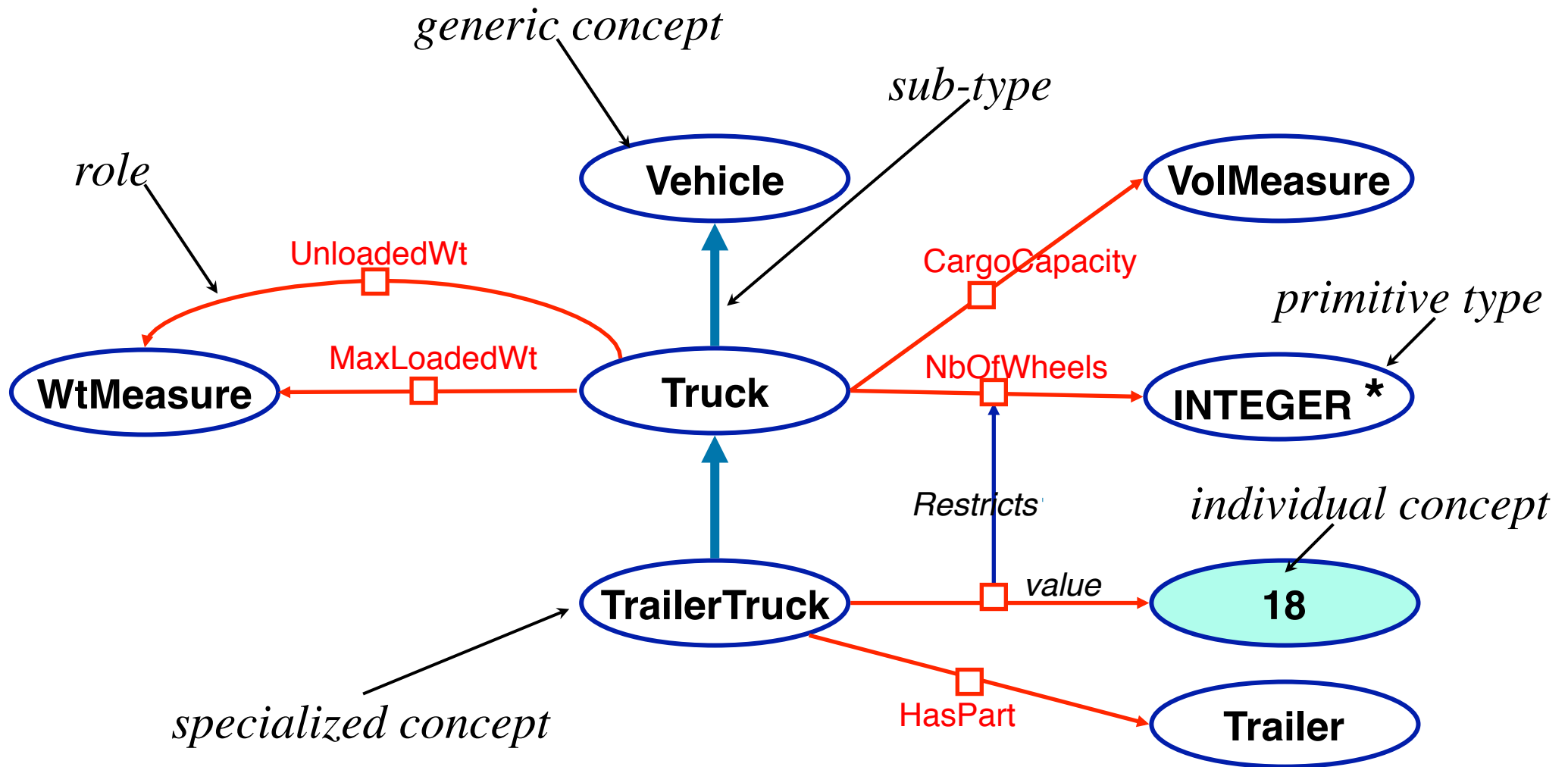
# Descriptive Ontologies

■ **Goal** : providing either information to align existing data structures or patterns to define new specialized ontologies

## ■ **ontologies**

- Definition of **concepts** through data structures and their inter-relationships, plus definition of background knowledge
  - ◆ Horse properties : usage, sex, weight, coat, owner, prizes...
  - ◆ Knowledge : The harem of a wild stallion consist in 1 to 10 mares...
- Terms --> **Concepts**
  - ◆ e.g. **concept H**, denoted by terms: Horse, Equus, caballus
  - ◆ classification of concepts : H is-a E (equine, equid)
  - ◆ relationships between concepts : H hasMare H
  - ◆ properties of concepts : H (sex, weight ...)

