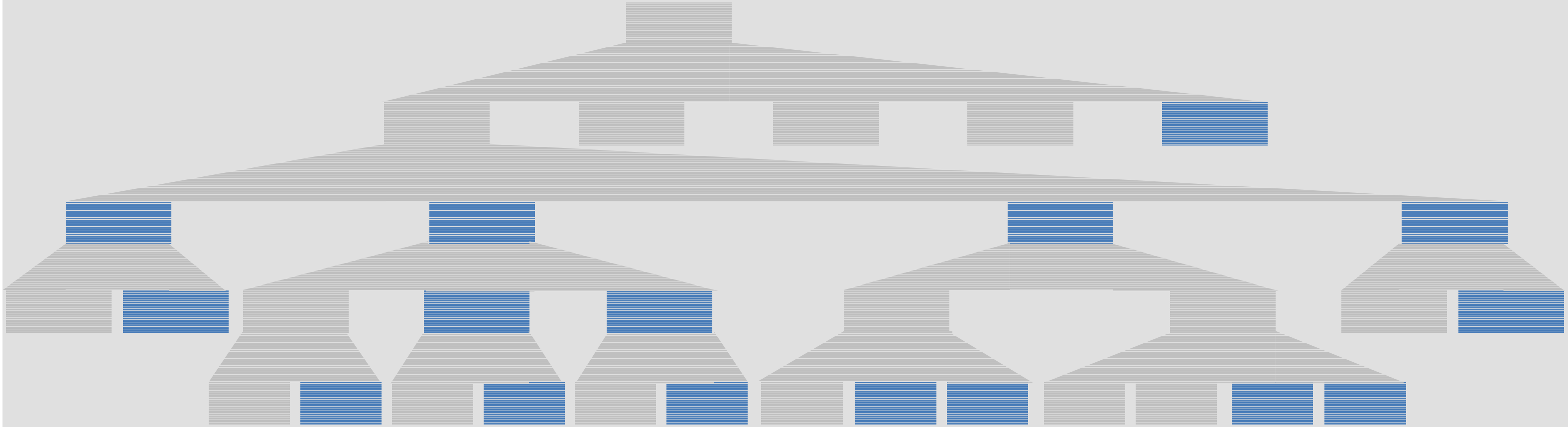
ConIPF: Configuration in Industrial Product Families

BOSCH



SPLC3

Boston, Massachusetts, USA

August 31st, 2004

John MacGregor, Robert Bosch GmbH

CONIPF

Agenda

I. The ConIPF Project

II. CPS: The Product Family used for the Demonstration

III. Configuration Models Used in The Demonstration

IV. Demonstration Platform Elements

V. Demonstration

Mission:

The ConIPF Project has defined a **METHODOLOGY** for

FEATURE BASED PRODUCT DERIVATION

where:

- The features specify the Capabilities of the product

and

- The corresponding product *artefacts* are determined through Structure-Based Configuration

