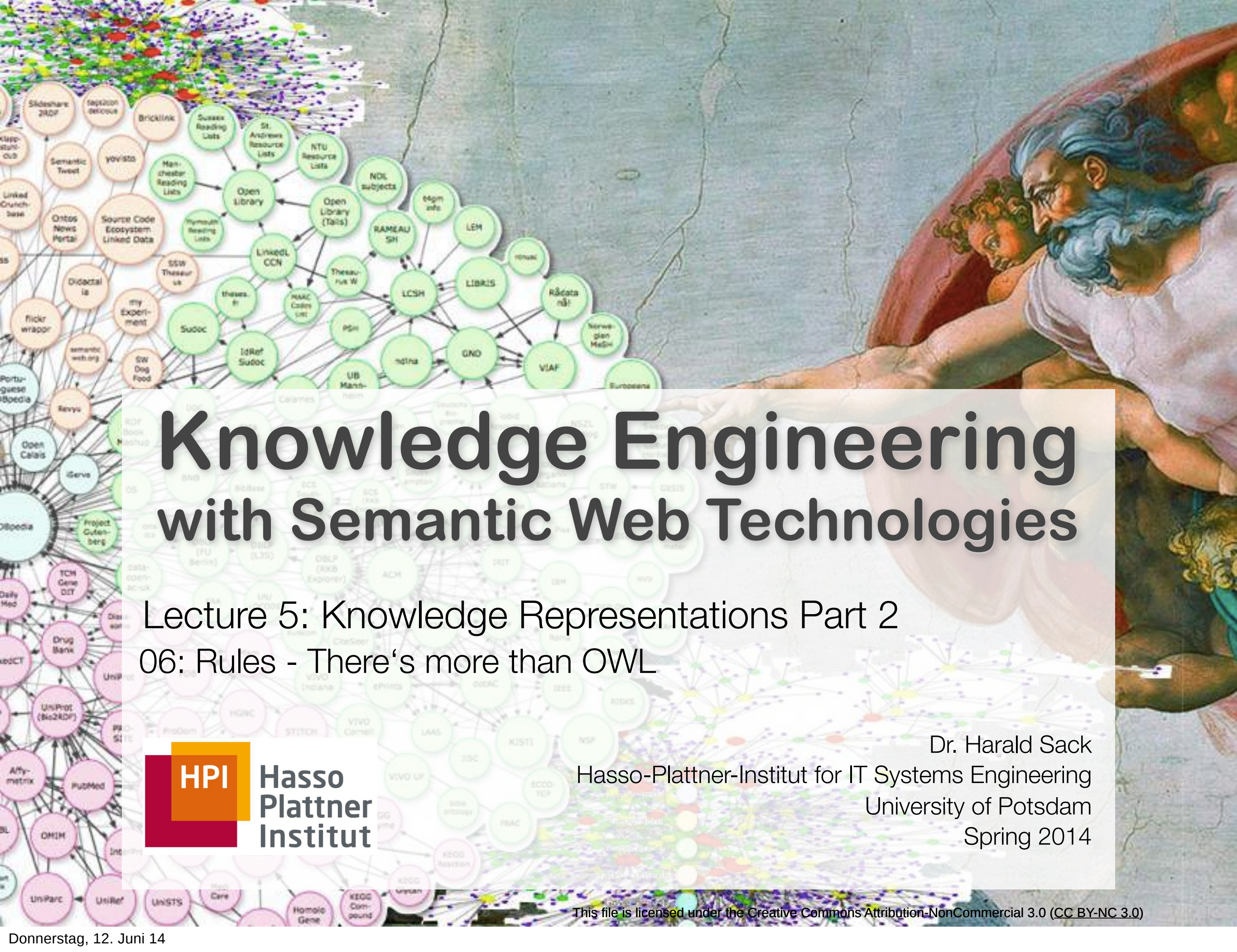




Knowledge Engineering with Semantic Web Technologies

Lecture 5: Knowledge Representations Part 2
06: Rules - There's more than OWL



Dr. Harald Sack
Hasso-Plattner-Institut for IT Systems Engineering
University of Potsdam
Spring 2014