# An Example : KL-ONE semantic network



# Disambiguation

- Descriptive ontologies can also provide means to achieve context-based disambiguation
- How do I know how to interpret:
  - « Where do you come from ? »
    - Geneva (the airport I started from) ?
    - Lausanne or Switzerland (the place where I am living) ?
    - France or Switzerland (the country I am a citizen of) ?
    - Milano (the place I was born) ?
    - The place nearby I was 5 minutes ago?

# Upper Level Ontologies (N.Guarino)

- Goal : provide fundamentals for building clean ontologies
  - formal semantics of descriptions (ontologies, schemas)
- Means: defining a small set of basic, domain independent, ontological categories, based on principled philosophical theories :
  - what kinds of things exist
  - what are their basic relationships

# Guarino's Basic Constructs

- Membership Property (MP)
  - equivalent to a unary predicate
  - e.g. “Being an human”
  - Class = in a given world, the set of instances that exhibit this MP
- Characteristics of a MP
  - Rigidity : the MP is always essential to its instances
    - ◆ Being a human : rigid
    - ◆ Being a student : anti-rigid
  - Identity : ability to recognize individual instances of the MP
    - ◆ The same human as a baby, a teenager, an adult...
  - Unity : instances are wholes composed of parts
    - ◆ Being an ocean : unity
    - ◆ Being an amount of water : anti-unity

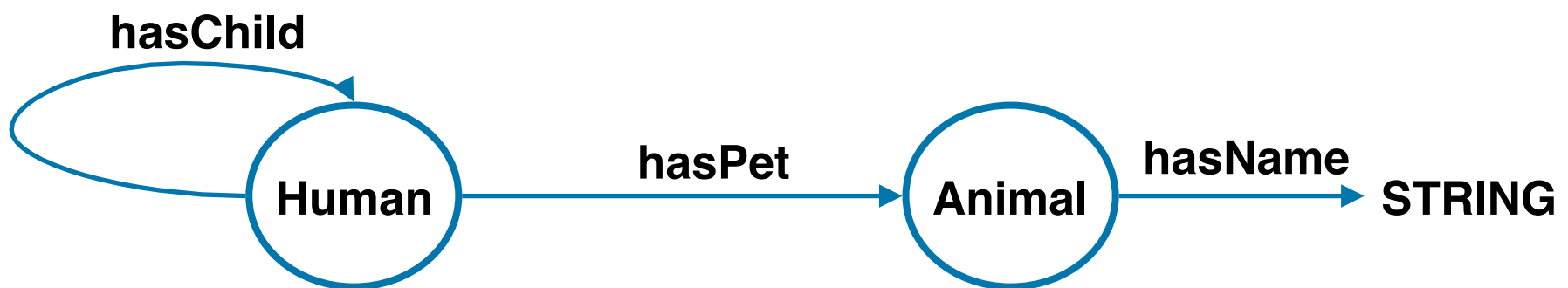
# Logic-based Languages

- Goal :
  - automated reasoning
  - understandable by machines
- Formal : semantics defined by models theory
  - First Order Logic, F-Logic, rules ...
  - **Description Logics (DL)**: decidable subsets of FOL
    - ◆ DAML + OIL, OWL
    - ◆ reasoning tools : Racer, Fact ...
- Degree of formality varies
  - more formal => more automated reasoning  
but less easily understandable by humans
- Description power varies
  - more powerful => decidability is more difficult



# DL : Basic Constructs

- Concept
  - Human, Animal, Elephant
- Role (oriented binary relationship)
  - (cyclic) roles may be symmetric, transitive
  - a role may have an inverse
- Generalization hierarchies : concepts and roles
- Domain of values

