Autonomes Fahren

SS 2019

Multi-Sensor-Data-Fusion 2

Technische Universität München

Ontologies for Autonomous Driving

- An Ontology is a formal naming and definition of types, properties and interrelationships of entities that fundamentally exist for a particular domain of discourse.
- Web Ontology Language (OWL) is a family of knowledge representation knowledges fo authoring ontologies.
- An Ontology describing families might include axioms stating that a “hasMother” property is only present between two individuals when “hasParent” is also present, and individuals of class “HasType0Blood” are never related via “hasParent” to members of the “HasTypeABBlood”.

Ontologies for Autonomous Driving

- Individual: Instances or objects
- Classes: Sets, collections, concepts, types of objects
- Attributes: Aspects, properties, features, parameters of objects
- Relations: Ways in which classes and individuals can be related
- Function terms: Complex structures formed from certain relations
- Restrictions: Formal stated descriptions to restrict input
- Rules: Statements for logical inferences
- Axioms: Assertions in a logical form

Ontologies for Autonomous Driving

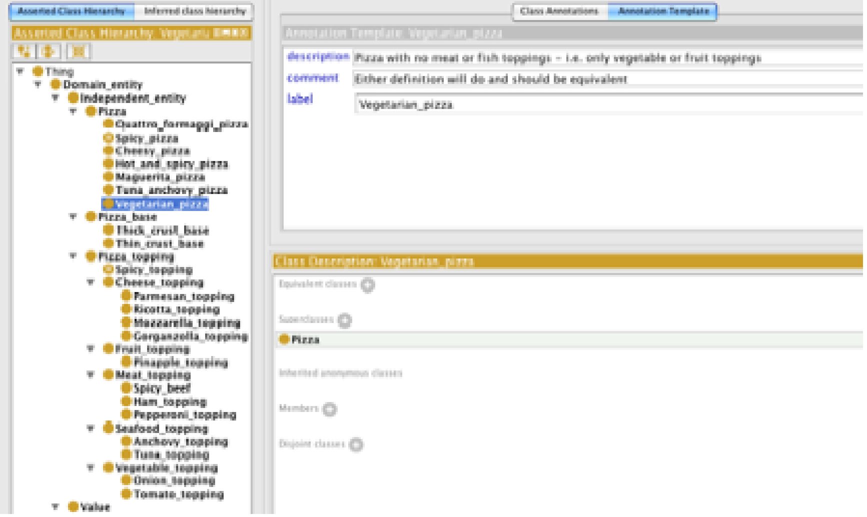
- Individual: Thorsten
- Classes: Persons
- Attributes: <has as name> Thorsten
- Relations: <is the father of> Maria

Ontologies for Autonomous Driving

- Famous example – OWL Pizza
- Class hierarchy:
  - Thing
    - Pizza
      - Vegetarian
      - Cheesey
      - Mageherita
      - Hot_And_Spicy
      - Seafood
    - Pizza_topping
      - Tomato_topping
      - Mozarella_topping
      - Spicy_beef_topping
      - Pepperoni_topping
    - Pizza_base
      - Thick_crust
      - Thin_crust

Ontologies for Autonomous Driving

- Famous example – OWL Pizza (https://protegewiki.stanford.edu/wiki/File:Alr-matrix-objproperties.png)
- Class hierarchy:
  - Thing
    - Pizza
      - Vegetarian
      - Cheesey
      - Mageherita
      - Hot_And_Spicy
      - Seafood
    - Pizza_topping
      - Tomato_topping
      - Mozarella_topping
      - Spicy_beef_topping
      - Pepperoni_topping
    - Pizza_base
      - Thick_crust
      - Thin_crust



- Thing
  - Domain_entity
    - Independent_entity
      - Pizza
        - Quattro_formaggi_pizza
        - Spicy_pizza
        - Cheesy_pizza
        - Hot_and_spicy_pizza
        - Maguerita_pizza
        - Tuna_anchovy_pizza
        - Vegetarian_pizza
      - Pizza_base
        - Thick_crust_base
        - Thin_crust_base
      - Pizza_topping
        - Spicy_topping
        - Cheese_topping
          - Parmesan_topping
          - Ricotta_topping
          - Mozzarella_topping
          - Gorganzolla_topping
        - Fruit_topping
          - Pinapple_topping
        - Meat_topping
          - Spicy_beef
          - Ham_topping
          - Pepperoni_topping
        - Seafood_topping
          - Anchovy_topping
          - Tuna_topping
        - Vegetable_topping
          - Onion_topping
          - Tomato_topping
    - Value

description: Pizza with no meat or fish toppings – i.e. only vegetable or fruit toppings