Lecture 5: Knowledge Representations II

Open HPI Course: Knowledge Engineering with Semantic Web Technologies

short recap:

OWL



$SHOIN(\mathcal{D})$

OWL2



$SHROIQ(\mathcal{D})$

OWL Ontologies

complex classes

property restrictions

general property inclusion

OWL profiles

Extension

OWL 2 Overview

Class Expressions

- Class names A, B
- Conjunction $C \sqcap D$
- Disjunction $C \sqcup D$
- Negation $\neg C$
- Exist. property restriction $\exists R.C$
- Univ property restriction $\forall R.C$
- Self $\exists S.\text{Self}$
- Greater-than $\geq n S.C$
- Less-than $\leq n S.C$
- Enumerated classes $\{a\}$

Properties

- Property names R, S, T
- Simple properties S, T
- Inverse properties R^{-}
- Universal property U

Tbox (Class axioms)

- Inclusion $C \sqsubseteq D$
- Equivalence $C \equiv D$

Rbox (Property Axioms)

- Inclusion $R_1 \sqsubseteq R_2$
- General Inclusion $R^{(-)}_1 \circ R^{(-)}_2 \circ \dots \circ R^{(-)}_n \sqsubseteq R$
- Transitivity
- Symmetry
- Reflexivity
- Irreflexivity
- Disjointness

Abox (Facts)

- Class membership $C(a)$
- Property relation $R(a, b)$
- Negated property relation $\neg S(a, b)$
- Equality $a = b$
- Inequality $a \neq b$