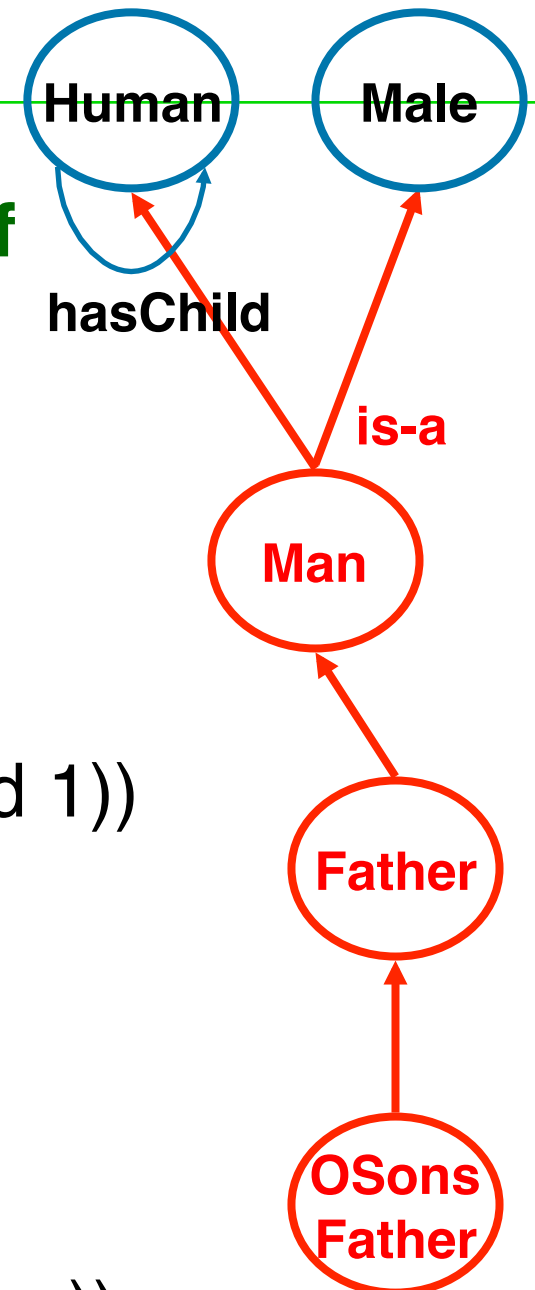


# DL - Logical Formulae

- They allow to define
  - subsumption hierarchies
    - ◆  $\text{PetLover} \subseteq \text{Human}$
  - Constraints on concepts
    - ◆ PetLover has at least one Pet
    - ◆  $\text{PetLover} \subseteq \text{minCardinality}(\text{hasPet } 1)$
  - New concepts defined by a formula
    - ◆  $\text{Man} = \text{IntersectionOf}(\text{Human}, \text{Male})$
- Two kinds of concepts
  - primitive concept
  - defined concept :  
concept = logical\_expression

# DL : Concept Constructors

- **intersectionOf, unionOf, complementOf**  
Man = **intersectionOf** (Human, Male)
- **minimum and maximum cardinality**  
maxCardinality, minCardinality  
Father = **intersectionOf** (Man,  
**minCardinality** (hasChild 1))
- **existential and universal quantifiers**  
allValuesFrom, someValuesFrom  
Father of only sons :  
OSonsFather = **intersectionOf** (Father,  
**allValuesFrom** (hasChild Man))



# Reasoning on the Description

- DL approaches integrate powerful inference mechanisms to reason on the description (Tbox)
  - Classifying concepts and roles
  - Inferring all implicit subsumptions
  - Checking the consistency of the schema
    - ◆ Satisfiability of a concept :  
there must exist at least one world in which the concept has at least one instance
- RACER and FACT are DL reasoning tools

# Consistency Checking

## ■ Satisfiability

■ Parent  $\equiv$  (Person  $\cap$   $\exists$  hasChild.Person)

■ Woman  $\equiv$  (Person  $\cap$  Female)

