PIRO
SYSTEMS ENGINEERING

Mehr aus Requirements machen

automatische Verarbeitung, Ontologien & mehr

27. April 2022

Agenda

- Who/What is PiRo?
- Where everything started
- General Idea
- Automatic Requirements Processing Possibilities

PiRo Systems Engineering GmbH

- Founded 2016
- Stuttgart/Boeblingen

»get things done«

»aSPICE, ISO26262, ISO9001, …«

- Systems & Requirement Engineering Consultancy
- Tools and Solutions

»we understand what we specify«

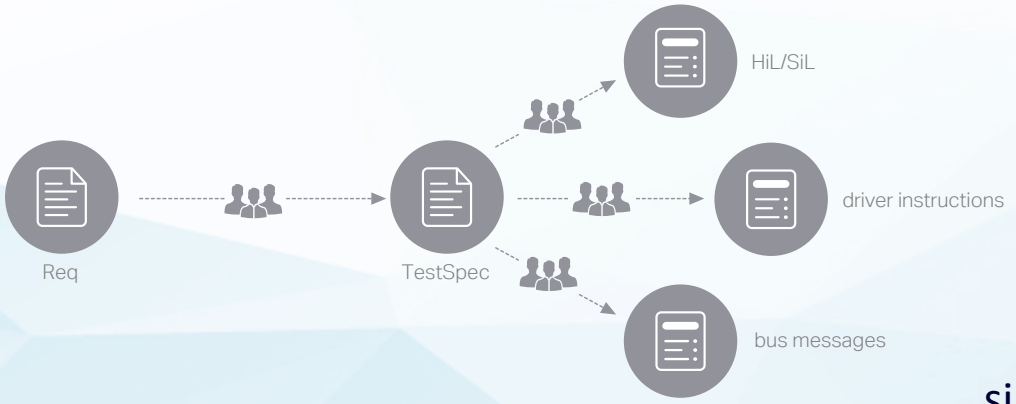
- Projects
  - OEM/Tier1/Tier2/StartUps/KMU
  - Camera/Ultrasonic/Lidar/DomainController/Data Logger/…

»get complexity managed«

Where everything started

Around 10 years ago, at a German OEM ….

Mit Diagnosesoftware die Funktion APC-Bremsen auscodieren (CAN-BR: Codierung CGW. FR-BR: Codierung in PARK, dafür Entwicklungs-cbf erforderlich). Drei Zündungswechsel durchführen (Übernahme Codierung).
Mit Diagnosesoftware die Botschaften APX_BRK_RQ (z.B. APX_BrkMd_Rq) und APX_BRK_RS (z.B. APX_BrkMd_Stat) beobachten.



Gute Testfälle

sind:

1. Gut “lesbar”
   - Schnell verständlich
   - Eindeutig / nicht interpretierbar → Maschinenlesbar?!
   - Einheitlich über alle Testspezifikationen
2. Schnell erstellt
3. Effektiv am HiL umsetzbar
4. Einfach änderbar
5. Auswertbar (Requirementabdeckung, …)
6. Debug-fähig, speziell bei HIL-Anwendungen
7. (Auch durch verschiedene externe Dienstleister eindeutig verwendbar)

Mit Diagnosesoftware die Funktion APC-Bremsen auscodieren (CAN-BR: Codierung CGW. FR-BR: Codierung in PARK, dafür Entwicklungs-cbf erforderlich). Drei Zündungswechsel durchführen (Übernahme Codierung).
Mit Diagnosesoftware die Botschaften APX_BRK_RQ (z.B. APX_BrkMd_Rq) und APX_BRK_RS (z.B. APX_BrkMd_Stat) beobachten.

- set scn:"APC-Bremsen
- action:IgnitionCycle[3]

→ Normung

… since 2015 known as ISO29119 Part5 “Keyword Driven Testing”

Development1.0

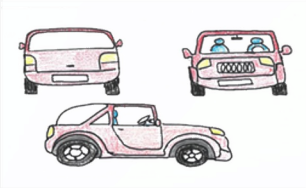
Islands instead of connected and interacting things

