Reasoning for Ontology Engineering and Usage

1. Introduction, standard reasoning (Ralf Möller)
2. Bottom-up approach for ontology design (Anni-Yasmin Turhan)
3. Understanding and repairing inferences (Matthew Horridge)
4. Data integration through ontologies
   (Diego Calvanese, Giuseppe de Giacomo, Mariano Rodriguez)

The TONES Consortium:
• Free University of Bozen-Bolzano
• Università di Roma "La Sapienza"
• The University of Manchester
• Technische Universität Dresden
• Technische Universität Hamburg-Harburg

http://www.tonesproject.org/

Reasoning for Ontology Engineering and Usage Part 1: Introduction

Ralf Möller
Hamburg University of Technology

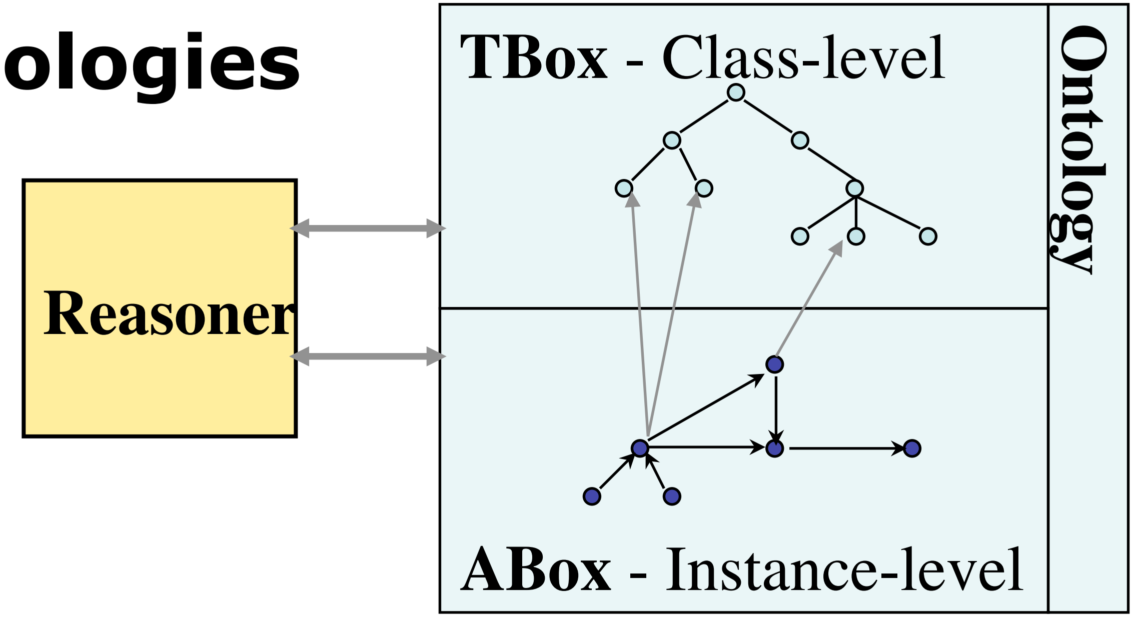
The TONES Consortium:
• Free University of Bozen-Bolzano
• Università di Roma "La Sapienza"
• The University of Manchester
• Technische Universität Dresden
• Technische Universität Hamburg-Harburg