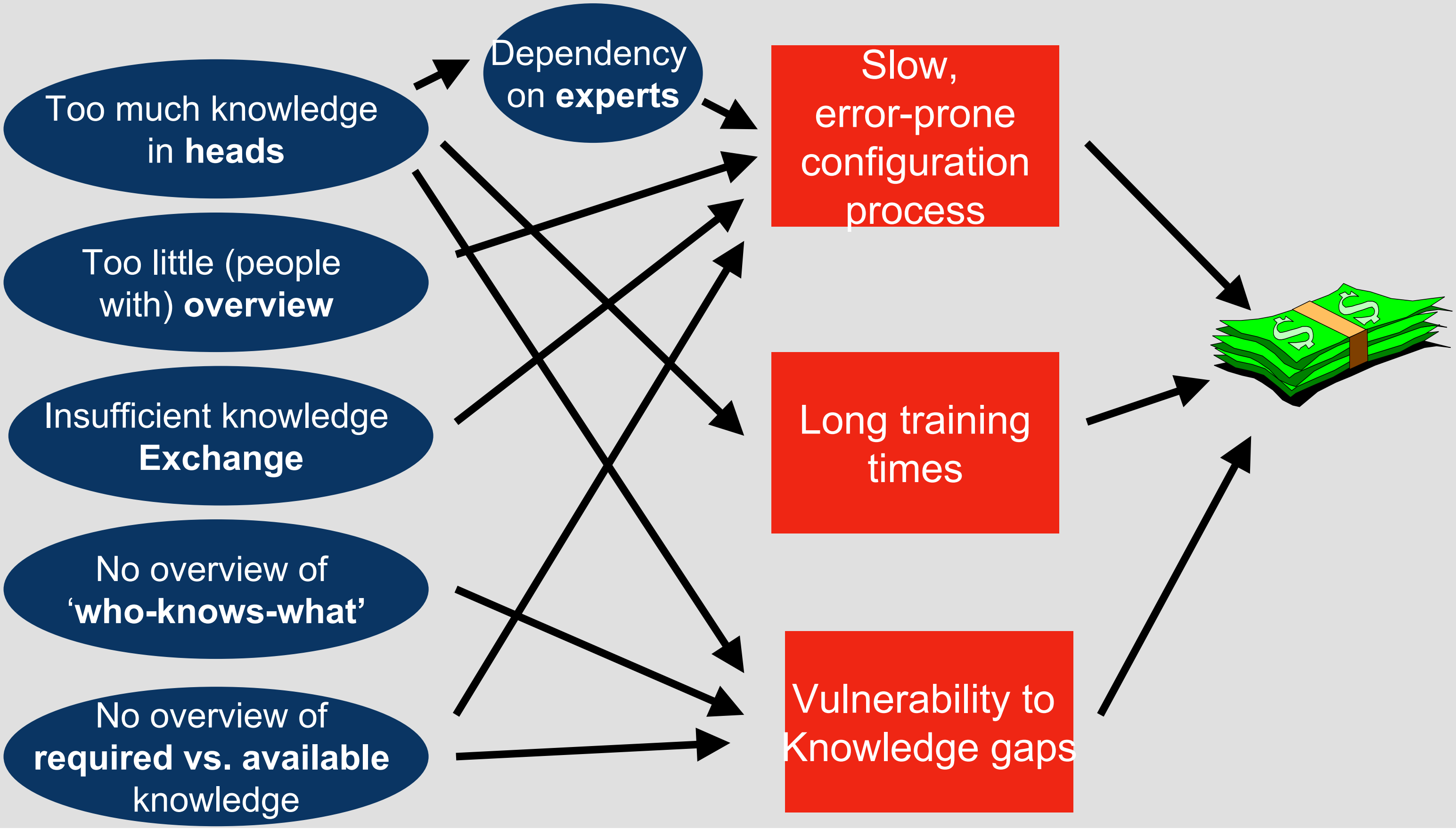
Industrial Complexity

➔ Staggering Complexity:

- Variability
  - Thousands of Products
  - (Tens of) Thousands of Components
  - (Tens/Hundreds of) Parameters per Component

- Combinability
  - Cardinality (Mandatory, Optional, Alternative, Limited Number)
  - Interdependence (Excludes, Requires,…)