comment: Either definition will do and should be equivalent

label: Vegetarian_pizza

Class Description: Vegetarian_pizza

Equivalent classes

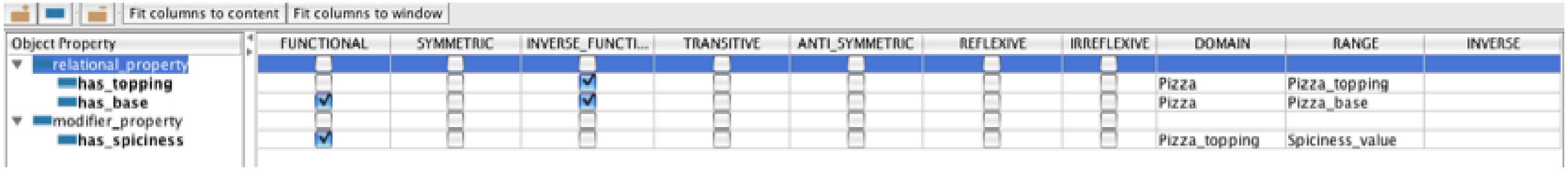
Superclasses

Pizza

Inherited anonymous classes

Members

Disjoint classes

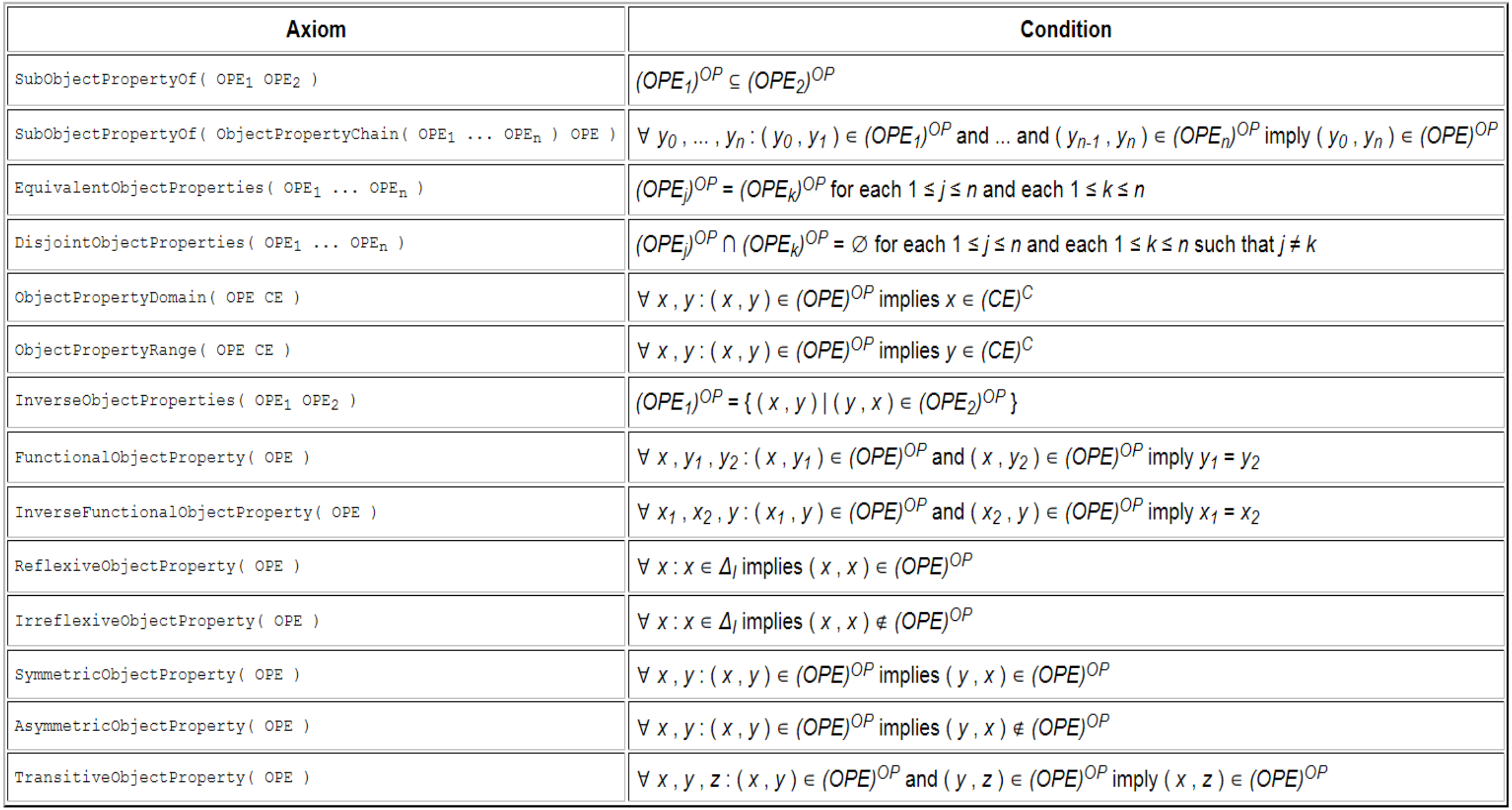


Object Property	FUNCTIONAL	SYMMETRIC	INVERSE_FUNCTI...	TRANSITIVE	ANTI_SYMMETRIC	REFLEXIVE	IRREFLEXIVE	DOMAIN	RANGE	INVERSE
relational_property	□	□	□	□	□	□	□			
has_topping	□	□	☑	□	□	□	□	Pizza	Pizza_topping	
has_base	☑	□	☑	□	□	□	□	Pizza	Pizza_base	
modifier_property	□	□	□	□	□	□	□			
has_spiciness	☑	□	□	□	□	□	□	Pizza_topping	Spiciness_value	

Ontologies for Autonomous Driving

- Symmetric: hasSpouse relates A with B if it relates B with A.
- Inverse: hasParent as inverse property of hasChild
- Transitive: Interlinks two individuals A and C when A is interlinked with B and B with C.
- Anti_Symmetric: If A is connected with B, B is not connected with A.
- Reflexive: Relates everything to itself
- Irreflexive: No individual can be related to itself through this property

