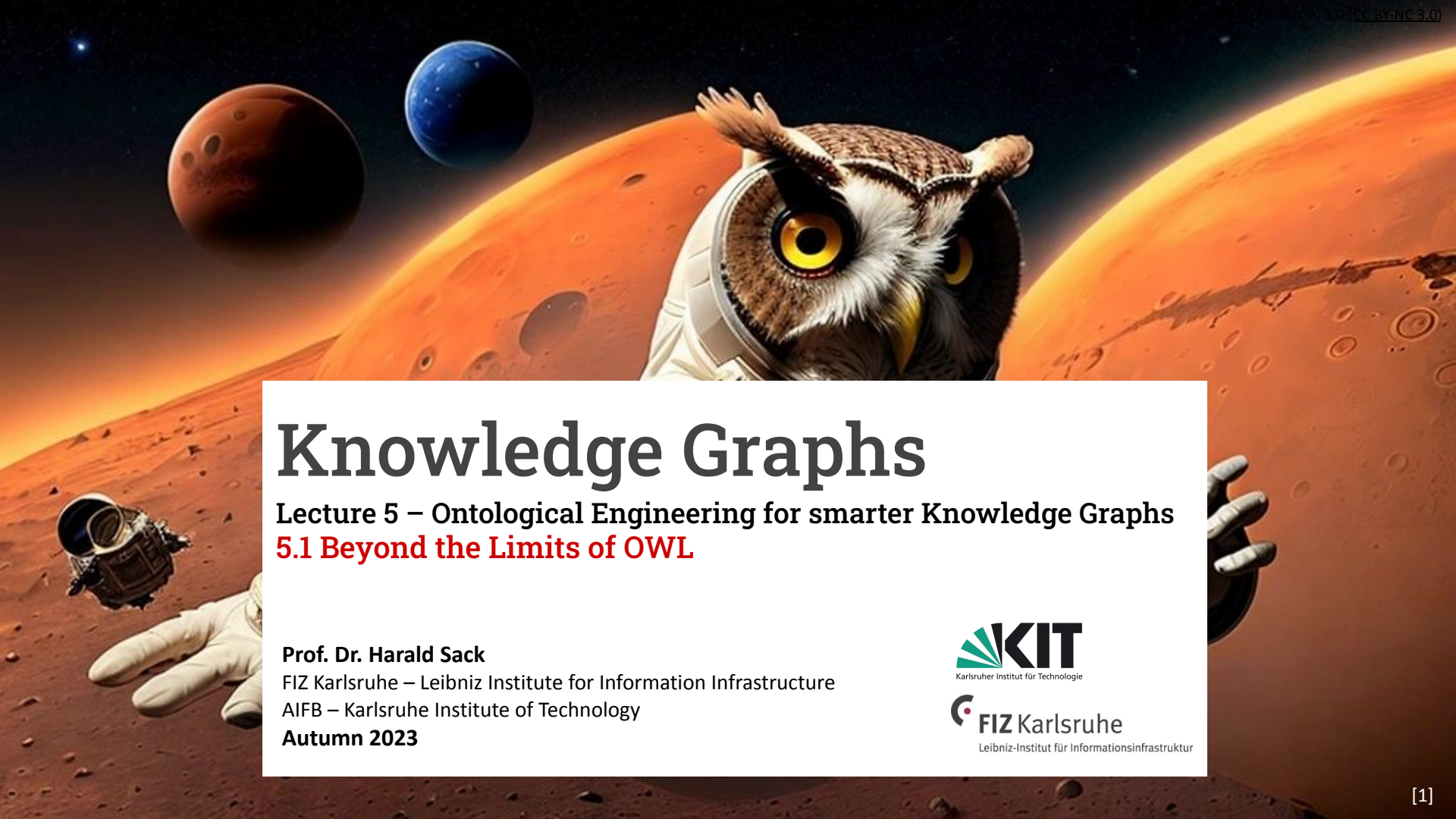


Knowledge Graphs

Lecture 5 – Ontological Engineering for smarter Knowledge Graphs

5.1 Beyond the Limits of OWL

Prof. Dr. Harald Sack
FIZ Karlsruhe – Leibniz Institute for Information Infrastructure
AIFB – Karlsruhe Institute of Technology
Autumn 2023