Organisational Situation



Too much knowledge in **heads**

Too little (people with) **overview**

Insufficient knowledge **Exchange**

No overview of **‘who-knows-what’**

No overview of **required vs. available** knowledge

Dependency on **experts**

Slow, error-prone configuration process

Long training times

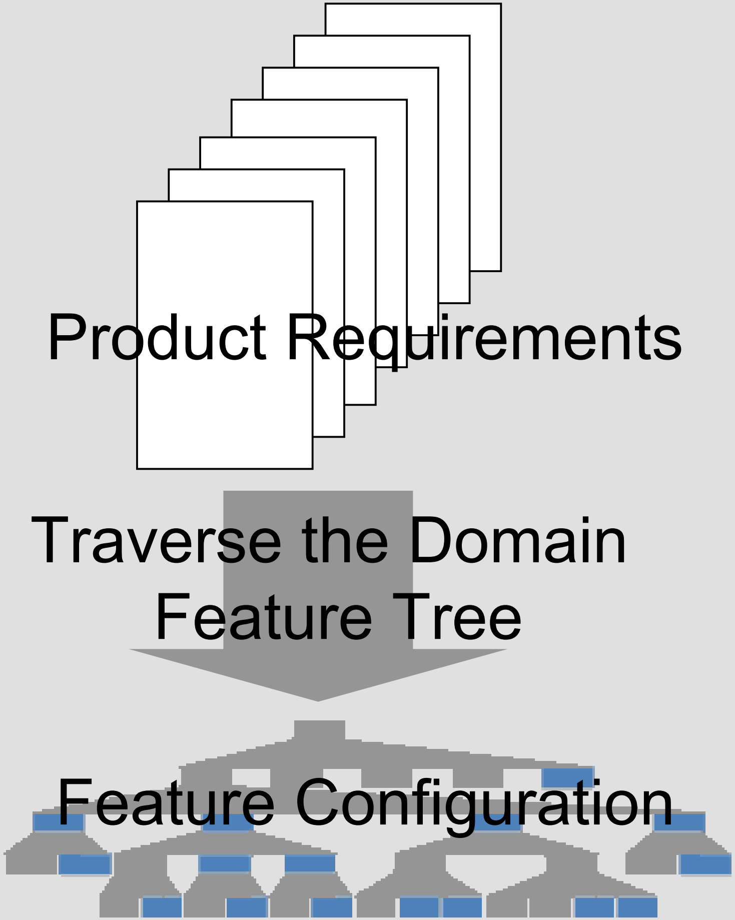
Vulnerability to Knowledge gaps

The ConIPF Approach (1/2)

- The systems engineer maps product requirements, which are formulated from the customer perspective, to features in the domain feature tree.

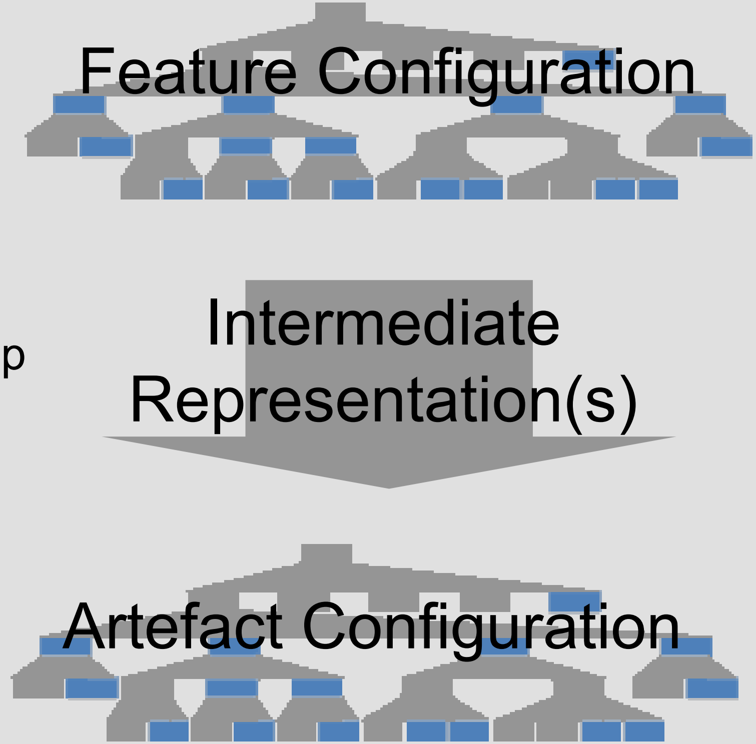
- That is, the engineer traverses the domain feature tree, selecting generic features and assigning them values.