Ontologies for Autonomous Driving

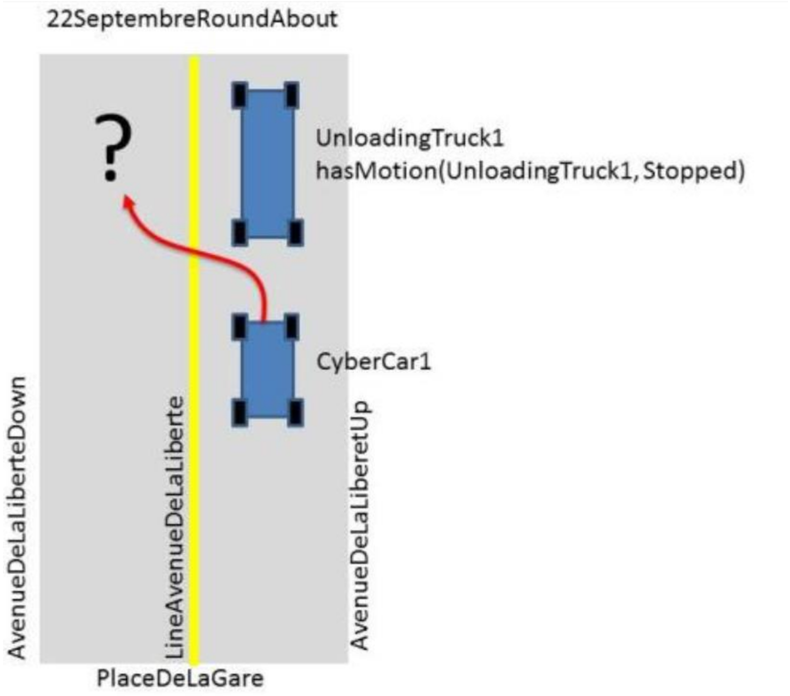


Axiom	Condition
SubObjectPropertyOf(OPE$_1$ OPE$_2$)	$(OPE_1)^{OP} \subseteq (OPE_2)^{OP}$
SubObjectPropertyOf(ObjectPropertyChain(OPE$_1$ ... OPE$_n$) OPE)	$\forall\ y_0 , \ldots , y_n : (y_0 , y_1) \in (OPE_1)^{OP}$ and ... and $(y_{n-1} , y_n) \in (OPE_n)^{OP}$ imply $(y_0 , y_n) \in (OPE)^{OP}$
EquivalentObjectProperties(OPE$_1$ ... OPE$_n$)	$(OPE_j)^{OP} = (OPE_k)^{OP}$ for each $1 \leq j \leq n$ and each $1 \leq k \leq n$
DisjointObjectProperties(OPE$_1$ ... OPE$_n$)	$(OPE_j)^{OP} \cap (OPE_k)^{OP} = \emptyset$ for each $1 \leq j \leq n$ and each $1 \leq k \leq n$ such that $j \neq k$
ObjectPropertyDomain(OPE CE)	$\forall\ x , y : (x , y) \in (OPE)^{OP}$ implies $x \in (CE)^{C}$
ObjectPropertyRange(OPE CE)	$\forall\ x , y : (x , y) \in (OPE)^{OP}$ implies $y \in (CE)^{C}$
InverseObjectProperties(OPE$_1$ OPE$_2$)	$(OPE_1)^{OP} = \{ (x , y) \mid (y , x) \in (OPE_2)^{OP} \}$
FunctionalObjectProperty(OPE)	$\forall\ x , y_1 , y_2 : (x , y_1) \in (OPE)^{OP}$ and $(x , y_2) \in (OPE)^{OP}$ imply $y_1 = y_2$
InverseFunctionalObjectProperty(OPE)	$\forall\ x_1 , x_2 , y : (x_1 , y) \in (OPE)^{OP}$ and $(x_2 , y) \in (OPE)^{OP}$ imply $x_1 = x_2$
ReflexiveObjectProperty(OPE)	$\forall\ x : x \in \Delta_I$ implies $(x , x) \in (OPE)^{OP}$
IrreflexiveObjectProperty(OPE)	$\forall\ x : x \in \Delta_I$ implies $(x , x) \notin (OPE)^{OP}$
SymmetricObjectProperty(OPE)	$\forall\ x , y : (x , y) \in (OPE)^{OP}$ implies $(y , x) \in (OPE)^{OP}$
AsymmetricObjectProperty(OPE)	$\forall\ x , y : (x , y) \in (OPE)^{OP}$ implies $(y , x) \notin (OPE)^{OP}$
TransitiveObjectProperty(OPE)	$\forall\ x , y , z : (x , y) \in (OPE)^{OP}$ and $(y , z) \in (OPE)^{OP}$ imply $(x , z) \in (OPE)^{OP}$