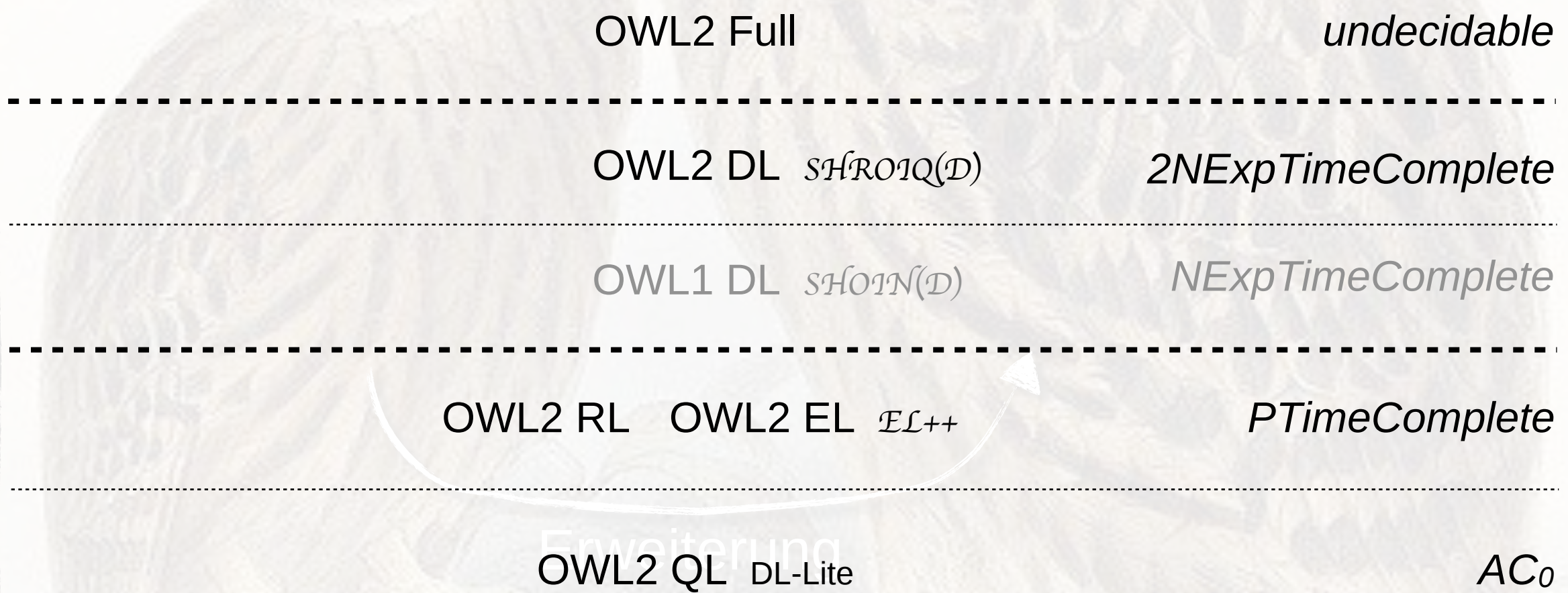
OWL 2

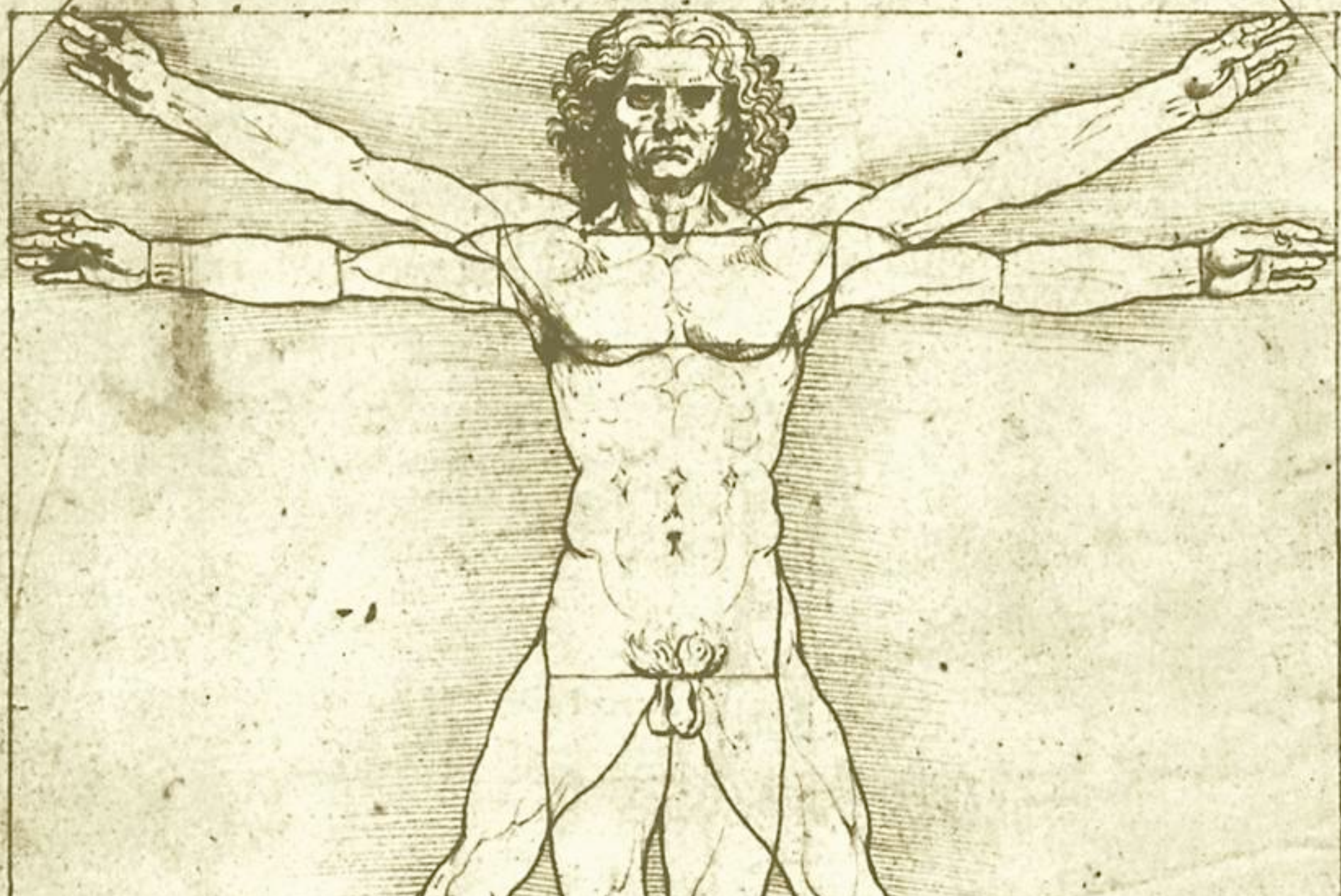
$\mathcal{SHROIQ}(\mathcal{D})$

Erweiterung

OWL – Web Ontology Language

- OWL is a semantic fragment of FOL
- OWL exists in different flavors
- $\text{OWL2 EL, OWL2 RL, OWL2 QL} \subseteq \text{OWL2 DL} \subseteq \text{OWL2 Full}$





06 Rules - There's more than OWL

Open HPI - Course: Knowledge Engineering with Semantic Web Technologies
Lecture 5: Knowledge Representations II

- The **Semantic Web** concentrates on **declarative** forms of knowledge representation
 - OWL, RDFS
- Rules are a common form of **procedural** knowledge representation in **Knowledge Engineering**
 - Expert Systems
 - Prolog, CLIPS, JESS, OPS, ...
- Knowledge representation formalisms of the Semantic Web have **expressive limitations** which can be overcome by rule-based knowledge
 - e.g. composition of complex classes from classes and properties

What are Rules?

IF A THEN B

A → B

- Interpretation of a rule depends on context
 - General Inference:
Premise → Conclusion
 - Hypothesis:
Cause → Effect
 - Production:
Condition → Action

- **Logical Rules** (FOL implication):
 - $F \rightarrow G$ is equivalent with $\neg F \vee G$
 - Logical extension of the KB (static)
 - Open World, declarative
- **Procedural Rules** (e.g. Production Rules):
 - If X then Y else Z
 - executable machine instructions (dynamic)
 - operational (semantics = effect at application)
- **Logic Programming Rules** (e.g. Prolog, F-Logic):
 - „ $\text{woman}(X) \leftarrow \text{person}(X) \text{ AND NOT } \text{man}(X)$ “
 - Approximation of logical semantics with operational aspects