- This ultimately results in the feature configuration for a particular product



The ConIPF Approach (2/2)

- An inference machine supports the configuration process to ensure consistency, correctness and completeness.
- Feature configurations do not usually map directly to the artefact configurations that realise the features.
- Intermediate knowledge bases are therefore be needed to transform the features into the solution.



Configuration

Meaning of the word:

→ An Action – the process of combining or arranging

→ As well as a Result – a list of necessary components

→ Relationship to Configuration Management

- Combination – „make“or IDE project description
- Lists – describe a deliverable that is stored and managed in a configuration management system

Configuration Knowledge

Configuring (as AI discipline) : *

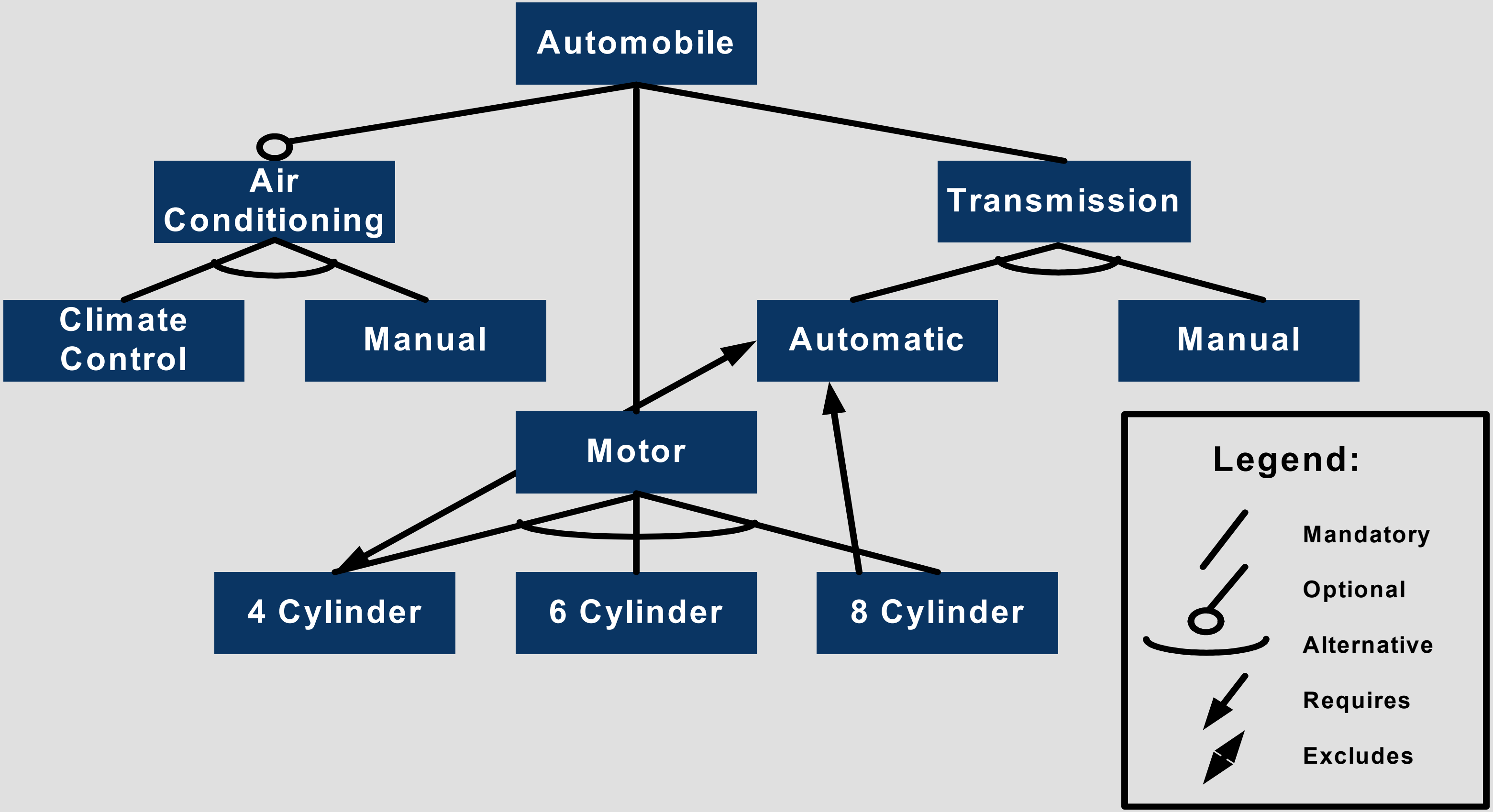
Assembly of a technical system from individual parameterisable objects to a configuration that fulfils a certain task (or purpose)