https://www.w3.org/TR/owl2-direct-semantics/#Object_Property_Expression_Axioms

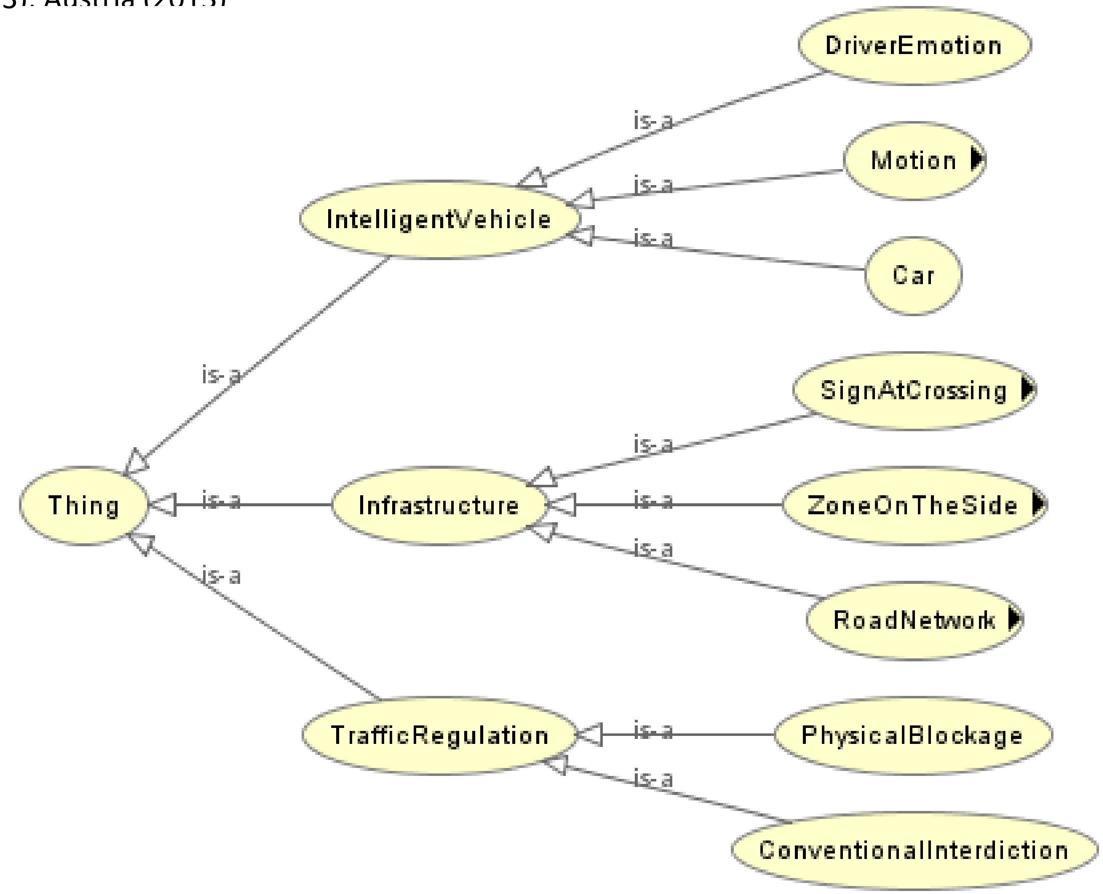
Ontologies for Autonomous Driving

Example: “An Ontology based approach to relax traffic regulation for autonomous vehicle assistance” Philippe Morignot, Fawzi Nashashib; 12th IASTED International Conference on Artificial Intelligence and Applications (AIA'13), Austria (2013)