 - Closed World (mostly), semi-declarative

- Rules as FOL implications (**Horn Clause**)

$$\underbrace{A_1 \wedge A_2 \wedge \dots \wedge A_n}_{\text{Body}} \rightarrow \underbrace{H}_{\text{Head}}$$

- semantically equivalent with

$$\neg A_1 \vee \neg A_2 \vee \dots \vee \neg A_n \vee H$$

- where A_i , H are atomic formulas
- Quantification most times omitted, free variables are considered to be universally quantified
 - i.e. the rule holds for all possible assignments

- Rules as FOL implications (*Horn Clause*)

$$\begin{array}{ccc} \underbrace{H}_{\text{Head}} & \leftarrow & \underbrace{A_1 \wedge A_2 \wedge \dots \wedge A_n}_{\text{Body}} \end{array}$$

often written from right to left ($\leftarrow$ or $:-$)

- semantically equivalent with

$$\neg A_1 \vee \neg A_2 \vee \dots \vee \neg A_n \vee H$$

- where A_i , H are atomic formulas
- Quantification most times omitted, free variables are considered to be universally quantified
 - i.e. the rule holds for all possible assignments

- **Lloyd-Topor Transformation**

- Several atoms in the head are usually considered as **conjunction**

$$A_1 \ A_2 \ . \ . \ . A_n \ \rightarrow \ H_1 \ H_2 \ . \ . \ . H_m$$

- is equal to

$$A_1 \ A_2 \ . \ . \ . A_n \ \rightarrow \ H_1$$

$$A_1 \ A_2 \ . \ . \ . A_n \ \rightarrow \ H_2$$

. . .

$$A_1 \ A_2 \ . \ . \ . A_n \ \rightarrow \ H_m$$

- this transformation is called **Lloyd-Topor Transformation**

Variants of FOL Rules

• Disjunctive Rules

- Disjunction of several non-negated Atoms

$$\underbrace{A_1 \wedge A_2 \wedge \dots \wedge A_n}_{\text{Body}} \rightarrow \underbrace{H_1 \vee H_2 \vee \dots \vee H_m}_{\text{Head}}$$

- reverse implication, as e.g.