Based on:

- Descriptions of objects and their attributes (concept hierarchy)
- Relations/restrictions between the objects
- Knowledge over the solution procedure (control knowledge)
- A description of the goal to be fulfilled

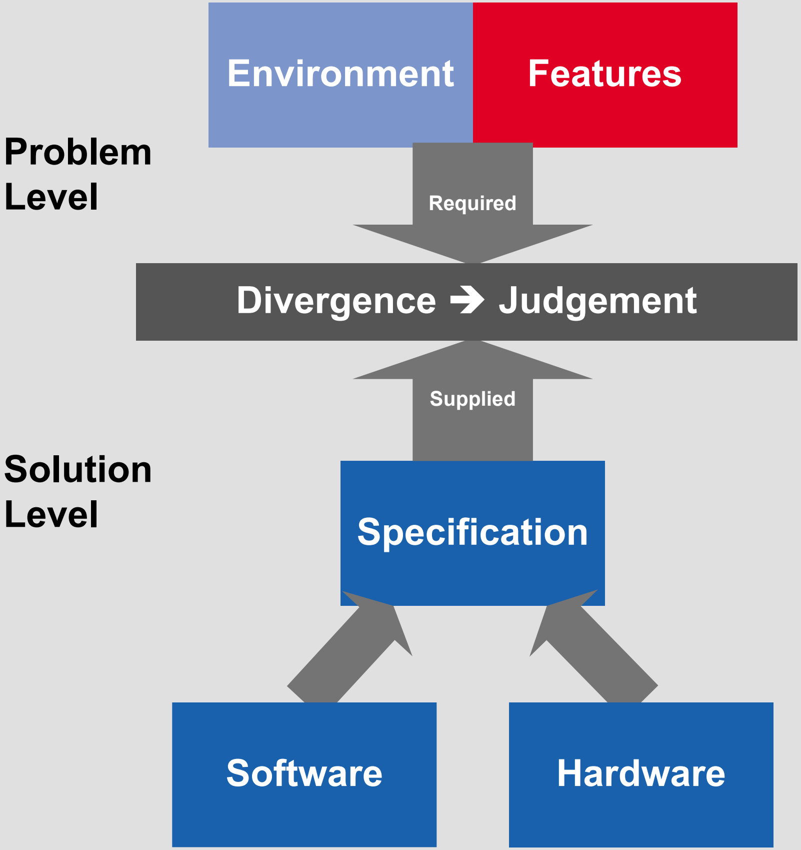
* A. Günter, University of Hamburg

Paradox: Feature Tree or Structure Description?



The Product Configurer’s Problem

- Both the target environment and the desired features constrain the choice of components
- The combined capabilities of the components must meet or exceed the capabilities specified by the features
- The components that are selected dictate the capabilities of the system



Two Levels of Configuration

- ➔ Problem Level: Selection of Capabilities
  - The user configures the product capabilities and the configuration system determines which components are needed
  - The level preferred by ConIPF
- ➔ Solution Level: Selection of Components
  - The user is responsible for ensuring that the capabilities of the selected components meet the product requirements
  - Traditional configuration
- ➔ ConIPF supports both levels
  - (Features can be defined either on the problem or solution levels)