Industrie4.0

PIRO
SYSTEMS ENGINEERING

Given Customer Requirements

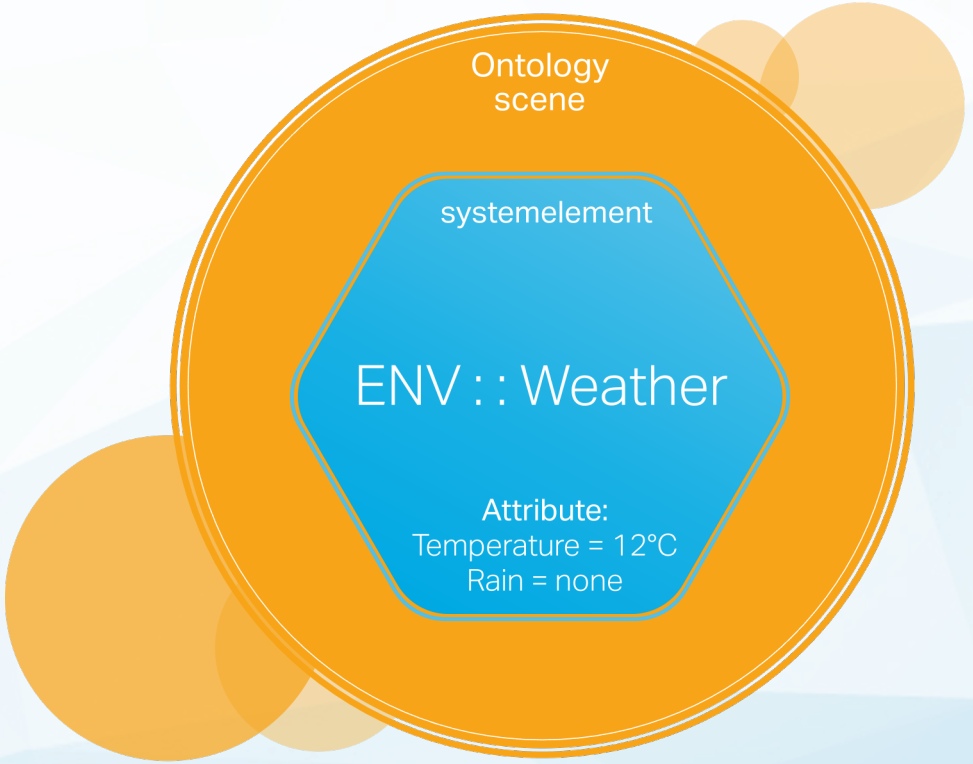
PiRo SYSTEMS ENGINEERING

Item ID	Summary	Description
38778	[ROT-38778] Aussehen	Der Wagen muss so aussehen 
38779	[ROT-38779] Größen	Der Wagen muss folgende Größen haben:
38780	[ROT-38780] Wolf	Der Wagen muss einem ausgewachsenem Wolf davonfahren können
38781	[ROT-38781] Straßen	Der Wagen muss außerhalb von Straßen und durch den Wald fahren können
38782	[ROT-38782] Jahreszeiten	Der Wagen muss zu allen Jahreszeiten funktionieren
38783	[ROT-38783] Kuchen	Der Wahen soll einen Kofferraum haben, in dem mindestens ein kuchen transportiert werden kann
38784	[ROT-38784] Sitzplätze	Der Wagen darf nicht mehr als zwei Sitzplätze haben

Maßangaben	**Werte**
Länge	3,9m
Breite	1,7m
Höhe	1,65m
Kofferraumvolumen	700L
Leere Gesamtgewicht	950Kg
Zulässige Beladung	500Kg