„if I see something, then the light is on or the sun is shining“

14

- FOL Rules

- **Clause:**

Disjunction of atomic formulas
or negated atomic formulas

- **Horn Clause:**

Clause with **at most one**
not negated atom

$\neg p \vee \neg q \vee \dots \vee \neg t \vee u$ can be written as
 $p \wedge q \wedge \dots \wedge t \rightarrow u$

- **Definite Clause:**

Clause with **exactly one**
not negated atom

- **Fact:**

Clause **of a single**
not negated atom

• Examples

$\text{Person}(x) \rightarrow \text{Woman}(x) \vee \text{Man}(x)$ (clause)

$\text{Man}(x) \wedge \text{hasChild}(x, y) \rightarrow \text{Father}(x)$ (definite clause)

$\text{hasBrother}(\text{mother}(x), y) \rightarrow \text{isUncle}(x, y)$
(with function symbol)

$\text{Man}(x) \wedge \text{Woman}(x) \rightarrow$ (horn clause)

$\text{Woman}(\text{Anja})$ (fact)

• Semantics of rules complies to FOL semantics

Description Logics vs. Rules

- Rules are usually considered to apply only to **known** constants.
- No possibility to „create“ new things „on the flight“ by using existential quantification $\exists$

Human $\sqsubseteq \exists \text{hasParent} . \text{Human}$

- If rules are considered FOL formulas, then combining rules with $\mathcal{ALC}$ leads to undecidability.
- What about decidable FOL-Rules....?

 **DATALOG**

- is a **logical rule language** that consists of
 - *horn clauses **without** function symbols*
 - *conjunction, constants, universally quantified variables, predicate symbols*
 - *no disjunction, no negation, no existential quantification, no function symbols*
- originally developed as foundation of deductive databases
- *Knowledge Bases* (Datalog Programs) are sets of horn clauses (without function symbols)
- **DATALOG** is decidable and computationally efficient, **ExpTime**

- **DATALOG Term:** constant **c** or variable **v**
- **DATALOG Atom:** $p(t_1, \dots, t_n)$
with predicate p ,
terms $t_1, \dots, t_n$
- **DATALOG Rule:** $\forall x_1 \dots \forall x_n (B_1 \wedge \dots \wedge B_n \rightarrow H)$
with $B_1, \dots, B_n, H$ atoms and
 $x_1, \dots, x_n$ variables
- **DATALOG Program:** set of DATALOG rules

19

(1) $\text{Vegetarian}(x) \wedge \text{FishProduct}(y) \rightarrow \text{dislikes}(x, y)$

(2) $\text{orderedDish}(x, y) \wedge \text{dislikes}(x, y) \rightarrow \text{Unhappy}(x)$