Features

→ Software Engineering Viewpoint:

are Aggregates of Requirements

→ ConIPF (structure-based configuration) Viewpoint

are Concepts with Attributes (not binary)

Features Types

➔ Capability Features (Problem Level)

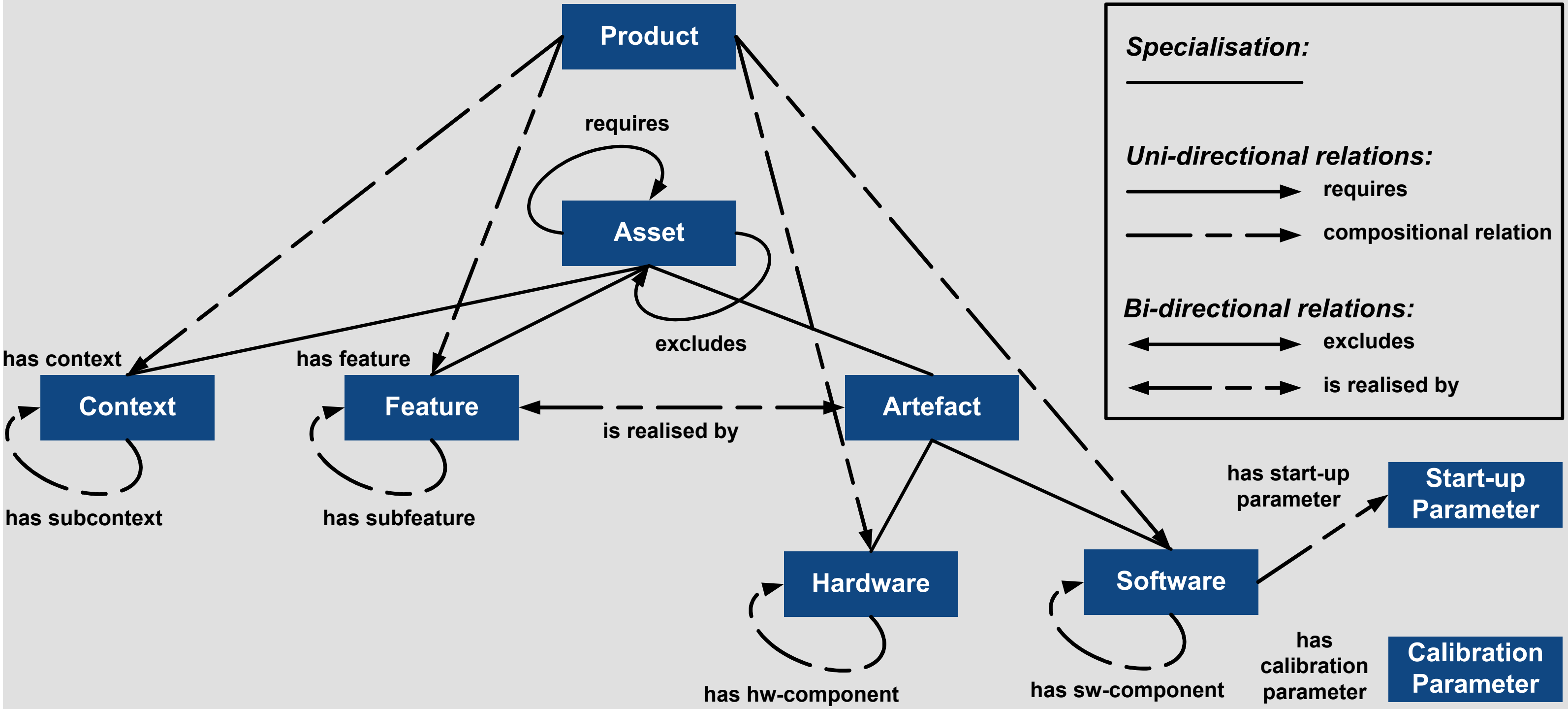
- Describe product capabilities, their attributes and extents
- Current Hypothesis:
  - Capability Features are the functionalities in the system
  - The attributes describe the feature's non-functional aspects
  - Example:
    Feature: Air-conditioning
    Attributes: Time to cool car, Effect on gas consumption