http://www.tonesproject.org/

Terminological knowledge

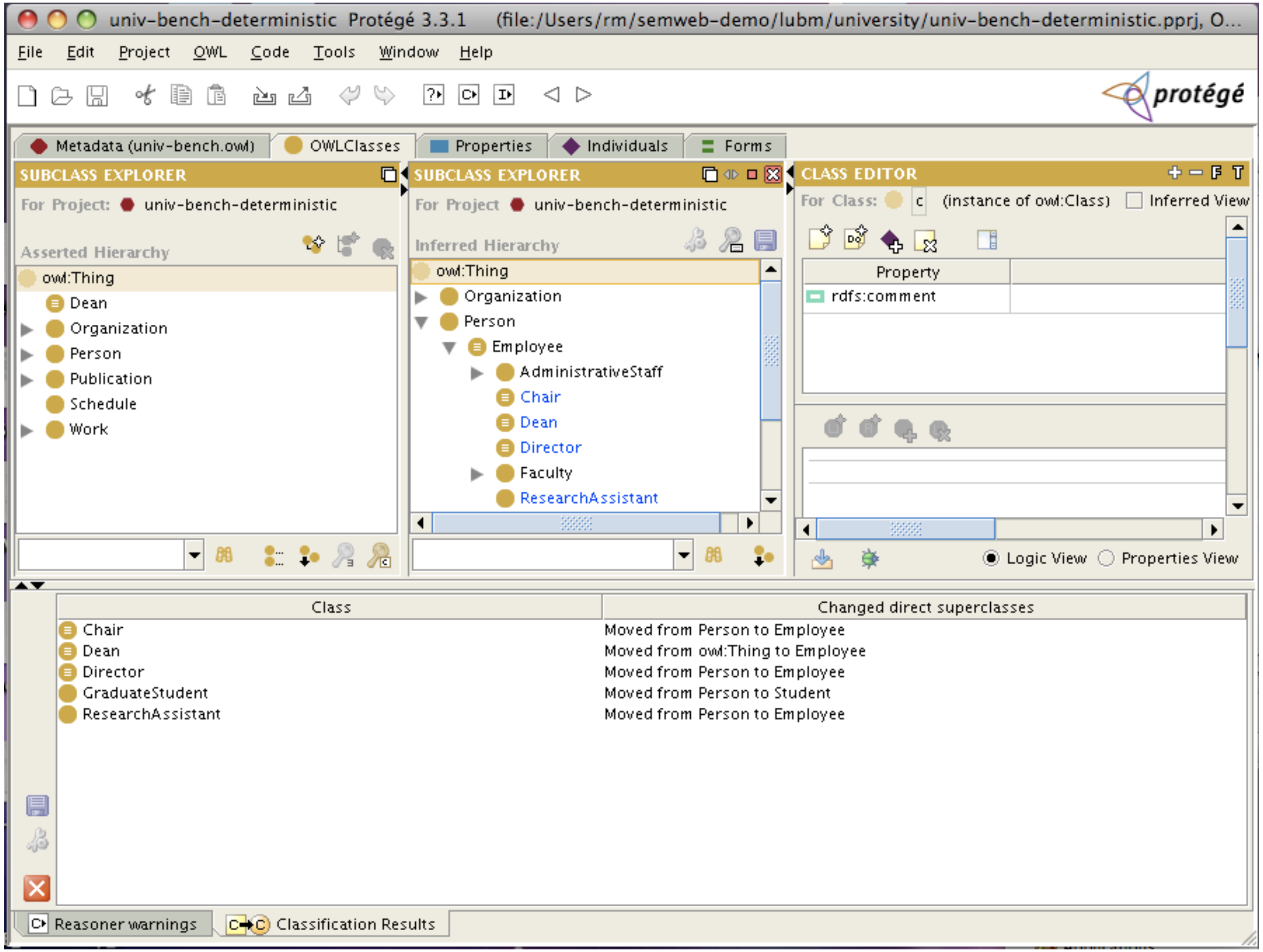
- Represent an application domain in terms of
  - Classes (concept descriptions),
  - Properties (relations, role descriptions), and
  - Objects (instances, individuals)
- First step: select names (define signature)
  - Atomic concept descriptions:
    Student, Professor, Chair, Department, …
  - Atomic role descriptions: *headOf, takesCourse, memberOf*
  - Individuals: *FullProfessor01, Department09, …*
- Use axioms to impose "constraints" (restrictions) on the interpretation of these names
  - A *chair* must be a *person*,
  - *Persons* are no *departments*