■ Mother  $\equiv$  (Female  $\cap$  Parent)

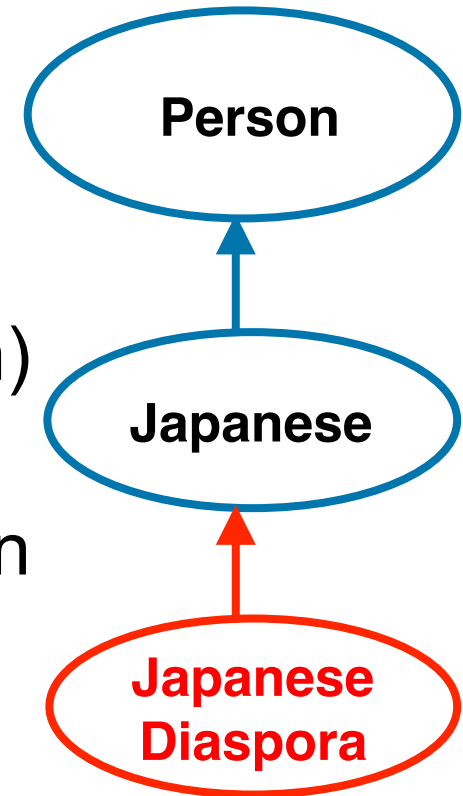
■ GayMother  $\equiv$  ( $\neg$  Woman  $\cap$  Mother)

■ GayMother can never be satisfied (instantiated)

■ A concept C is satisfiable iff there exists an interpretation  $\mathcal{I}$  such that  $C^{\mathcal{I}} \neq \emptyset$   
( $\mathcal{I}$  is called a model of C)

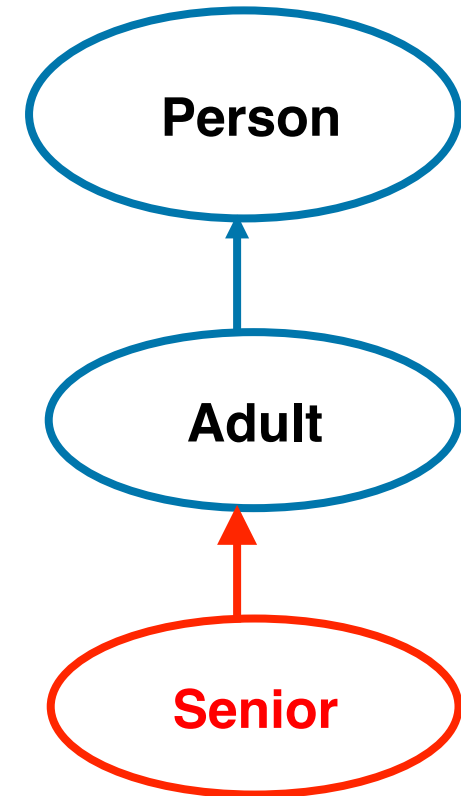
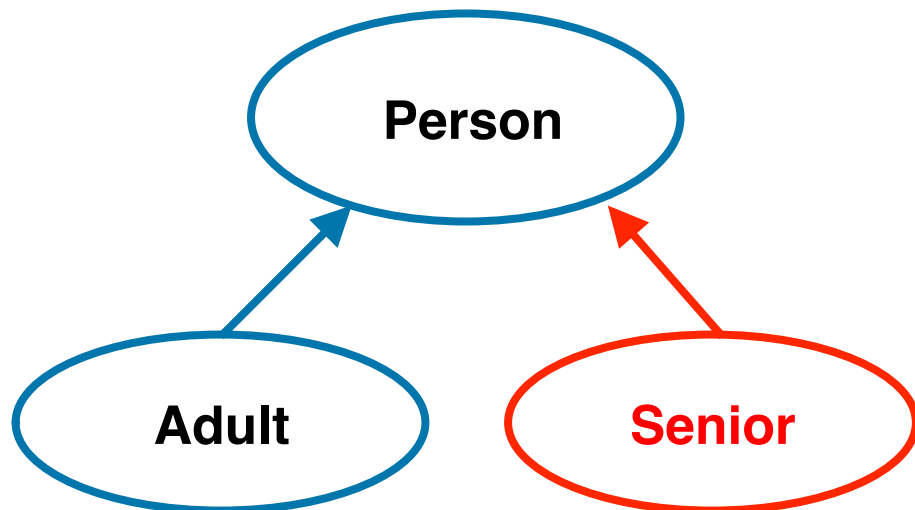
# Subsumption Reasoning