➔ Product Aspect Features (Solution Level)

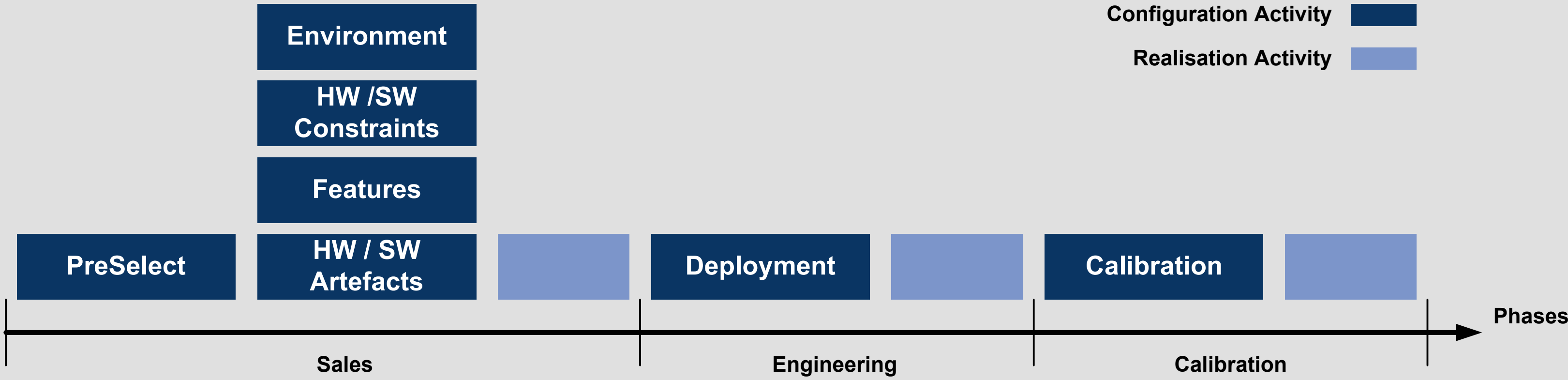
- Relate to requirements (customer preferences, perhaps)
- Example: The number and type of knobs to control the a/c

➔ Perhaps more types

The ConIPF Common Application Model (CAM)