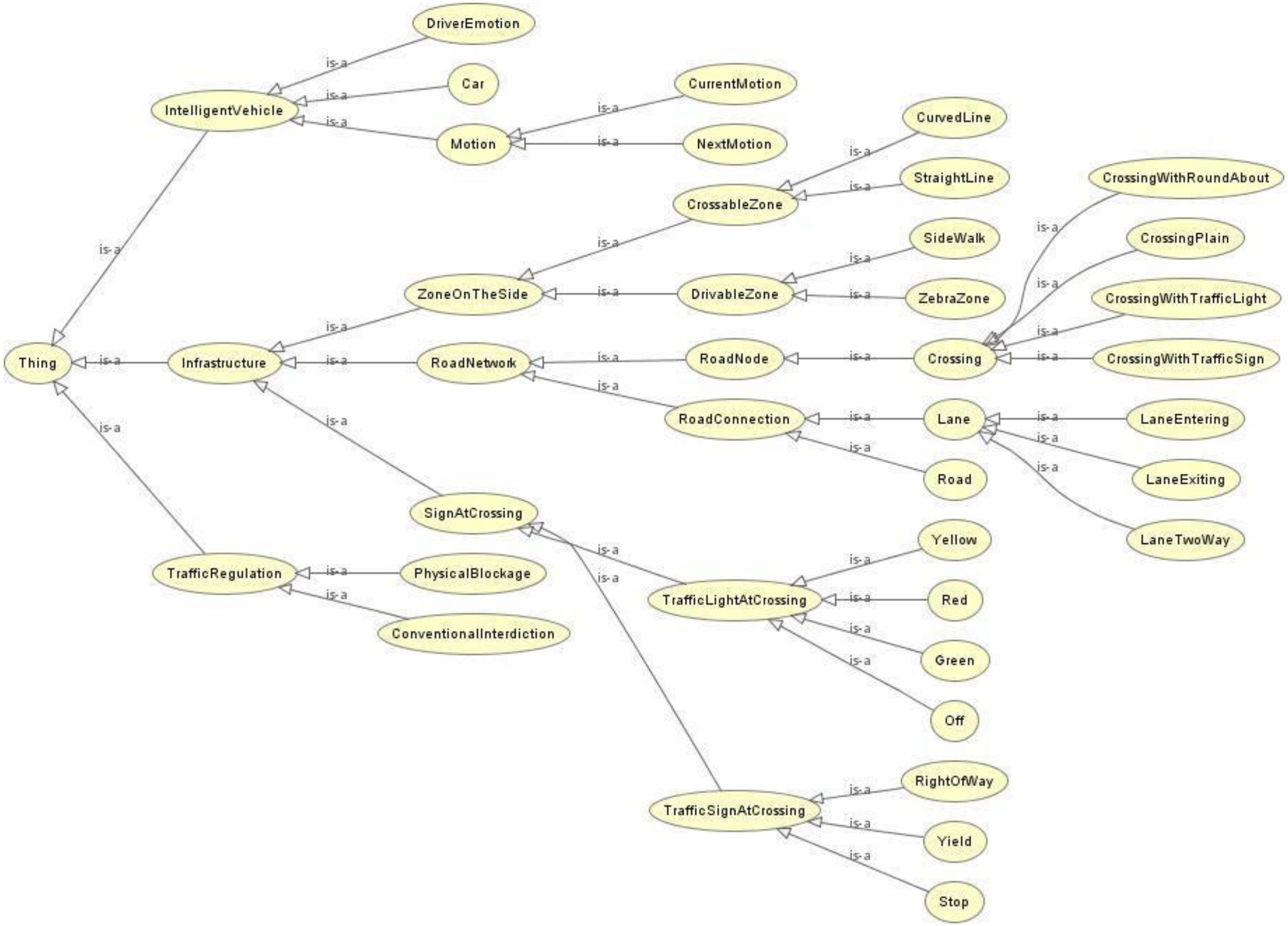
Ontologies for Autonomous Driving

Example: “An Ontology based approach to relax traffic regulation for autonomous vehicle assistance” Philippe Morignot, Fawzi Nashashib; 12th IASTED International Conference on Artificial Intelligence and Applications (AIA'13), Austria (2013)



Ontologies for Autonomous Driving

Example: “An Ontology based approach to relax traffic regulation for autonomous vehicle assistance” Philippe Morignot, Fawzi Nashashib; 12th IASTED International Conference on Artificial Intelligence and Applications (AIA'13), Austria (2013)



Ontologies for Autonomous Driving

Example: “An Ontology based approach to relax traffic regulation for autonomous vehicle assistance” Philippe Morignot, Fawzi Nashashib; 12th IASTED International Conference on Artificial Intelligence and Applications (AIA'13), Austria (2013)

SWRL Rules:

```
CrossableZone(?s), Car(?a), Car(?b),
Lane(?l1), Lane(?l2),
hasEmotion(?a, Nervous),
isAfter(?a, ?b),
hasBesides(?l1, ?s), hasBesides(?l2, ?s),
hasMotion(?a, Stopped),
isOn(?a, ?l1), isOn(?b, ?l1), DifferentFrom (?l1, ?l2),
isIllegal(?l1, ?l2),
isClear(?l2)
->
isNextOn(?a, ?l2)
```

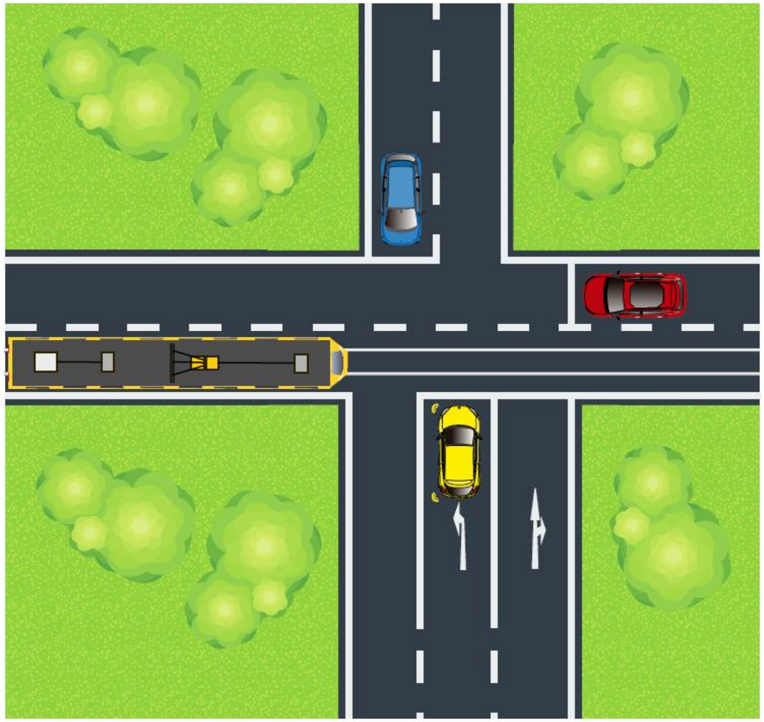
```
DrivableZone(?s), Car(?a), Car(?b), Lane(?l),
hasEmotion(?a, Nervous),
isAfter(?a, ?b),
hasBesides(?l, ?s),
hasMotion(?a, Stopped),
isOn(?a, ?l), isOn(?b, ?l),
isIllegal( ?l, ?s),
->
isNextOn(?a, ?s)
```