- concept: Person
- role: citizenOf (Person, Country)
- role: livesIn (Person, Country)
- Japanese  
 $\equiv (\text{Person} \cap \exists \text{ citizenOf.Country} = \text{Japan})$
- JapaneseDiaspora  
 $\equiv (\text{Person} \cap (\exists \text{ citizenOf.Country} = \text{Japan} \wedge \exists \text{ livesIn.Country} \neq \text{Japan}))$
- Also at the instance level:  
**Hitori: JapaneseDiaspora**  
**=> Hitori: Japanese**



# Subsumption Reasoning Limitation

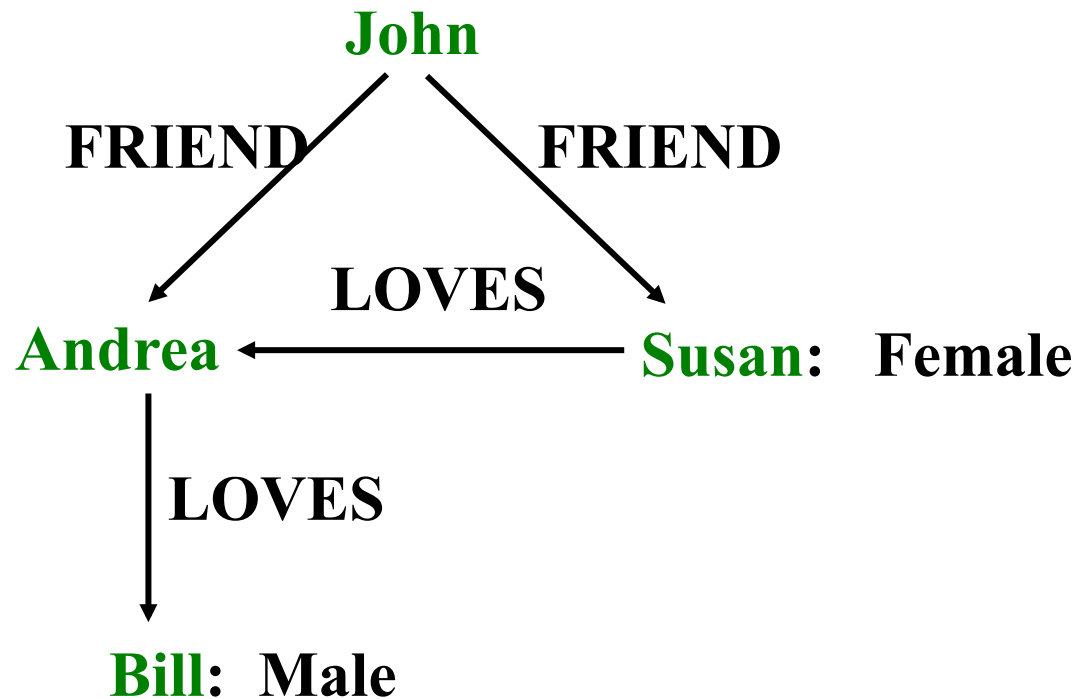
- concept: Person
- role: hasAge (Person, Integer)
- Adult  $\equiv$  (Person  $\cap$   $\exists$  hasAge > 30)
- Senior  $\equiv$  (Person  $\cap$   $\exists$  hasAge > 60)



**NOT POSSIBLE in some DL**

# Inference: Case Reasoning

- Does John have a female friend loving a male person?



