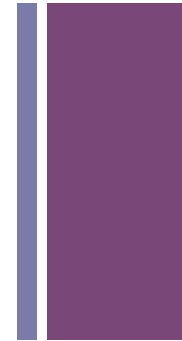


Representação de Conhecimento: Ontologias

Prof. Elder Rizzon Santos
ersantos@inf.ufsc.br

+ Ontologia

- Uma disciplina filosófica
 - Lida com a **natureza** e **organização** da realidade
- Ciência do “Ser” (Aristóteles, Metafísica)
- Tenta responder questões (difíceis) como:
 - *O que caracteriza ser?*
 - *O que é ser?*
- Também lida com organização de conhecimento
 - *Como as coisas devem ser classificadas?*



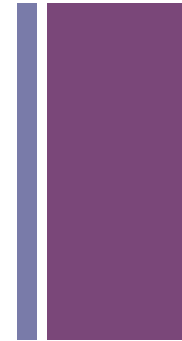
+ Ontologias – aspectos gerais

- Uma ontologia pode ser vista como um artefato de engenharia contendo (Horrocks):
 - Um vocabulário utilizado para descrever algum domínio
 - Uma especificação explícita do *significado* do vocabulário
 - Restrições capturando *conhecimento adicional*
- *Idealmente* uma ontologia deve:
 - Capturar um *entendimento compartilhado* de um domínio de interesse
 - Fornecer um modelo do domínio, que seja:
 - **formal e manipulável por máquinas**

+ Exemplo

- Ontologia de Domínio
- Vocabulário e significado (“definições”)
 - Elefante é um conceito cujos membros são do tipo animal
 - **Herbívoro** é um conceito cujos membros são *exatamente* aqueles animais que só comem plantas *ou partes de plantas*
 - **Elefante Adulto** é o conceito cujos membros são exatamente aqueles *elefantes* cuja *idade* é superior à 20 anos.
- Conhecimento do domínio / restrições / “axiomas gerais”
 - **Elefantes adultos** *pesam* no mínimo 2.000kg
 - *Todos elefantes* são **Elefantes Africanos** *ou* **Elefantes Indianos**
 - Nenhum indivíduo pode ser um **Herbívoro** e um **Carnívoro**

+ Ontologias – Tipos/níveis



■ **Base/Genérica/Alto Nível**

- Mais próximas da noção filosófica
- Construtos universais
- *Upper-Ontologies*. Ex.: SUMO, DOLCE, Sowa

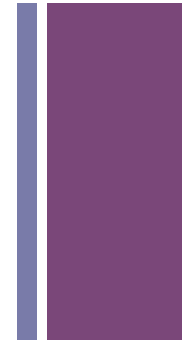
■ **Domínio**

- Conceitos e relações de uma área de interesse particular. Ex.: FMA, UMLS, Gene Ontology

■ **Tarefa/Método**

- Conceitualizações necessárias para realizar uma tarefa
- Ontologias de estados e transições. Tate's plan Onto

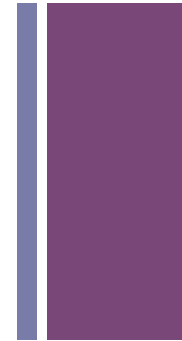
+ Ontologias – em computação



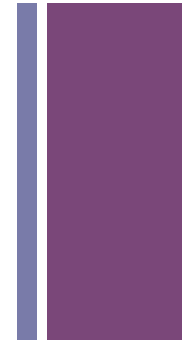
- Uma especificação formal e explícita de uma conceitualização compartilhada. Gruber
 - Formal
 - Explícita
 - Conceitualização
 - Compartilhada
- IA, KR, Engenharia de Conhecimento, Engenharia de Ontologias

+ Aplicações

- e-Science, e-Government, Bioinformática
 - The Gene Ontology
 - The Protein Ontology (MGED)
- SGBD
 - Integração de bases heterogêneas
 - Resposta à perguntas
- Processamento de Linguagem Natural
- Web Semântica
- Sensores e contexto
- Sistemas de Recomendação
 - Modelos/perfis de usuários