Ontologies



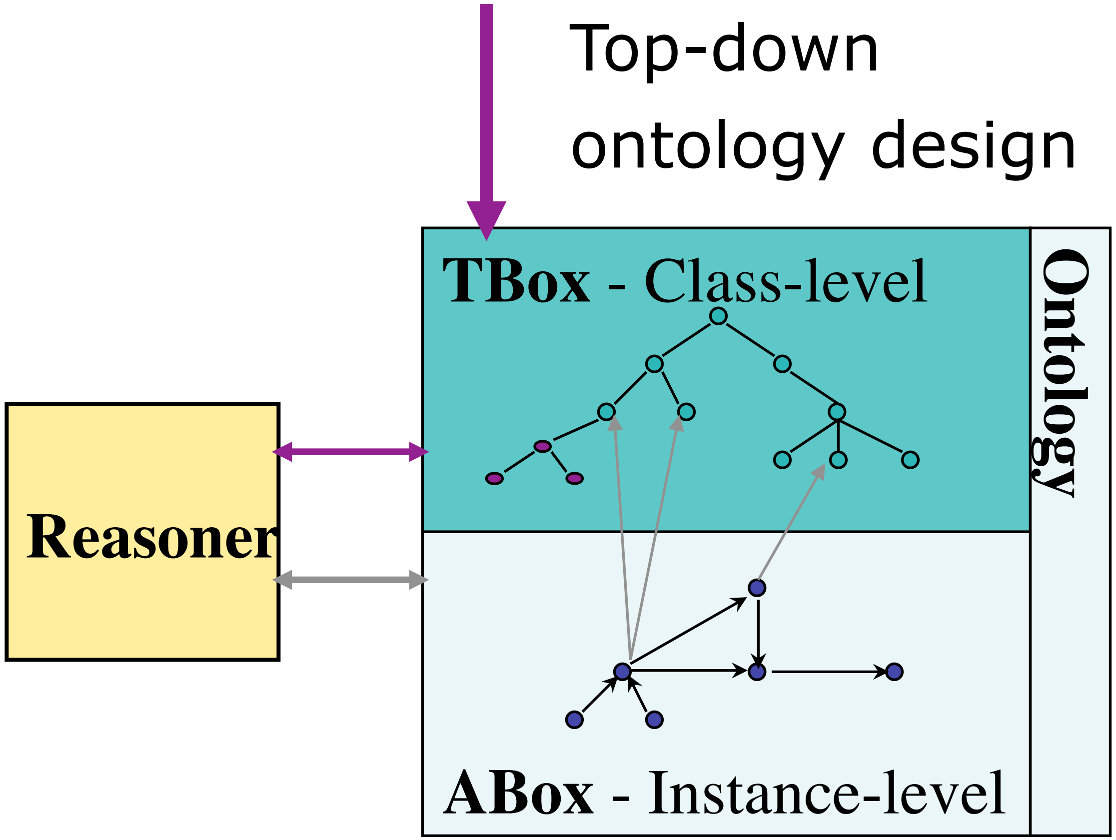
- Standardization of ontology languages
  - DL (abstract syntax), OWL 2 (and various sublanguages)
- Standardization of query and manipulation languages
  - DIG 1.2, OWL API, OWLlink
- Decision problems
  - Does the ontology make sense (satisfiability)
  - Find implicit terminology (e.g. subsumption)
  - Find implicit facts
- Need **reasoning** for solving problems
- Need optimized techniques to achieve scalability

Editing Ontologies



Various approaches for design

- Top-down, bottom-up, reuse, …



Software infrastructure for ontology engineering

- Standard Reasoning:
  - CEL, Fact++, Pellet, RacerPro
- Non-Standard Reasoning:
  - Commonalities, Approximation, Matching, Modularization, Explanation
  - SONIC as a Racer plugin, Pellet Extensions
- Ontology-based data access
  - RacerPro, QuOnto
- User Interfaces:
  - RacerPorter (Tutorial Part 1) + SONIC plugin
  - Protégé + Plugins:
    - Racer plugin (Tutorial Part 1)
    - SONIC plugin (Tutorial Part 2)
    - Modularization plugin (Tutorial Part 3)
    - Explanation plugin (Tutorial Part 3)
    - OBDA plugin (Tutorial Part 4)

OWL, a textual ontology language

- Various **syntaxes**
  - OWL/XML syntax
  - RDF/XML syntax
  - Functional syntax(es)

- Need to understand the **semantics**
- Semantics based on **description logics**
  - Abstract syntax

Representative DL: $\mathcal{ALCQ}$

Let A and R be atomic concept and role descriptions, resp.

$\mathcal{ALCQ}$-concept descriptions for *complex concepts* C or D are defined inductively:

$$
\begin{array}{lcll}
C, D & \longrightarrow & A & \mid \text{atomic concept} \\
 & & C \sqcap D & \mid \text{conjunction} \\
 & & C \sqcup D & \mid \text{disjunction} \\
 & & \neg C & \mid \text{negation} \\
 & & \exists R.C & \mid \text{existential restriction} \\
 & & \forall R.C & \mid \text{value restriction} \\
 & & \exists_{\leq n} R.C & \mid \text{qualified minimum restriction} \\
 & & \exists_{\geq n} R.C & \mid \text{qualified maximum restriction}
\end{array}
$$

Example concept descriptions

$\mathsf{FullProfessor}$

$\mathsf{Staff} \sqcup \mathsf{VisitingProfessor}$

$\mathsf{Person} \sqcap \exists\, \mathsf{worksFor} \,.\, \mathsf{Organization}$

$\mathsf{Professor} \sqcap \exists\, \mathsf{headOf} \,.\, \mathsf{Department}$

$\mathsf{Student} \sqcap \forall\, \mathsf{takesCourse} \,.\, \mathsf{GraduateCourse}$