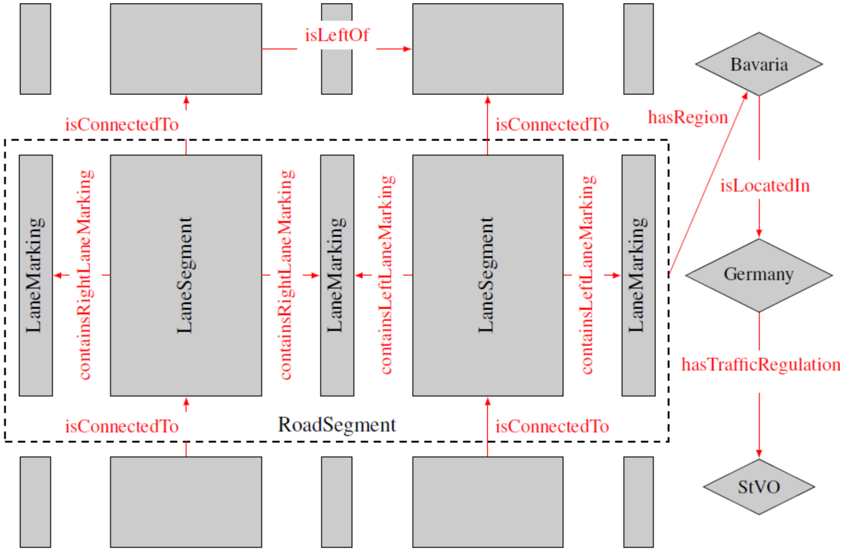
```
Car(?a), Car(?b),
hasNextMotion(?a, Forward),
isBefore(?a, ?b)
->
hasNextMotion(?b, Forward)
```

```
Car(?a), Car(?b),
hasMotion(?a, Stopped),
isBefore(?a, ?b)
->
hasMotion(?b, Stopped)
```

Ontologies for Autonomous Driving

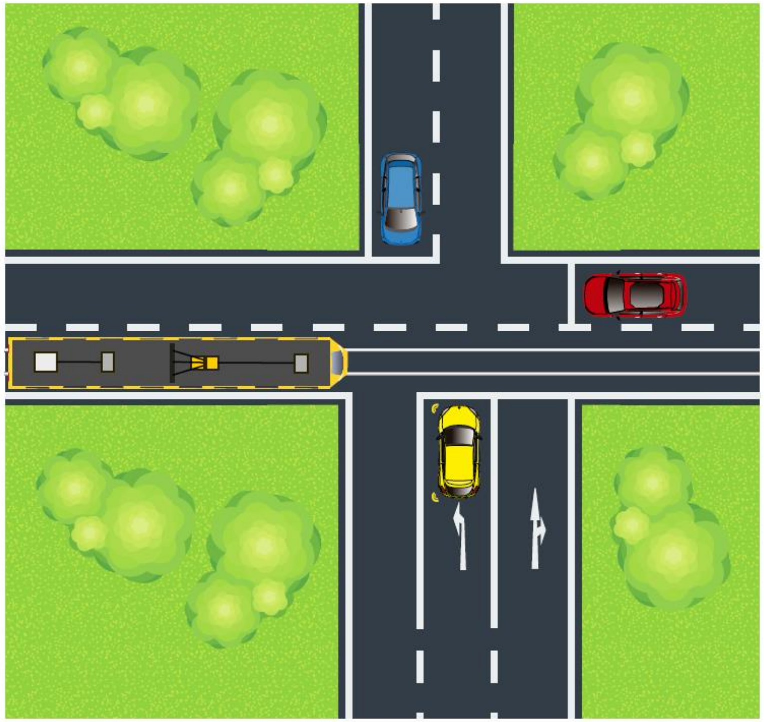
Example: “Ontology-Based Traffic Scene Modeling, Traffic Regulations Dependent Situational Awareness and Decision-Making for Automated Vehicles”, M. Büchel, G. Hinz et. Al. in *IEEE Intelligent Vehicles Symposium, 2017*.