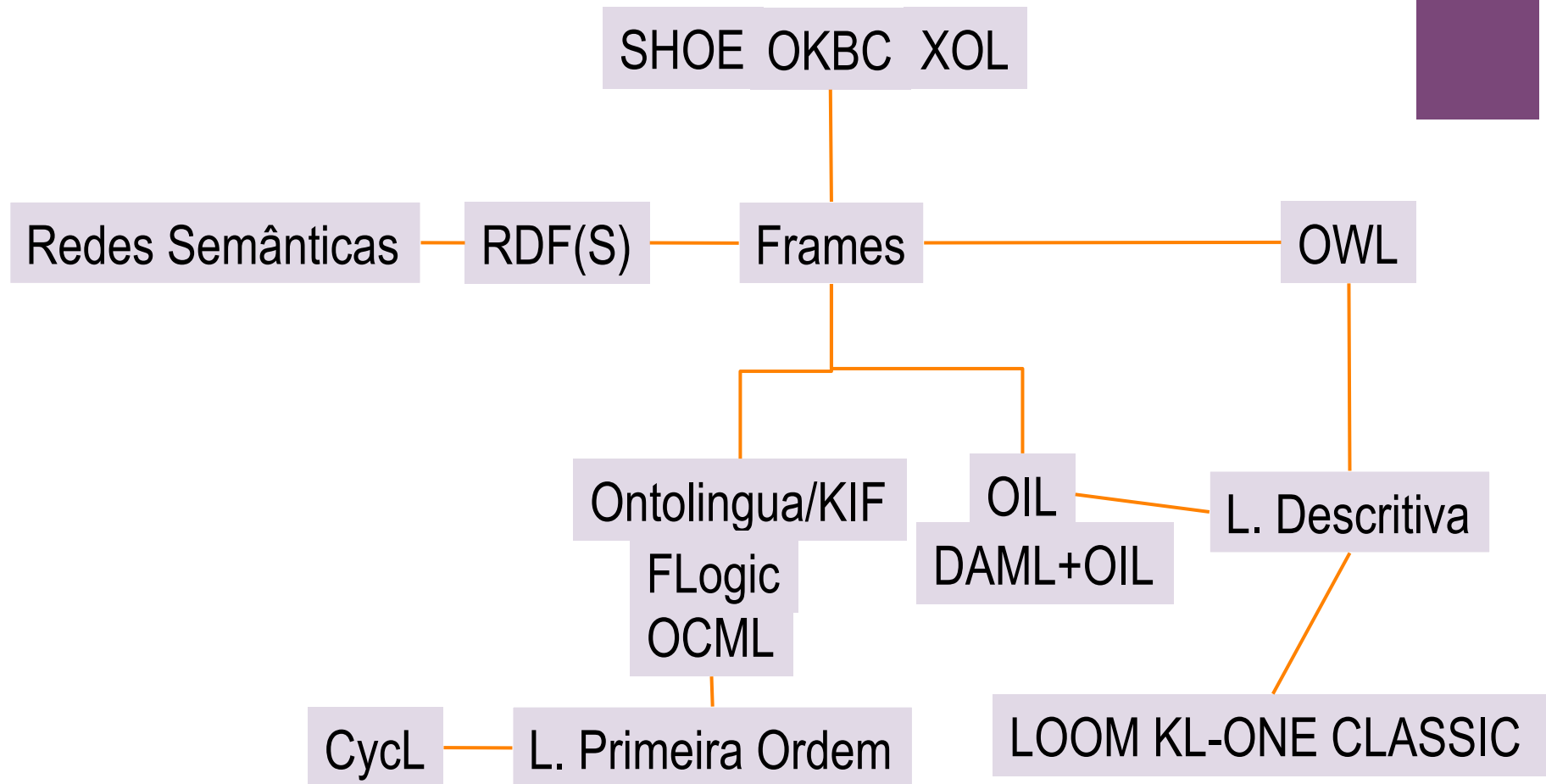


+ Linguagens ontológicas



- Possibilitam uma especificação **explícita** do domínio
 - Lógica
 - Grafos conceituais
 - Lógicas não clássicas (modalidades, graus de crença, etc.)
 - Probabilísticas, bayesianas, fuzzy
- Diferentes níveis de expressividade
- A formalização (sintaxe e semântica) possibilita o raciocínio automatizado (inferência)