$\mathsf{Article} \sqcap \forall\, \mathsf{publicationAuthor} \,.\, (\mathsf{Student} \sqcup \mathsf{Professor})$

FullProfessor

$\mathsf{Staff} \sqcup \mathsf{VisitingProfessor}$

$\exists$ worksFor . Organization

$\forall$ takesCourse . GraduateCourse

$$\text{Person} \sqcap \exists\, \text{worksFor} \,.\, \text{Organization}$$

$\text{Person} \sqcap \exists\, \text{worksFor} \,.\, \text{Organization}$

Functional style: Manchester Syntax

```
Class: Person and (worksFor some Organization)
```

OWL/XML Syntax

```
<ox:Intersection>
   <ox:OWLClass ox:URI="Person">
   <ox:ObjectSomeValuesFrom>
       <ox:ObjectProperty ox:URI="worksFor"/>
       <ox:OWLClass ox:URI="Organization"/>
   <ox:ObjectSomeValuesFrom/>
<ox:Intersection/>
```

$\mathcal{ALCQ}$ Semantics

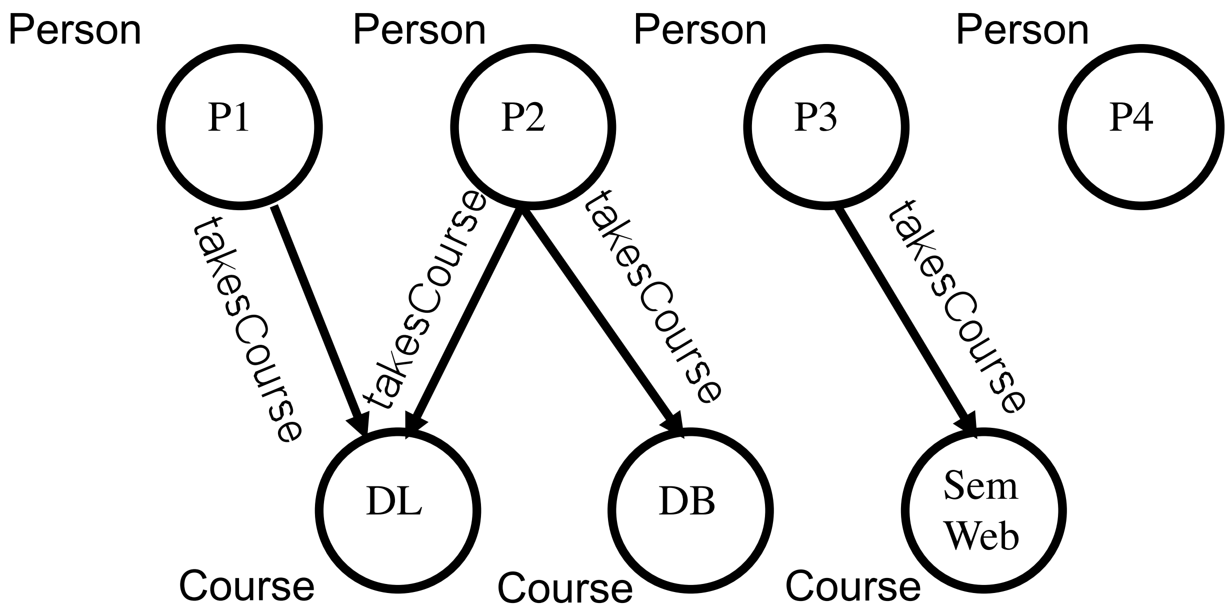
An *interpretation* $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$, where $\Delta^{\mathcal{I}}$, a non-empty set, is the domain and $\cdot^{\mathcal{I}}$ is an interpretation function if:

- $A^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$ for every atomic concept descr. A
- $R^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$ for every (atomic) role R