Requirements Modelling

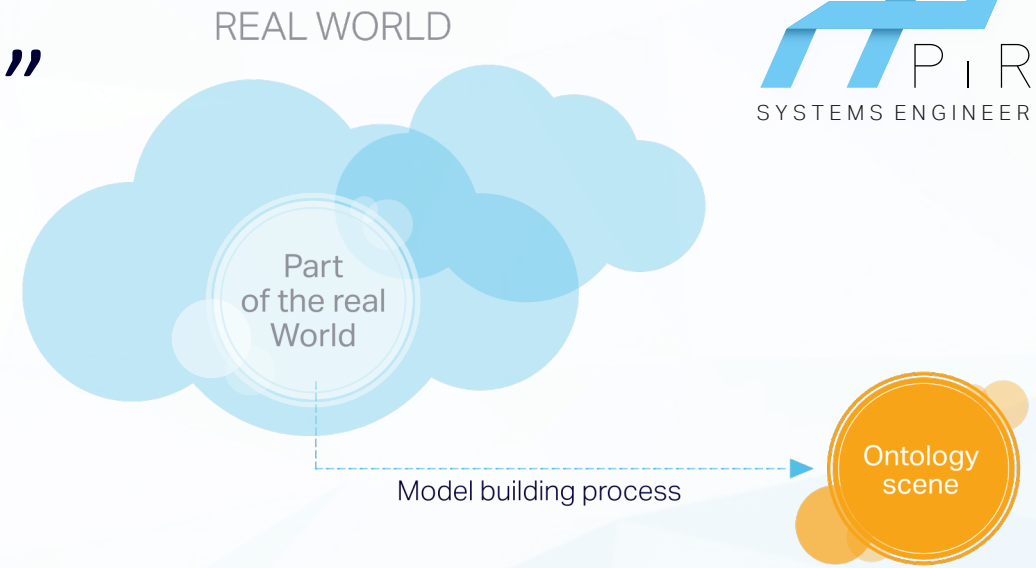
- A single object will be called a systemelement.
- A systemelement has a qualitative description given by a class and an entity.
- A systemelement has a quantitative description given by a set of attributes.



We may have a lot of systemelements in an ontology scene. Each systemelement has a unique class and entity but may have a lot of attributes.

Requirements “Modelling”

- VEH::Rot(…)
  - **length**=3.9m
  - **width**=1.7m
  - **high**=1.65m
  - **TrunkVolume**=700L
  - **TotalEmptyWeight**=950Kg
  - **NumberOfSeats**=2
  - **MaxSpeed**=50km/h
- ENV::Season(..)
  - **SeasonType**=winter/spring/summer/autumn
- ENV::Loading
  - **LoadingType**=cake
- ENV::Road(type)
  - **RoadType**=TarredRoad, Offroad, …
- ENV::DangerousObject()
  - **ObjectType**=Wolf



The **Veh::Rot** shall have the **length=3.9m**

OR

Veh::Rot(length=3.9)

OR

…

Automatic Analysis

PIRO SYSTEMS ENGINEERING

Class/Entity Existence Validation	Parameter Existence Validation	Parameter Data Type Validation	Parameter Range Validation
Check classes and entities for existence – generate error messages in case of missing classes and entities	Check Parameter for existence – generate error messages in case of missing parameter	Check parameter data types against type definitions – generate error messages in case of wrong numeric formats	Check parameter ranges for bound violations – generate error messages in case range errors