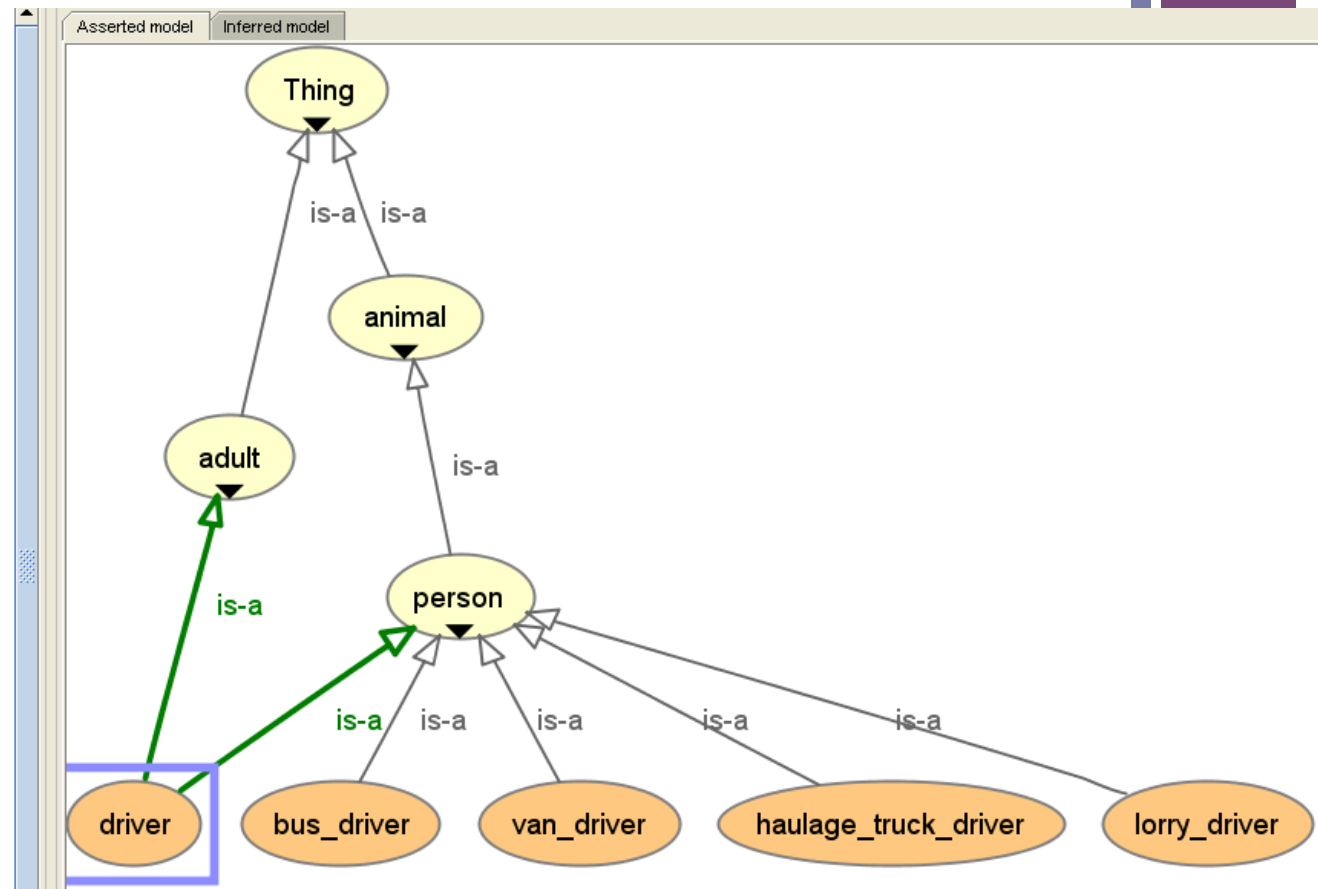
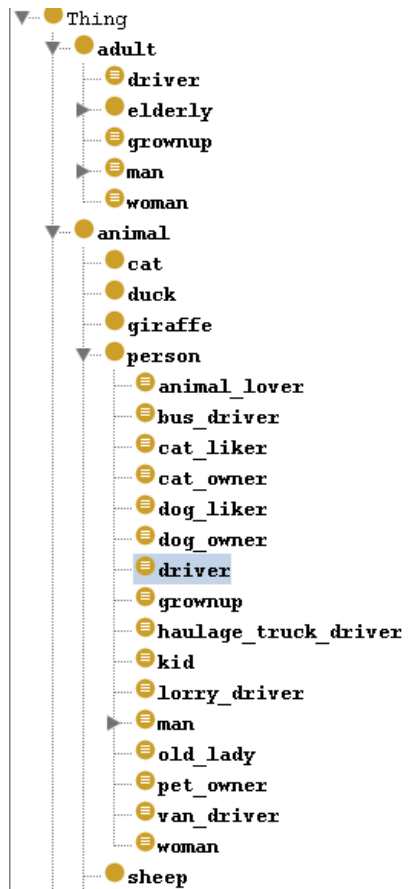
+ Formalismos & Linguagens