For complex concept descriptions, the interpretation function is extended as follows:

$$
\begin{aligned}
(C \sqcap D)^{\mathcal{I}} &= C^{\mathcal{I}} \cap D^{\mathcal{I}} \\
(C \sqcup D)^{\mathcal{I}} &= C^{\mathcal{I}} \cup D^{\mathcal{I}} \\
(\neg C)^{\mathcal{I}} &= \Delta^{\mathcal{I}} \backslash C^{\mathcal{I}} \\
(\exists R.C)^{\mathcal{I}} &= \{x \mid \exists y.(x,y) \in R^{\mathcal{I}} \text{ and } y \in C^{\mathcal{I}}\} \\
(\forall R.C)^{\mathcal{I}} &= \{x \mid \forall y. \text{ if } (x,y) \in R^{\mathcal{I}} \text{ then } y \in C^{\mathcal{I}}\} \\
(\exists_{\leq n} R.C)^{\mathcal{I}} &= \{x \mid \sharp\{y \mid (x,y) \in R^{\mathcal{I}} \text{ and } y \in C^{\mathcal{I}}\} \leq n\} \\
(\exists_{\geq n} R.C)^{\mathcal{I}} &= \{x \mid \sharp\{y \mid (x,y) \in R^{\mathcal{I}} \text{ and } y \in C^{\mathcal{I}}\} \geq n\}
\end{aligned}
$$

Interpretation of concept descrs



$$Person^{\mathcal{I}} = \{\text{P1, P2, P3, P4}\}$$

$$Course^{\mathcal{I}} = \{\text{DL, DB, SemWeb}\}$$

$$takesCourse^{\mathcal{I}} = \{(\text{P1, DL}), (\text{P2, DL}), (\text{P2, DB}), (\text{P3, SemWeb})\}$$

$$(Person \sqcap \exists takesCourse.Course)^{\mathcal{I}} = \{\text{P1, P2, P3}\}$$

$$(Person \sqcap \exists_{\geq 2} takesCourse.Course)^{\mathcal{I}} = \{\text{P2}\}$$

Satisfiability of concept descriptions

An interpretation $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ *satisfies* a concept description C if $C^{\mathcal{I}} \neq \emptyset$. In this case, $\mathcal{I}$ is called a *model* for C.

Abbreviations:

$$\begin{array}{rcl|l} \top & = & A \sqcup \neg A & \text{top} \\ \bot & = & A \sqcap \neg A & \text{bottom} \end{array}$$

Tbox

A *Tbox* is a set of *generalized concept inclusions*, *GCIs* $C \sqsubseteq D$.

$C \equiv D$ stands for $C \sqsubseteq D$ and $D \sqsubseteq C$.

$$
\begin{array}{lcl}
\mathsf{Professor} & \sqsubseteq & \mathsf{Person} \\
\mathsf{FullProfessor} & \sqsubseteq & \mathsf{Professor} \\
\mathsf{Chair} & \equiv & \mathsf{Person} \sqcap \exists\, \mathsf{headOf} \,.\, \mathsf{Department} \\
\mathsf{Student} & \equiv & \mathsf{Person} \sqcap \exists\, \mathsf{takesCourse} \,.\, \mathsf{Course} \\
\mathsf{UndergraduateStudent} & \sqsubseteq & \mathsf{Student} \\
\mathsf{Department} & \sqsubseteq & \mathsf{Organization}
\end{array}
$$

Model of Tbox, Subsumption

An interpretation $\mathcal{I}$ *satisfies* a GCI $C \sqsubseteq D$ if $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$.

An interpretation is a *model* of a Tbox if it satisfies all GCIs in the TBox.

A concept description C *is subsumed by* a concept description D w.r.t. a Tbox if the GCI $C \sqsubseteq D$ is satisfied in all models of the Tbox. In this case, we also say that D *subsumes* C.

Tbox inference problems

- *Concept satisfiability*:

  A concept C is satisfiable w.r.t a TBox $\mathcal{T}$ if there exist a model $\mathcal{I}$ of $\mathcal{T}$ such that $C^{\mathcal{I}} \neq \emptyset$.

- *TBox satisfiability*:

  A TBox is satisfiable if it admits a model.

- *Concept subsumption*:

  $C \sqsubseteq D$ w.r.t. a TBox $\mathcal{T}$ iff in all models of $\mathcal{T}$ $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$.

$\mathcal{ALCQ}$ as a fragment of FOL

- Concepts = unary predicates
- Roles = binary predicates
- Concept descriptions = FOL-formulae with one free variable
- GCIs = FOL-formulae without free variables (sentences)
- KB = set of sentences