**Male**  $\equiv$  **Human**  $\cap$  ( $\neg$  **Female**)



# Satisfiability - Example

- FishingCottage = intersectionOf ( Cottage, SomeValuesFrom (is-touching River) )
- MosquitoFreeForest = intersectionOf ( Forest, AllValuesFrom (is-connected ComplementOf River) )
- ParadiseCottage = intersectionOf ( FishingCottage, SomeValuesFrom (is-inside Forest), AllValuesFrom (is-inside MosquitoFreeForest))

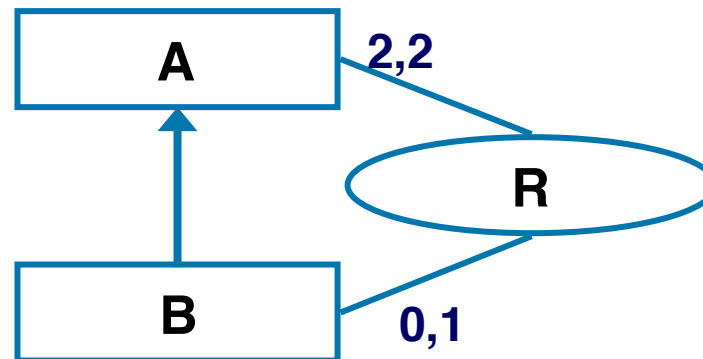
implies :

Paradise cottages do not exist !

# DL and Conceptual Modeling

- DL can be used to check the consistency of the structure of conceptual schemas
  - 1) Translating the schema into DL formulae
  - 2) Checking the satisfiability of each class