+ Elementos da Linguagem

- <http://www.w3.org/TR/owl2-syntax/>

+ Exemplo – Raciocínio TBox



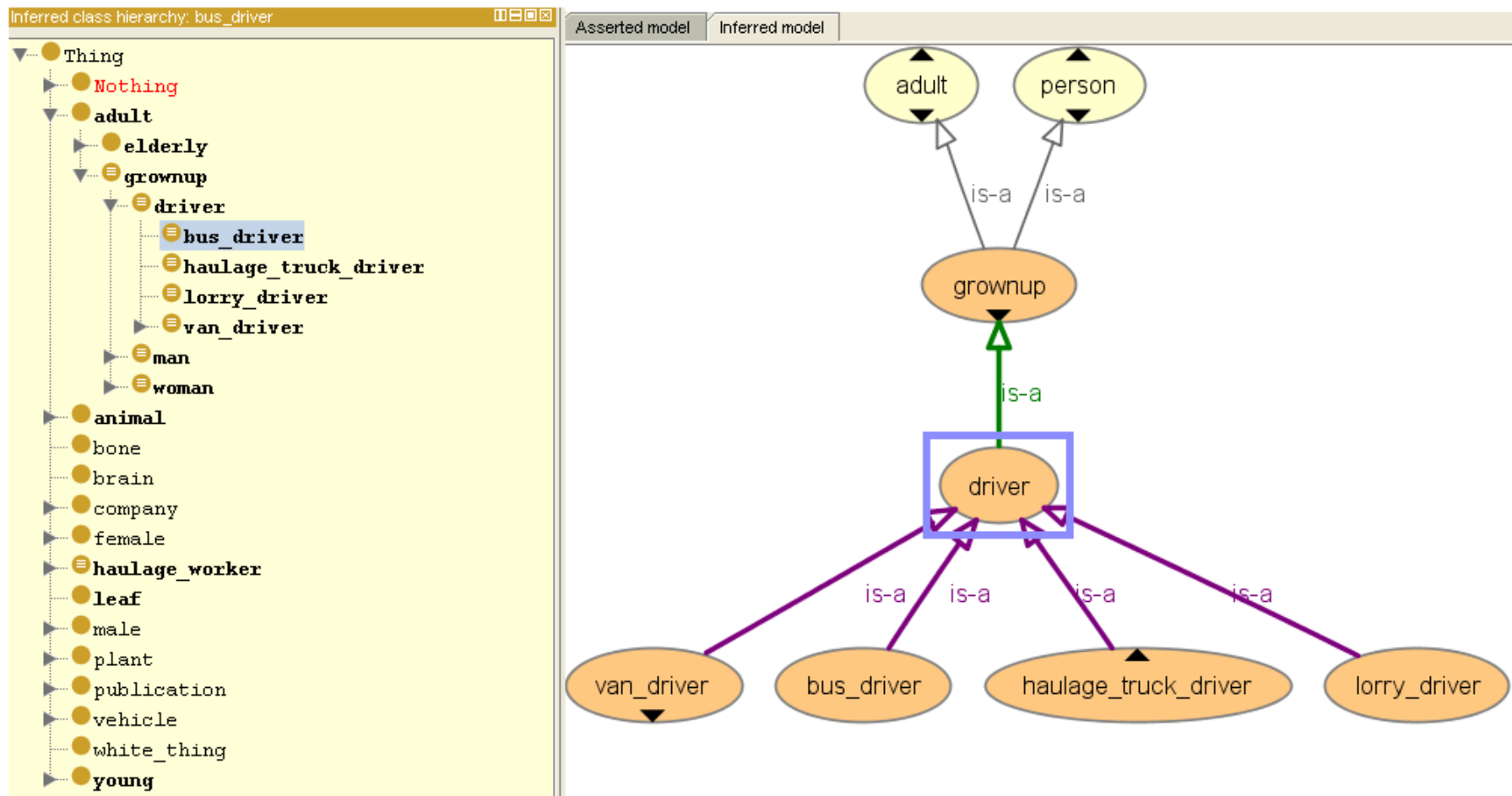
+ Exemplo – Raciocínio tBox

- `Class(a:bus_driver complete intersectionOf(a:person restriction(a:drives someValuesFrom (a:bus))))`
- `Class(a:driver complete intersectionOf(a:person restriction(a:drives someValuesFrom (a:vehicle))))`
- `Class(a:bus partial a:vehicle)`
- Um `motorista_de_onibus` é uma pessoa que dirige um ônibus.
- Um ônibus é um veículo.
- `Motorista_de_onibus` dirige um veículo, logo é motorista.

+ Exemplo – raciocínio Tbox

■ Ontologia Inferida (classificação)

- Bus_driver é subclasse de Driver
- Todo Driver é Grownup



+ Exemplo – Raciocínio ABox

Description: Tom	Property assertions: Tom
<p>Types +</p> <ul style="list-style-type: none">Thingcatpet <p>Same individuals +</p> <p>Different individuals +</p>	<p>Object property assertions +</p> <ul style="list-style-type: none">is_pet_of Minnie <p>Data property assertions +</p> <p>Negative object property assertions +</p> <p>Negative data property assertions +</p>
Description: Minnie	Property assertions: Minnie
<p>Types +</p> <ul style="list-style-type: none">elderlyfemaleold_lady <p>Same individuals +</p> <p>Different individuals +</p>	<p>Object property assertions +</p> <ul style="list-style-type: none">has_pet Tomlikes Tom <p>Data property assertions +</p> <p>Negative object property assertions +</p> <p>Negative data property assertions +</p>

+ Exemplo – Raciocínio abox

Individual(a:Minnie type(a:female) type(a:elderly) value(a:has_pet a:Tom))

Individual(a:Tom type(owl:Thing))

ObjectProperty(a:has_pet domain(a:person) range(a:animal))

Class(a:old_lady complete intersectionOf(a:person a:female a:elderly))

Class(a:old_lady partial intersectionOf(restriction(a:has_pet allValuesFrom (a:cat))
restriction(a:has_pet someValuesFrom (a:animal))))

- **Ou seja, Tom, é declarado somente como Thing e é pet de Minie**
- A restrição de domínio permite inferir um tipo mais específico (Animal) e a quantificação universal possibilita inferir o tipo dessa instância