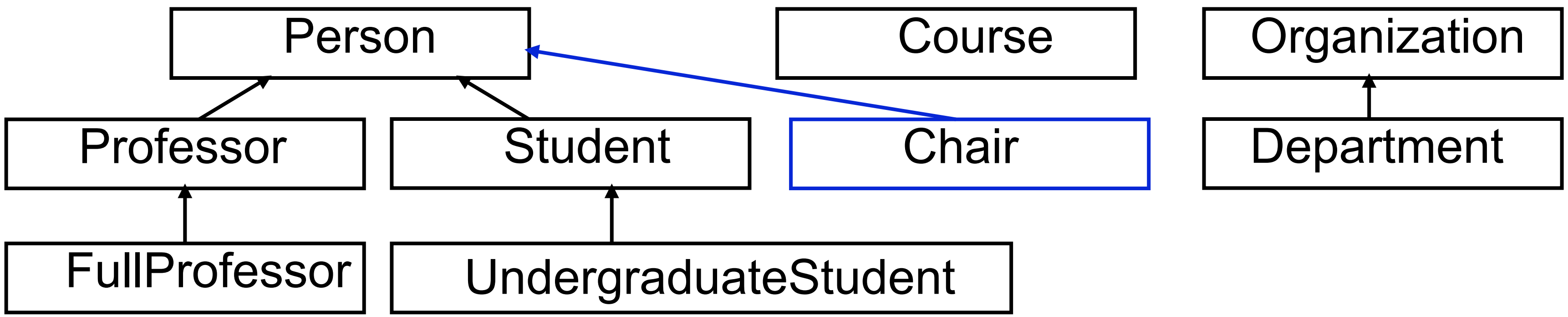
$$\forall\ \mathsf{headOf}\ .\ \mathsf{Department} = \forall\ y\ .\ (\mathrm{headOf}(x, y) \rightarrow \mathrm{Department}(y))$$

$$\exists\ \mathsf{teacherOf}\ .\ \mathsf{Course} = \exists\ y\ .\ (\mathrm{teacherOf}(x, y) \land \mathrm{Course}(y))$$

$$\mathsf{Chair} \sqsubseteq \mathsf{Person} = \forall\ x\ .\ \mathrm{Chair}(x) \rightarrow \mathrm{Person}(x)$$

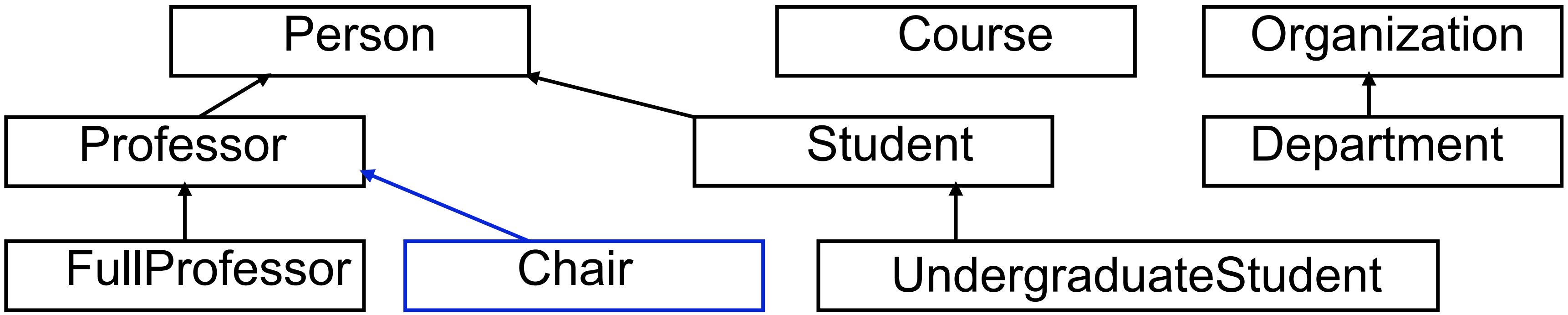
Classification

$$
\begin{array}{lcl}
\mathsf{Professor} & \sqsubseteq & \mathsf{Person} \\
\mathsf{FullProfessor} & \sqsubseteq & \mathsf{Professor} \\
\mathsf{Chair} & \equiv & \mathsf{Person} \sqcap \exists\, \mathsf{headOf}\ .\ \mathsf{Department} \\
\mathsf{Student} & \equiv & \mathsf{Person} \sqcap \exists\, \mathsf{takesCourse}\ .\ \mathsf{Course} \\
\mathsf{UndergraduateStudent} & \sqsubseteq & \mathsf{Student} \\
\mathsf{Department} & \sqsubseteq & \mathsf{Organization}
\end{array}
$$