Automatic Analysis

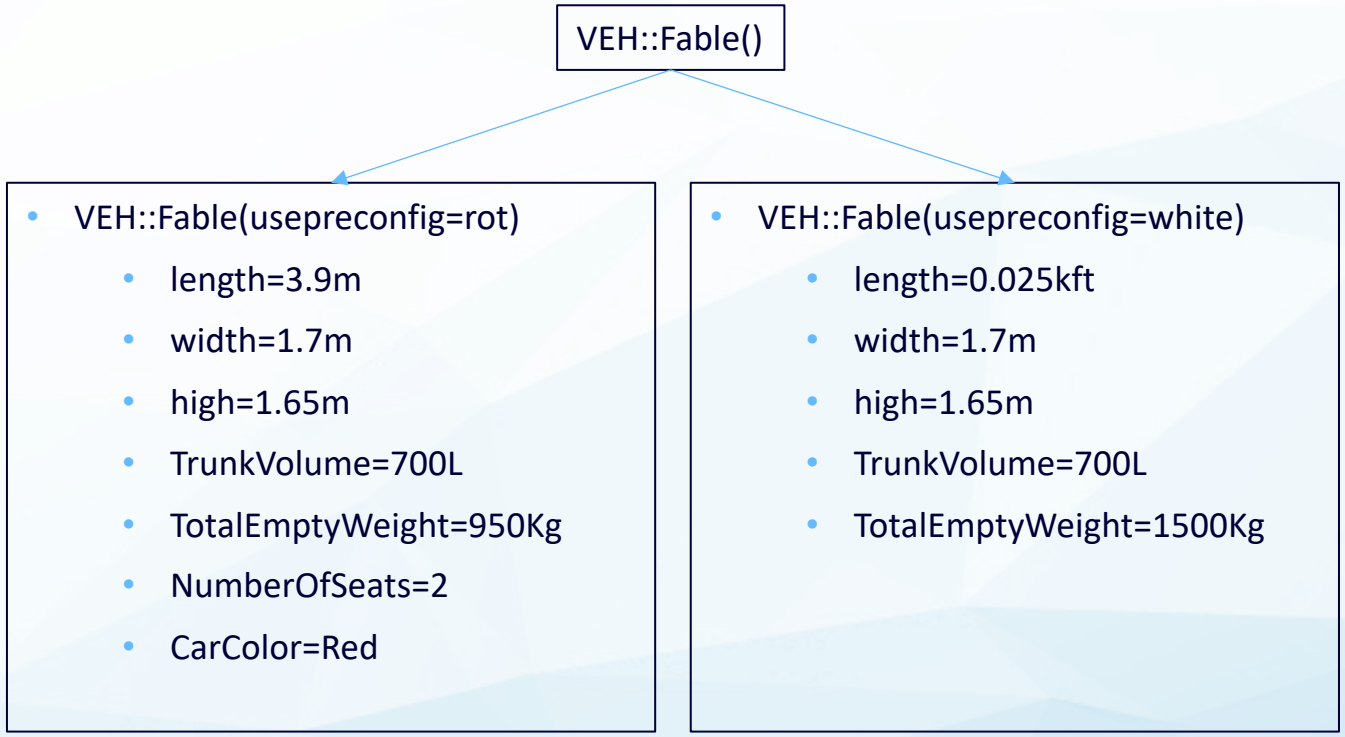
PIRO SYSTEMS ENGINEERING

Physical Unit validation and manipulation	Use comparison module	Dynamics & combinatorics
Check physical units of parameters – modify physical units to basic units evaluate scaling transformations generate error messages in case unit errors	Find comparable parameter generations – generate warning messages in case of different values	Apply Dynamics to parameters with effects to parameter values and ranges Generate combinatorics out of different dynamics

Automatic Analysis

Configs for a generic system element

Define configurations for a generic system element.