Karlsruher Institut für Technologie

FIZ Karlsruhe
Leibniz-Institut für Informationsinfrastruktur

Knowledge Graphs

Lecture 5: Ontological Engineering for smarter Knowledge Graphs

Karlsruher Institut für Technologie

FIZ Karlsruhe
Leibniz Institute for Information Infrastructure

5.1 Beyond the Limits of OWL

Excursion 7: The Semantic Web Rule Language SWRL

5.2 How to design your own Ontology

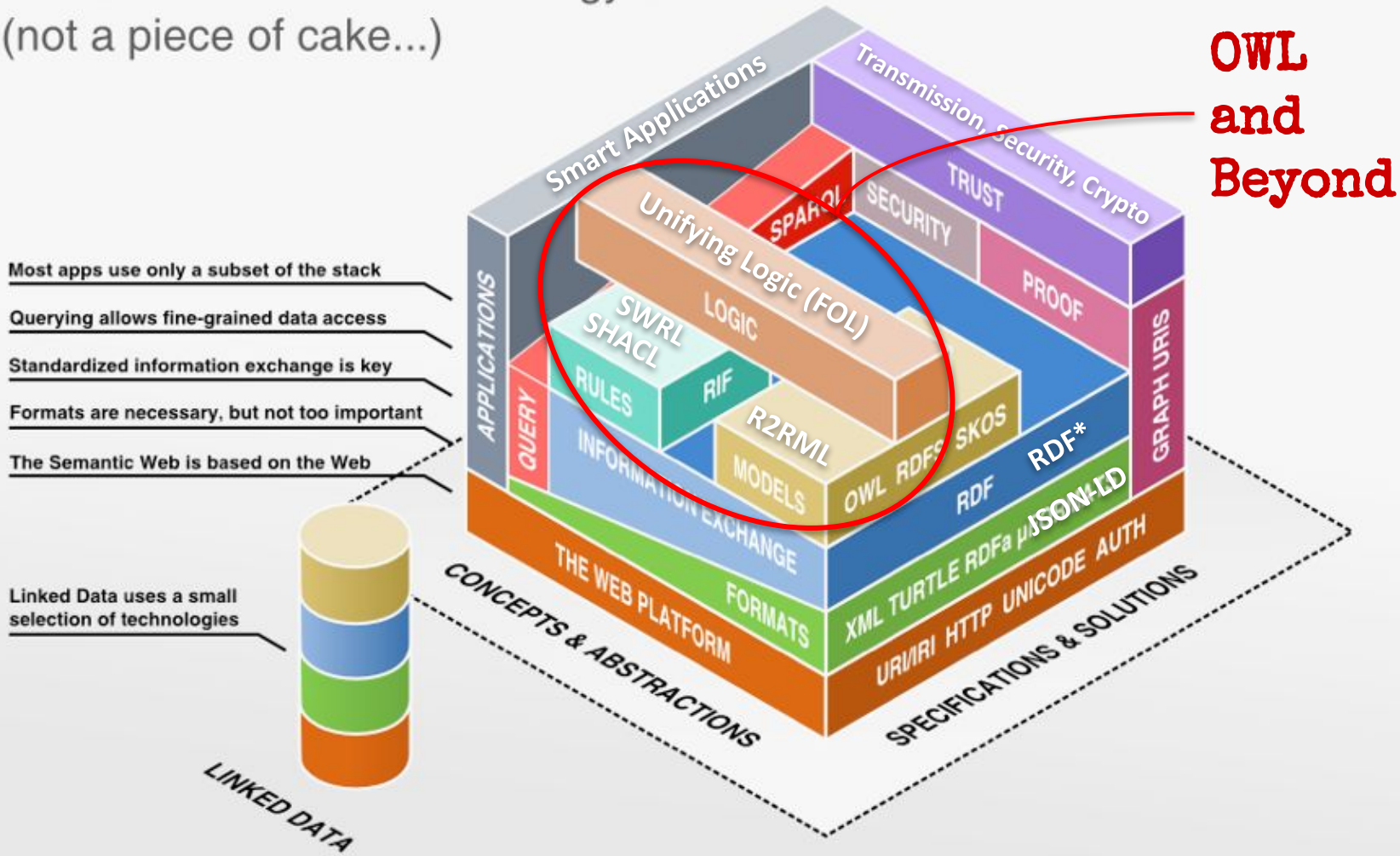
5.3 How to design better Ontologies

5.4 Ontological Engineering

5.5 Knowledge Graph Construction

5.6 Ontologies & Knowledge Graphs – Best Practices

The Semantic Web Technology Stack
(not a piece of cake...)



[2]

Beyond OWL

- Consider the following example:

  "A younger sibling of x is a sibling that is younger than x."

  - `:Paul :youngerThan :Mary .`
  - `:Paul :siblingOf :Mary .`

  - `youngerSiblingOf a owl:ObjectProperty; ??`

- We need a constructor to create an intersection of Properties.
- **Problem:** OWL does not provide any way to entail a property intersection.

Beyond OWL

- More **expressivity** also means more **complexity**.
  This might lead to **undecidability** (as for FOL).
- Do we really need more expressivity than OWL DL offers?
- Consider the following example:
  "A squanderer is a person whose expenses are higher than their income."
  - $\texttt{Squanderer} \sqsubseteq \texttt{Person}$
  - $\texttt{Squanderer} \sqsubseteq \texttt{hasExpenses}.\top$
  - $\texttt{Squanderer} \sqsubseteq \texttt{hasIncome}.\top$

  ?
- We need a constructor to combine Classes and Properties.
- **Problem:** Mixing of TBox and ABox.