Classification

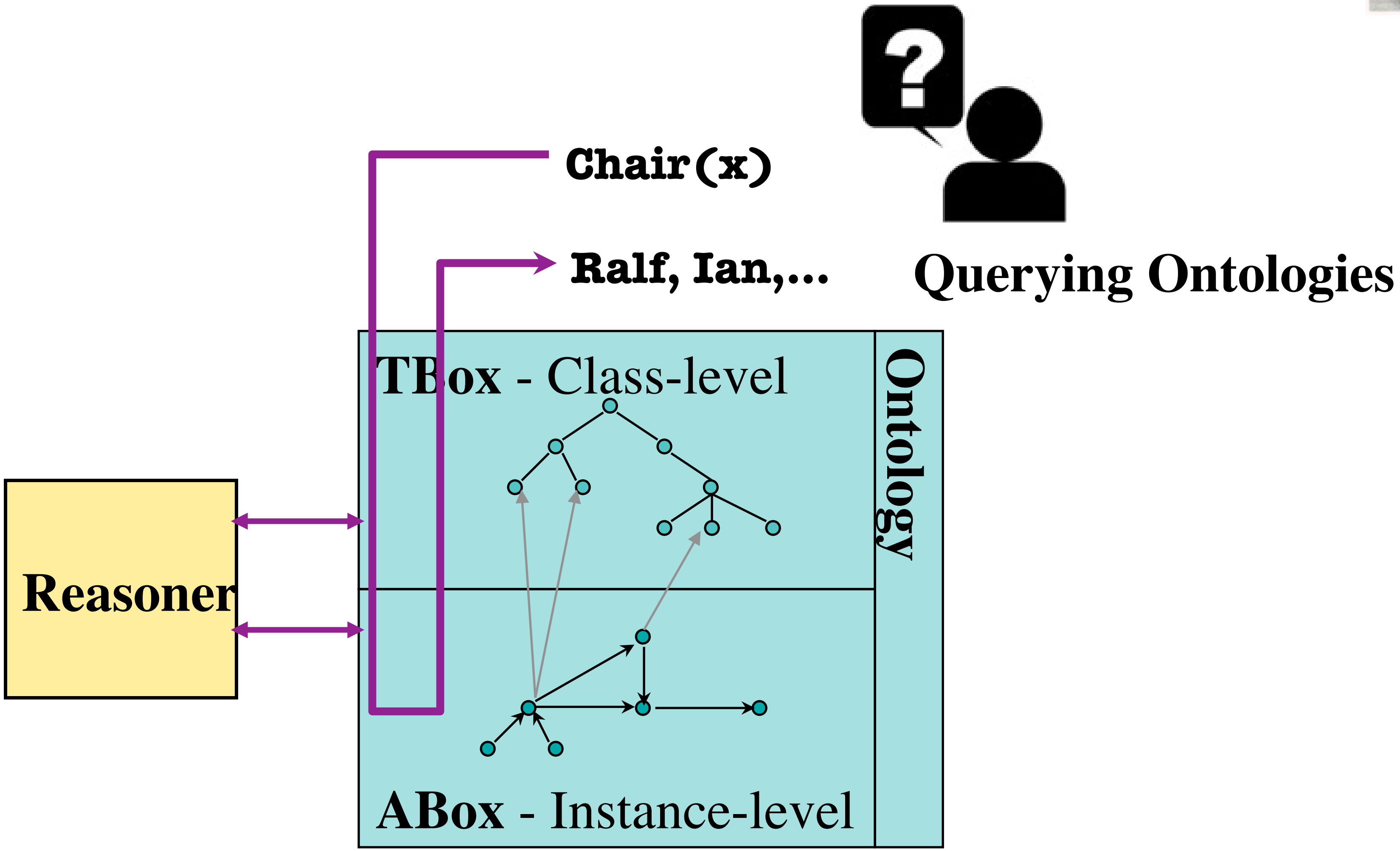
$$
\begin{array}{lcl}
\mathsf{Professor} & \sqsubseteq & \mathsf{Person} \\
\mathsf{FullProfessor} & \sqsubseteq & \mathsf{Professor} \\
\mathsf{Chair} & \equiv & \mathsf{Person} \sqcap \exists\, \mathsf{headOf}\,.\, \mathsf{Department} \\
\mathsf{Student} & \equiv & \mathsf{Person} \sqcap \exists\, \mathsf{takesCourse}\,.\, \mathsf{Course} \\
\mathsf{UndergraduateStudent} & \sqsubseteq & \mathsf{Student} \\
\mathsf{Department} & \sqsubseteq & \mathsf{Organization} \\
\exists\, \mathsf{headOf}\,.\, \top & \sqsubseteq & \mathsf{Professor}
\end{array}
$$



Demo

- RacerPro and RacerPorter

What about individuals?