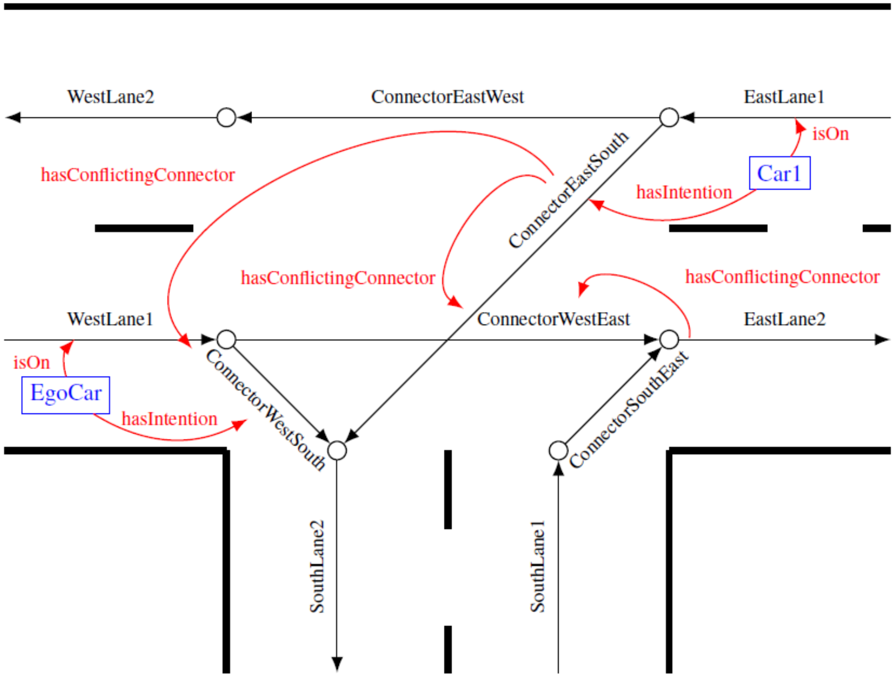




Ontologies for Autonomous Driving

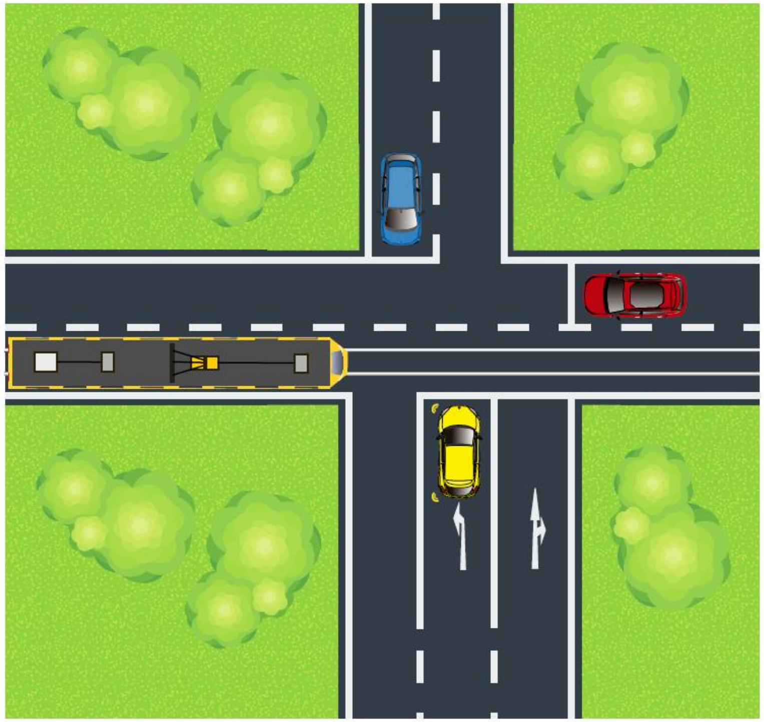
Example: “Ontology-Based Traffic Scene Modeling, Traffic Regulations Dependent Situational Awareness and Decision-Making for Automated Vehicles”, M. Büchel, G. Hinz et. Al. in *IEEE Intelligent Vehicles Symposium, 2017*.

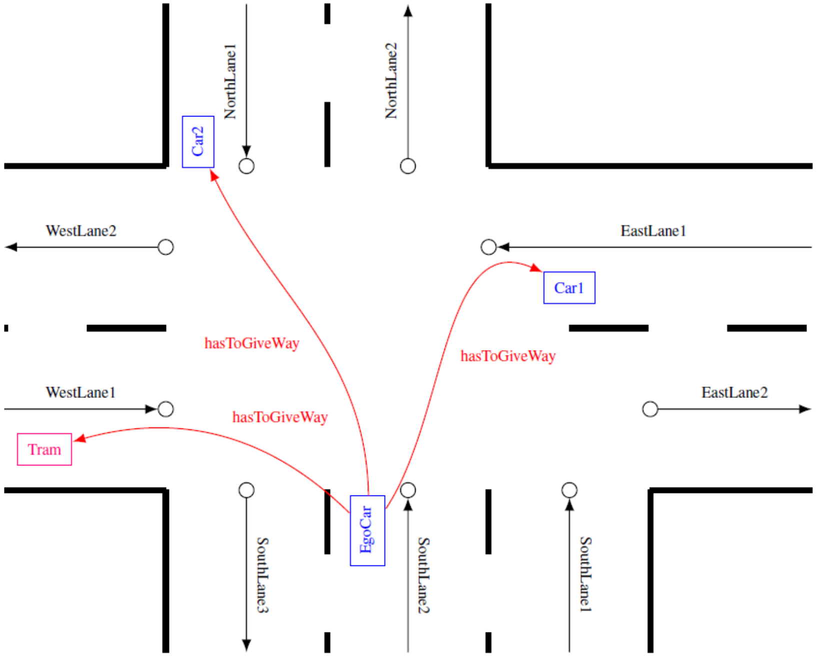




Ontologies for Autonomous Driving

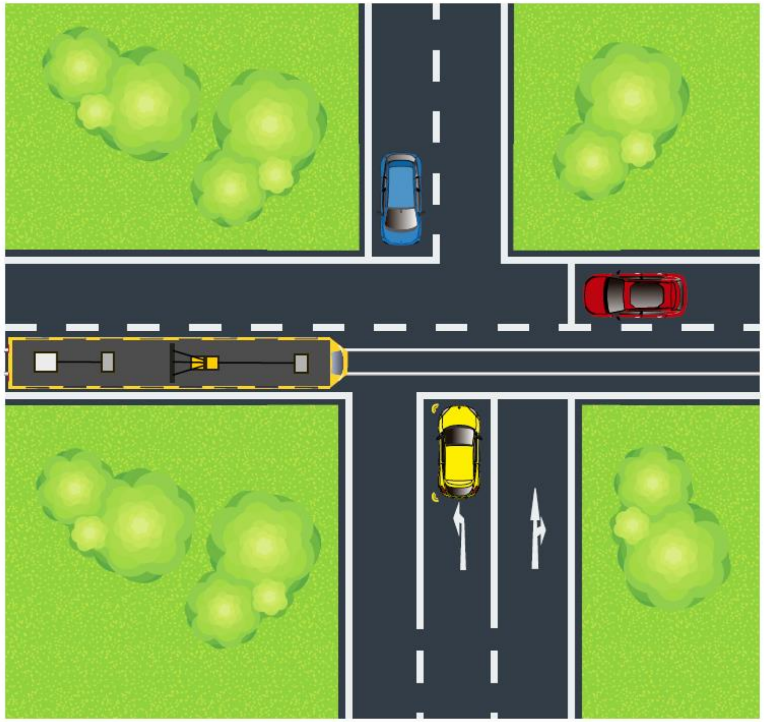
Example: “Ontology-Based Traffic Scene Modeling, Traffic Regulations Dependent Situational Awareness and Decision-Making for Automated Vehicles”, M. Büchel, G. Hinz et. Al. in *IEEE Intelligent Vehicles Symposium, 2017*.