Rules Beyond OWL

- The following example can be expressed via a **FOL-Rule**:

  "A squanderer is a person whose expenses are higher than their income."

$$\begin{aligned} &\texttt{Person(x)} \\ &\quad \wedge \texttt{ hasIncome(x,y)} \\ &\quad \wedge \texttt{ hasExpenses(x,z)} \\ &\quad \wedge \texttt{ (z > y)} \\ &\quad \rightarrow \texttt{ Squanderer(x)} \end{aligned}$$

- Arithmetics can be part of rules and modeled like a predicate:

  $\texttt{(z > y)} \triangleq \texttt{greaterThan(z,y)}$

Rules and the Semantic Web

- The **Semantic Web** focuses on **declarative** forms of knowledge representation
  Description Logics, OWL, RDFS
- **Rules** are a common form of **procedural** knowledge representation in Knowledge Engineering
  Expert Systems, Prolog, etc.
- Rules:

```
IF  A ... THEN B ...
    A     →      B
```

Premise → Conclusion

FOL as Rule Language

- Rules as FOL implications (Horn Clause)

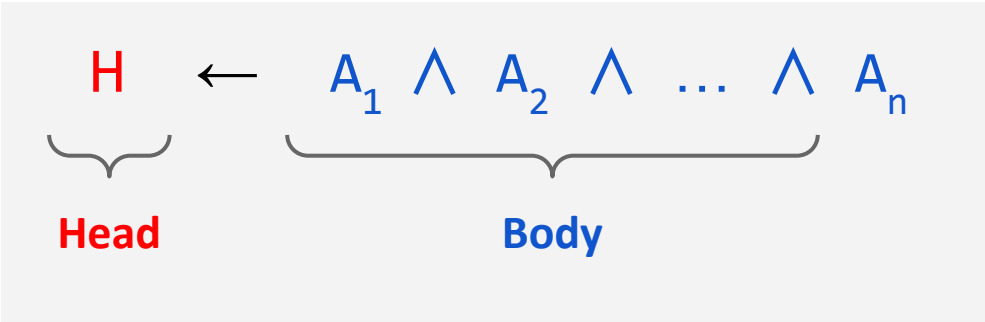
$$\underbrace{A_1 \wedge A_2 \wedge \ldots \wedge A_n}_{\text{Body}} \rightarrow \underbrace{H}_{\text{Head}}$$

- semantically equivalent with $\neg A_1 \vee \neg A_2 \vee \ldots \vee \neg A_n \vee H$
- 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

FOL as Rule Language

- Rules as FOL implications (Horn Clause)



- *often written also from right to left (← or :-)*

Example Revisited

- The following example can be expressed via a **FOL-Rule**:

  "A squanderer is a person whose expenses are higher than their income."

$$\begin{aligned} &\text{Person}(x) \\ &\quad \wedge\ \text{hasIncome}(x,y) \\ &\quad \wedge\ \text{hasExpenses}(x,z) \\ &\quad \wedge\ (z > y) \\ &\quad \rightarrow\ \text{Squanderer}(x) \end{aligned}$$

DL and Rules

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

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

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

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 (Gallaire, Minkers, 1978)
- Knowledge Bases (Datalog Programs)
  are sets of horn clauses (without function symbols)
- DATALOG is decidable and computationally efficient, ExpTime

DATALOG Syntax

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

DATALOG Examples

- $Vegetarian(x) \wedge FishProduct(y) \rightarrow dislikes(x,y)$
- $orderedDish(x,y) \wedge dislikes(x,y) \rightarrow Unhappy(x)$
- $orderedDish(x,y) \rightarrow Dish(y)$
- $dislikes(x,z) \wedge Dish(y) \wedge contains(y,z) \rightarrow dislikes(x,y)$
- $\rightarrow Vegetarian(Matthias)$
- $Happy(x) \wedge Unhappy(x) \rightarrow$

- 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**.

Excursion 7: The Semantic Web Rule Language SWRL

Next Lecture...

[4]

Bibliographic References:

- Pascal Hitzler, Markus Krötzsch, Bijan Parsia, Peter F. Patel-Schneider, Sebastian Rudolph (eds., 2012), *OWL 2 Web Ontology Language Primer (Second Edition)*, W3C Recommendation 11 December 2012.
- Stefano Ceri, Georg Gottlob, Letizia Tanca (March 1989). "*What you always wanted to know about Datalog (and never dared to ask)*". IEEE Transactions on Knowledge and Data Engineering. 1 (1): 146–166.
- Aidan Hogan (2020), *The Web of Data*, Springer.
  Chap. 5.4.7 Features not supported in OWL, pp. 244–250.

Picture References:

[1] "A large owl in a space suit floating in deep space next to its spaceship over the surface of Mars.", created via ArtBot, Deliberate, 2023, [CC-BY-4.0], https://tinybots.net/artbot

[2] Benjamin Nowack, *The Semantic Web - Not a Piece of cake…*, at bnode.org, 2009-07-08 , [CC BY 3.0], https://web.archive.org/web/20220628120341/http://bnode.org/blog/2009/07/08/the-semantic-web-not-a-piece-of-cake

[3] "The Greek philosopher Heraclitus", created via ArtBot, Ceipher Female Model, 2023, [CC-BY-4.0], https://tinybots.net/artbot

[4] "A large owl in a space suit floating in deep space next to its spaceship over the surface of Mars.", created via ArtBot, Deliberate, 2023, [CC-BY-4.0], https://tinybots.net/artbot