Abox

An *Abox* is a set of assertions.

Assertions are of the form

$$C(i) \text{ (concept assertion) or}$$
$$R(i, j) \text{ (role assertion)}$$

where C is a concept description, R is a role description, and i, j are individuals.

Example

Professor(Ralf)

Course(Course01)

Department(Department09)

teacherOf(Ralf, Course01)

memberOf(Ralf, Department09)

Abox consistency

A concept assertion $C(i)$ is satisfied if there exists a model $\mathcal{I}$ of $\mathcal{T}$ such that it holds: $i^{\mathcal{I}} \in C^{\mathcal{I}}$.

A role assertion $R(i, j)$ is satisfied if there exists a model $\mathcal{I}$ of $\mathcal{T}$ such that it holds: $(i^{\mathcal{I}}, j^{\mathcal{I}}) \in R^{\mathcal{I}}$.

An interpretation satisfying all assertions in an Abox $\mathcal{A}$ is called a model for $\mathcal{A}$.

An Abox $\mathcal{A}$ is called *consistent* if such a model exists, it is called *inconsistent* otherwise.

Abox inference problems

- *ABox consistency*:

  An ABox $\mathcal{A}$ is consistent w.r.t. a TBox $\mathcal{T}$ iff it has a model that is also a model of $\mathcal{T}$.

- *Instance test* or *instance problem*:

  An individual i an instance of C w.r.t. a TBox $\mathcal{T}$ and an ABox $\mathcal{A}$ iff in all models of $\mathcal{T}$ and $\mathcal{A}$ $i^{\mathcal{I}} \in C^{\mathcal{I}}$.

- *Instance retrieval*:

  W.r.t. the query concept C, a TBox and an ABox find all individuals i mentioned in an ABox such that i is an instance of C.

Ontology usage

- Example: curriculum design

Tbox: $\{ReasearchProfessor \sqsubseteq \exists_{\leq 1}\ teacherOf.Course$
$Seminar \sqsubseteq Course, Lecture \sqsubseteq Course$
$Seminar \sqsubseteq \neg Lecture\}$

Abox: $\{ResearchProfessor(bob), Lecture(dl), teacherOf(bob,\ dl),$
$Seminar(sem1), teacherOf(bob,\ sem1)\}$

Unique name assumption

- Different individuals are mapped to different domain objects
- Example:

Tbox: $\{Professor \sqsubseteq \exists_{\leq 1}\, headOf.Department\}$

Abox: $\{Professor(ian),\ Department(cs),\ headOf(ian,\ cs),$
$Department(comp\_s), headOf(ian,\ comp\_s\}$

Example

- Find eager students

$\mathsf{q} = \{(?x) \mid EagerStudent(?x)\}$

Tbox: $\{EagerStudent \equiv Student \sqcap \exists_{\geq 3}\, enrolledIn.Course\}$

Open world assumption

- If something cannot be proven, it is not concluded that the negation holds
- Example: find lazy professors

$$\mathsf{q} = \{(?x) \mid LazyProfessor(?x)\}$$

$$\text{Tbox: } \{LazyProfessor \equiv Professor \sqcap \exists_{\leq 1}\, teacherOf.Course\}$$

$$\text{Abox: } \{Professor(ralf),\ Course(db),\ teacherOf(ralf,\ db)\}$$

$$\text{Answer: } []$$

- Epistemic aspects in query languages required (e.g., nRQL)

Query Answering

$$q = \{(?x, ?y) \mid Chair(?x), headOf(?x, ?y)\}$$

Tbox: $\{Chair \equiv Person \sqcap \exists headOf.Department\}$

Abox: $\{Person(p),\ Department(d),\ headOf(p,\ d)\}$

Answer: $[?x \leftarrow p, ?y \leftarrow d]$

Query answering w.r.t. ontologies

- Incomplete information, but need the certain answers
- Expressivity of different query languages
  - **Grounded conjunctive queries** plus **additions**
    - In principle:
      reduction to instance tests (standard service)
    - But: non-trivial optimization techniques required (e.g., query execution plans)
    - Efficient QA in practical applications for expr. DLs
      - Aggregation operators and
      - Server-side processing of query results with optimization
      - Practical implementation as part of **RacerPro**

State of the Art

Expressive DLs, answering GCQs

- 5 years ago: 100 Individuals
- 3 years ago: 1000 Individuals
- Now: 10000 individuals, interactive queries, up to 100000 depending on the expressivity used in the Tbox
- Note that we talk about sound and complete reasoning