(3) $\text{orderedDish}(x, y) \rightarrow \text{Dish}(y)$

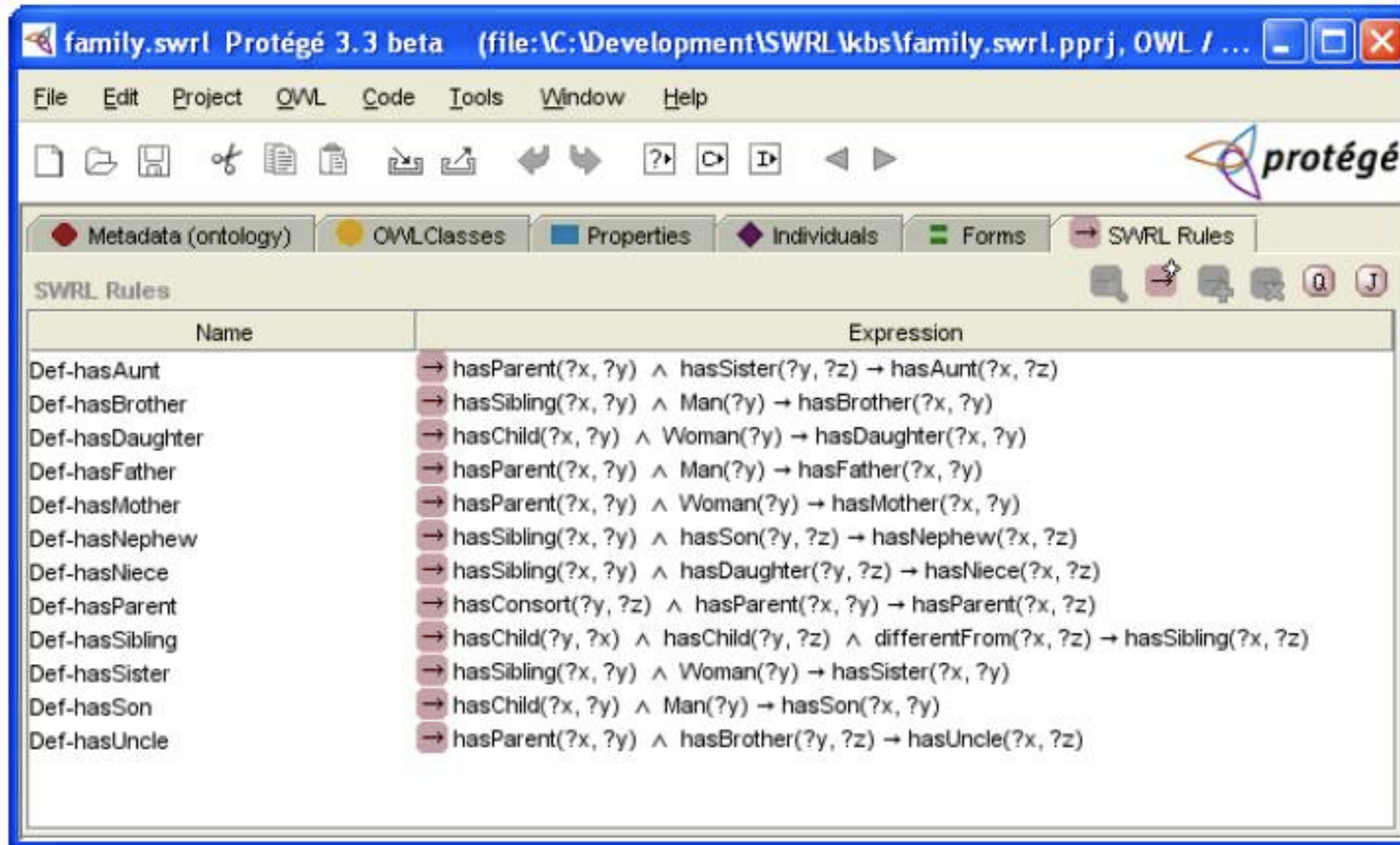
(4) $\text{dislikes}(x, z) \wedge \text{Dish}(y) \wedge \text{contains}(y, z) \rightarrow \text{dislikes}(x, y)$

(5) $\rightarrow \text{Vegetarian}(\text{Matthias})$

(6) $\text{Happy}(x) \wedge \text{Unhappy}(x) \rightarrow$

- **DATALOG** Rules allow mixing classes and relations (i.e. unary and binary predicates)
- therefore it can be **more expressive than DL**
- A combination of **DATALOG** and OWL is the **SWRL Language**
(*not subject of this course*)

SWRL - Semantic Web Rule Language



- based on combination of parts of **OWL** and **RuleML/Datalog**
- **Idea:** Datalog Rules that apply on OWL ontologies
 - Symbols in rules can be OWL identifiers (or new Datalog identifiers)
- SWRL is undecidable



Lecture 6: Knowledge Engineering

Open HPI Course: Knowledge Engineering with Semantic Web Technologies