Ontologies for Autonomous Driving

Example: “Ontology-Based Traffic Scene Modeling, Traffic Regulations Dependent Situational Awareness and Decision-Making for Automated Vehicles”, M. Büchel, G. Hinz et. Al. in *IEEE Intelligent Vehicles Symposium, 2017*.



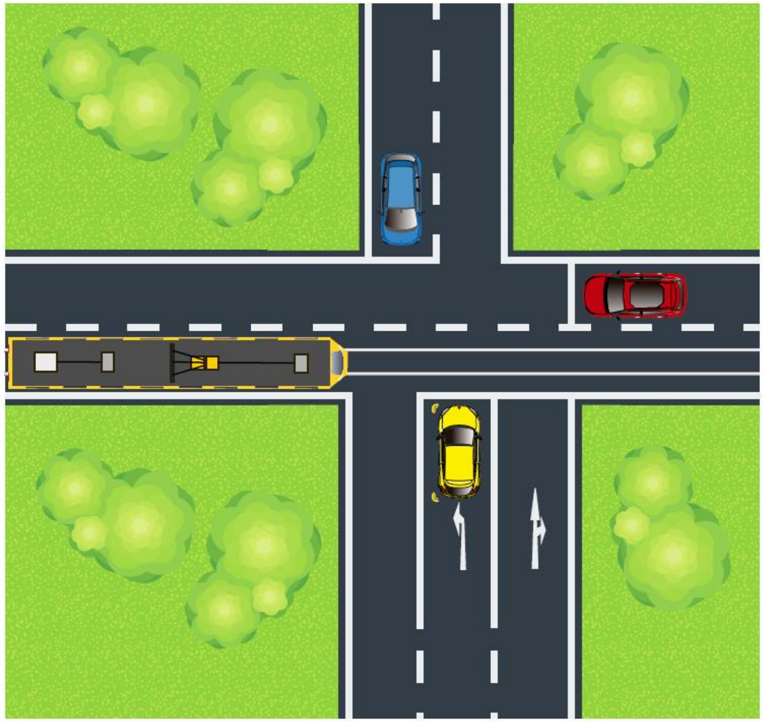
$$
\begin{aligned}
&containsTrafficLight\ some\ TrafficLight \text{ and}\\
&containsTrafficRegulatingPerson\\
&exactly\ 0\ TrafficRegulatingPerson
\end{aligned}
$$

$$
\begin{aligned}
&TrafficLightRegulatedIntersection(?scen1)\\
&\wedge Vehicle(?vehicle1)\ \wedge\ isOn(?vehicle1, ?lane)\\
&\wedge isPartOf(?vehicle1, ?scen1)\\
&\wedge containsTrafficLight(?lane, ?light)\\
&\wedge hasColor(?light, Red)\\
&\Rightarrow\ hasToWaitForTrafficLight(?vehicle1, ?light)
\end{aligned}
$$

$$
\begin{aligned}
&RoadSegment(?road)\\
&\wedge preferredLane(?trafficregulations, "right")\\
&\wedge hasRightMostLane(?road, ?lane)\\
&\wedge hasTrafficRegulations(?road, ?trafficregulations)\\
&\Rightarrow\ hasPreferredLane(?road, ?lane)
\end{aligned}
$$

Ontologies for Autonomous Driving

Example: “Ontology-Based Traffic Scene Modeling, Traffic Regulations Dependent Situational Awareness and Decision-Making for Automated Vehicles”, M. Büchel, G. Hinz et. Al. in *IEEE Intelligent Vehicles Symposium, 2017*.



$$containsTrafficLight\ exactly\ 0\ TrafficLight \text{ and } containsTrafficRegulatingPerson\ exactly\ 0\ TrafficRegulatingPerson$$

$$\begin{aligned}
&UncontrolledIntersection(?scen1)\\
&\wedge Vehicle(?vehicle1)\ \wedge\ Vehicle(?vehicle2)\\
&\wedge isPartOf(?vehicle1, ?scen1)\\
&\wedge hasConflictingConnector(?c1, ?c2)\\
&\wedge hasIntention(?vehicle1, ?c1) \wedge hasIntention(?vehicle2, ?c2)\\
&\Rightarrow\ hasToGiveWay(?vehicle1, ?vehicle2)
\end{aligned}$$

$$\begin{aligned}
&TrafficRegulatedPersonIntersection(?scen1)\\
&\wedge Vehicle(?vehicle1)\ \wedge\ contains(?scen1, ?vehicle1)\\
&\wedge isOn(?vehicle1, ?lane1)\ \wedge\ isBlockedBy(?lane1, ?police)\\
&\Rightarrow\ hasToWaitForTRP(?vehicle1, ?police)
\end{aligned}$$