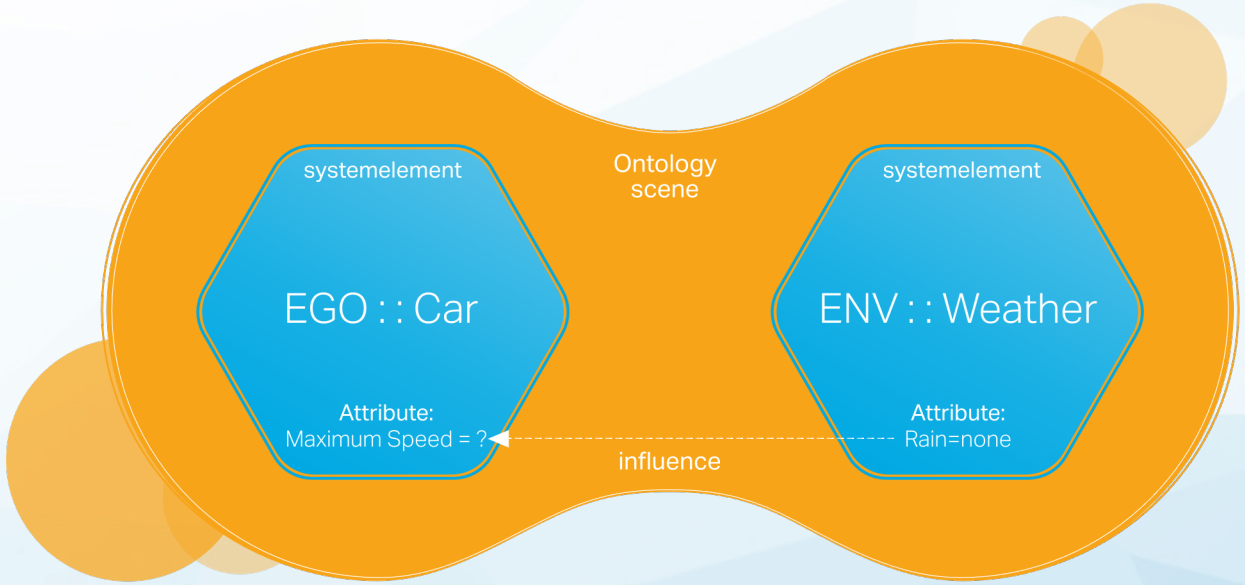
Automatic Analysis

Influences

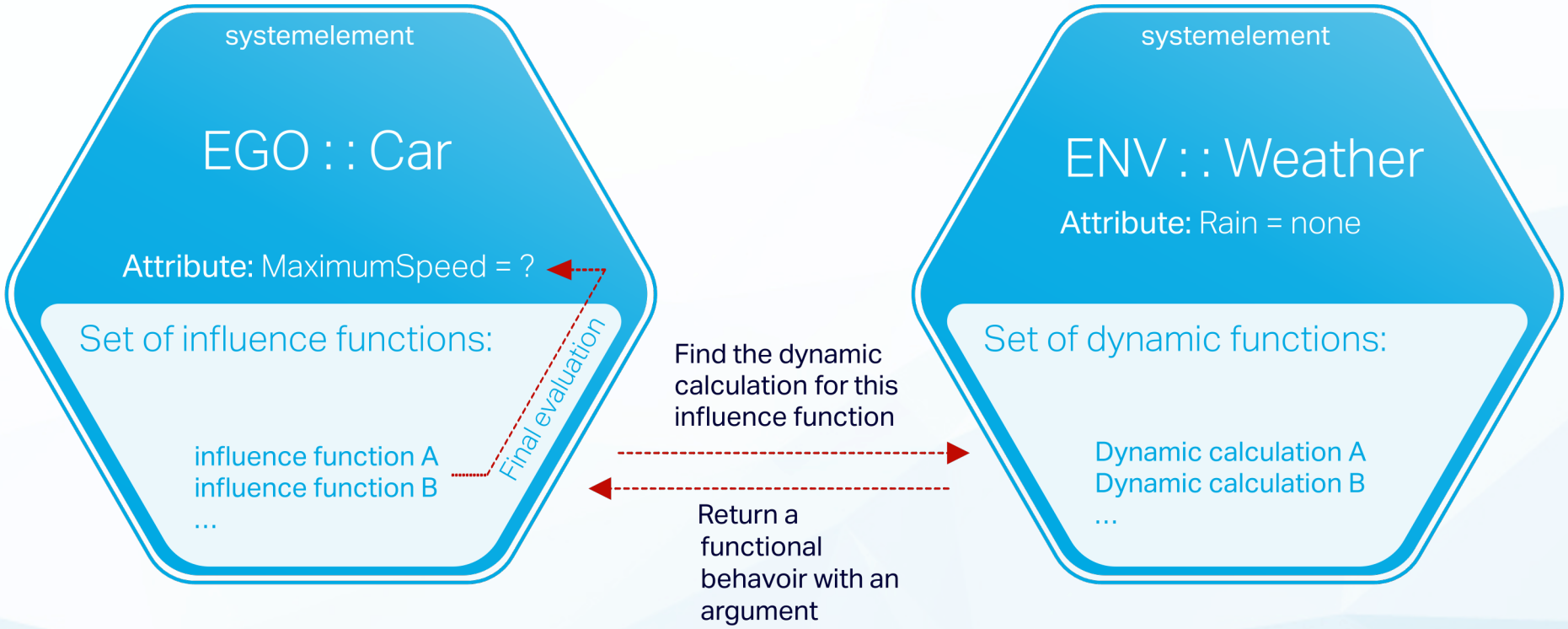
Apply OntologyScene Influences /Dynamics to parameters with effects to parameter values and ranges

Different attributes of different systemelements may have some interaction.

This means: If we want to get the current value of a single attribute of a given systemelement we have to respect influences from other attributes of other systemelements



Interaction between Attributes



To do this we start with an initial (default) value of the attribute where we want to know the value. This initial value can be overwritten or modified by the functional behavoir that comes from the influence function.

ENV::DangerousObject(type=Wolf) —influences→ VEH::Rot(ForestSpeed)

? EGO::Driver(DriverState=alive) ?

© 2020 PiRo Systems Engineering GmbH
Der Inhalt dieses Handouts ist durch das Urheberrecht geschützt. Er darf ohne vorherige Zustimmung des Urhebers weder ganz noch teilweise kopiert, veröffentlicht oder verändert werden. Die Informationen und Konzepte sind vertraulich und dürfen ohne vorherige Zustimmung des Urhebers nicht weitergegeben oder auf irgendeine Weise weiterverwendet werden.