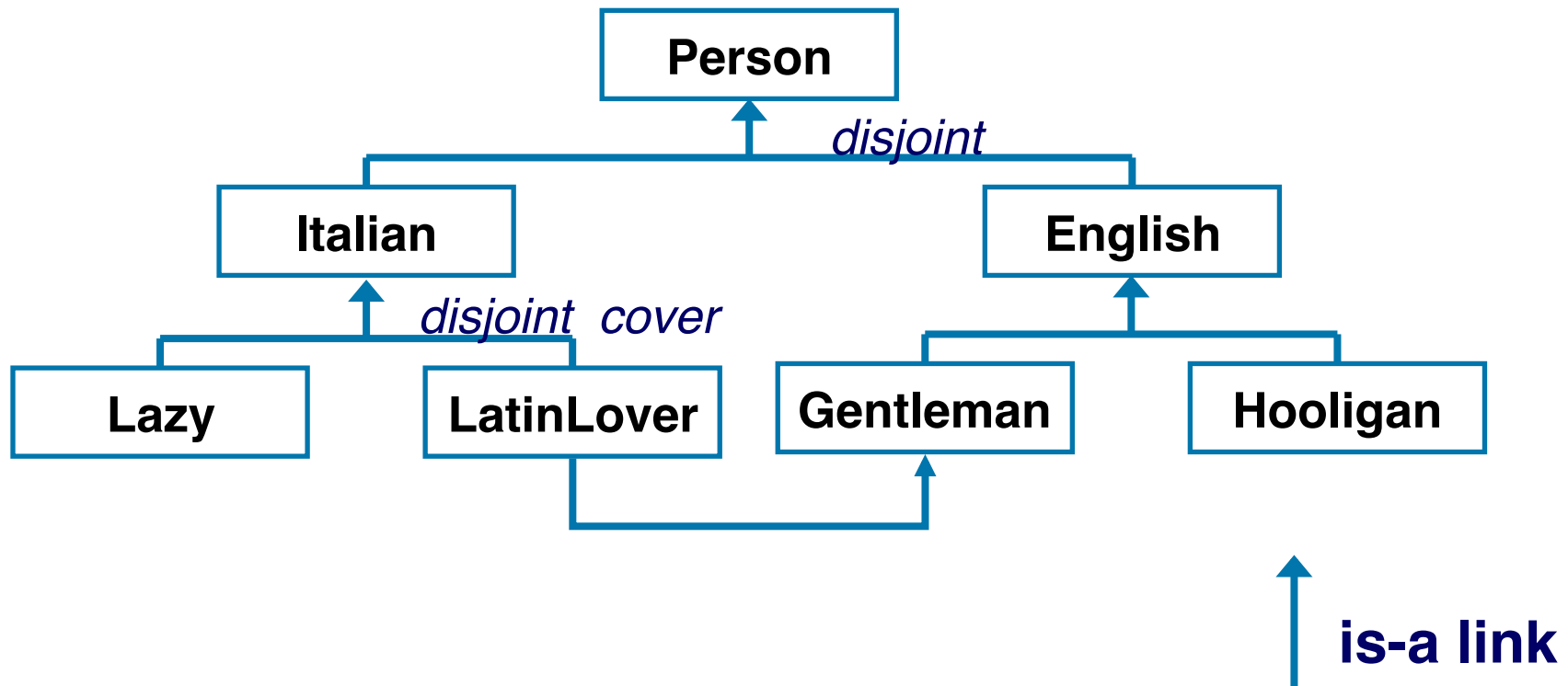
- Example



**implies**

**The two classes A and B contain an infinite number of instances**

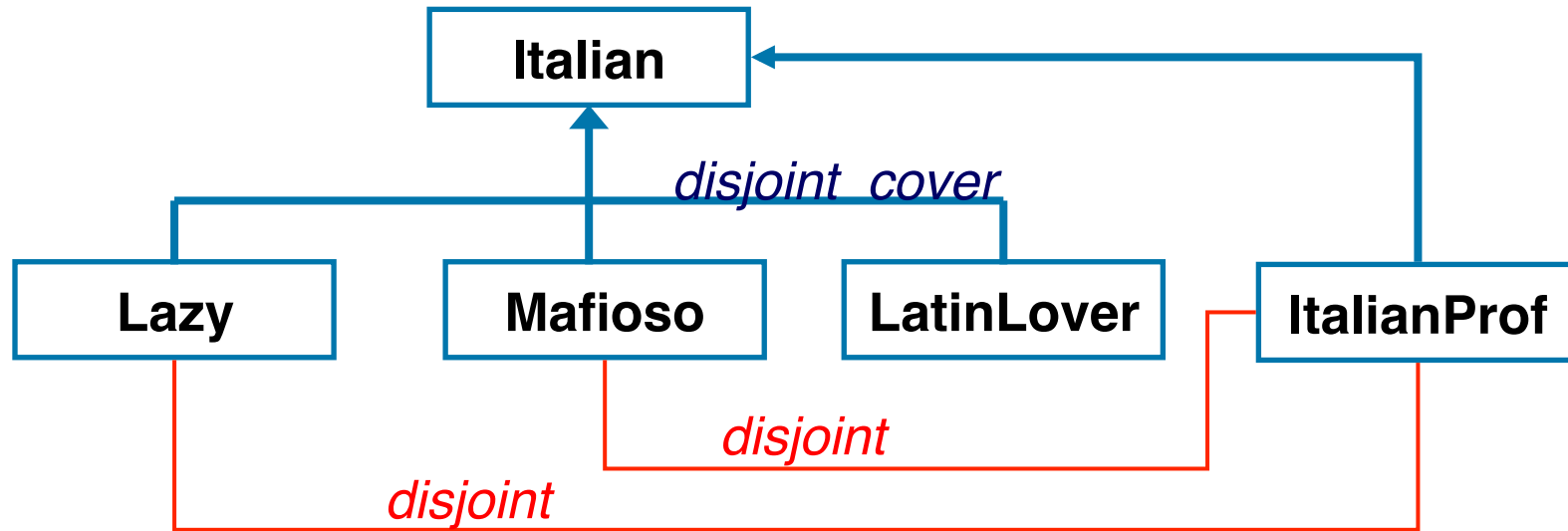
# Class Inconsistency (1/2)



## ■ implies

- LatinLover =  $\emptyset$
- Italian = Lazy

# Class Inconsistency (2/2)

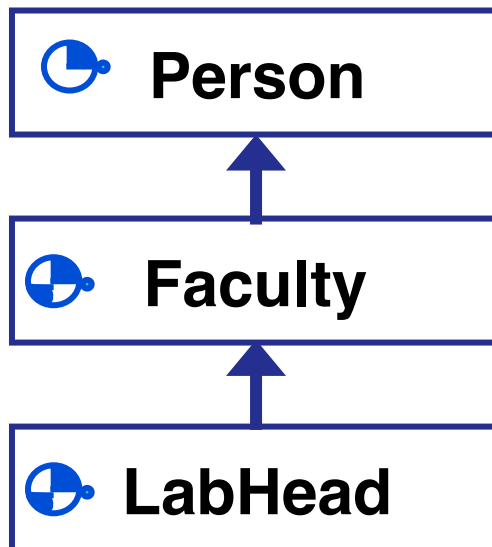
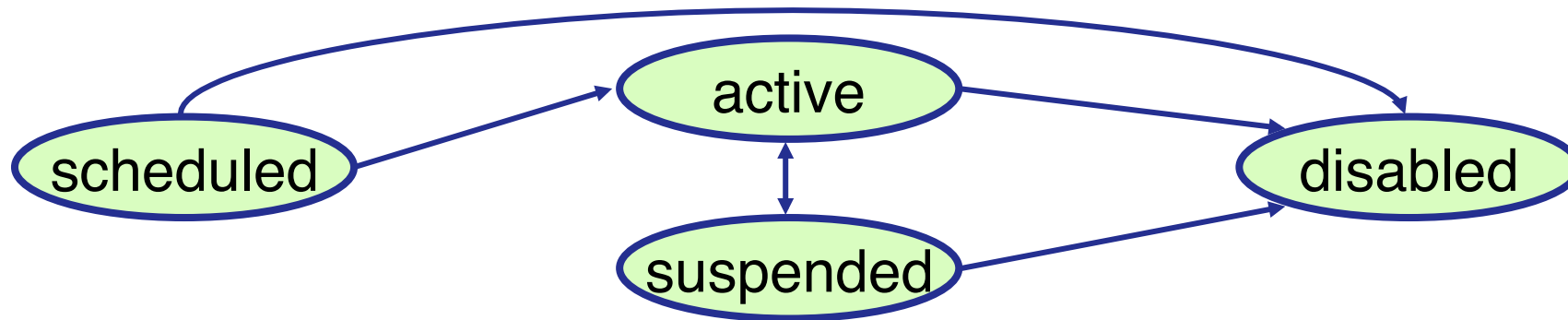


↑ is-a link  
↓ disjoint constraint

- implies
  - ItalianProf  $\subseteq$  LatinLover

# Consistency of CM - Example

- Temporal classes and is-a links
  - Temporal class : active, suspended, deleted, scheduled



IS-A : lifespan inclusion  
or  
activespan inclusion ?

# Strong Features of Logical Approaches

- Inference capabilities
- Consistency checks
- Decentralization
- Easier evolution
- Usability by programs
- Potential for checking CM specifications

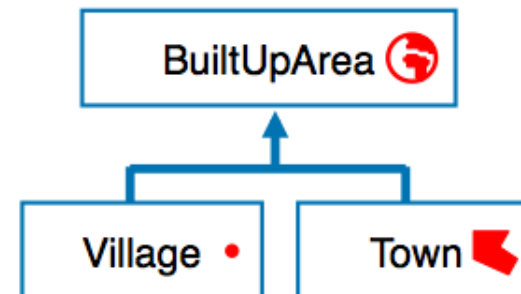
# Strong Features of Conceptual Approaches

- Better understandability of descriptions thanks to support of complex object
- Friendly user interfaces
- Value domains are easily handled

-----

- Desirable (and feasible) extensions of CM approaches include:
  - provide classifying criteria for is-a clusters

BuiltUpArea : Village if area < 2 km<sup>2</sup>  
Town if area ≥ 2 km<sup>2</sup>



- reasoning for inserting in the generalization hierarchy views, result of queries, derived object types

# Some Open Problems

## ■ Inheritance links

- DB currently do not provide a way to express what is the specialization criterion that defines a sub-class: no possibility of automatically positioning an instance in a sub-class

## ■ Multi-instantiation

- Default rules are different:
- DL: by default any two concepts may share instances
- MADS: by default two types do not share instances

## ■ Spatio-temporal information

- DL: very limited support
- MADS: good support: discrete and continuous views, ST relationships



**THANK YOU FOR YOUR ATTENTION**