The ConIPF Derivation Process


Configuration Activity ▇
Realisation Activity ▇

Environment

HW /SW Constraints

Features

PreSelect

HW / SW Artefacts

Deployment

Calibration

Phases

Sales

Engineering

Calibration

Process Elements

→ Direct Derivation

- The capabilities of available artefacts match those of the features chosen

→ Calibration

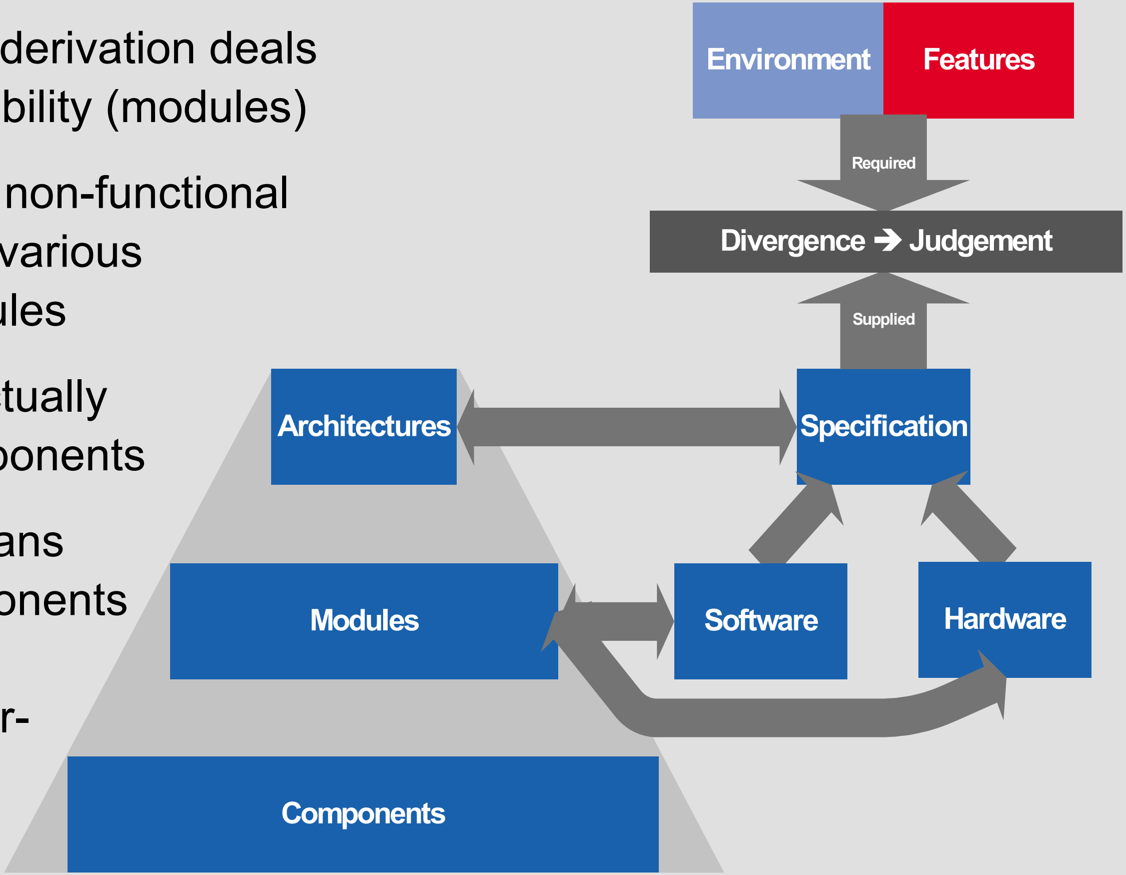
- Where the derived configuration is fine-tuned
  - Overengineered configurations are slimmed down
  - Underengineered configurations are beefed up

→ Evolution

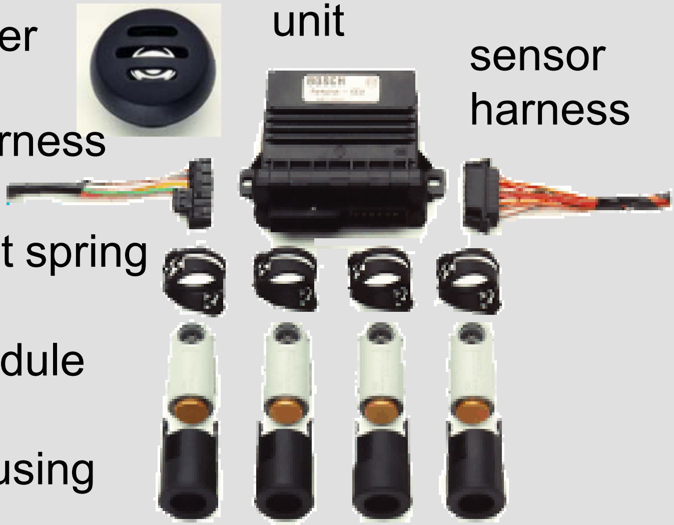
- The capabilities available artefacts do not match those of the features chosen
  - → New Development
  - → Update the Asset Repository, Configuration Model
  - → Repeat Direct Derivation

Evolution Considerations

- ➔ Fundamentally, direct derivation deals with packages of variability (modules)
- ➔ Architectures address non-functional requirements through various arrangements of modules
- ➔ These modules are actually configurations of components
- ➔ New development means developing new components and configuring new packages or reconfiguring existing packages



Car Periphery Systems



loudspeaker

control unit

sensor harness

system harness

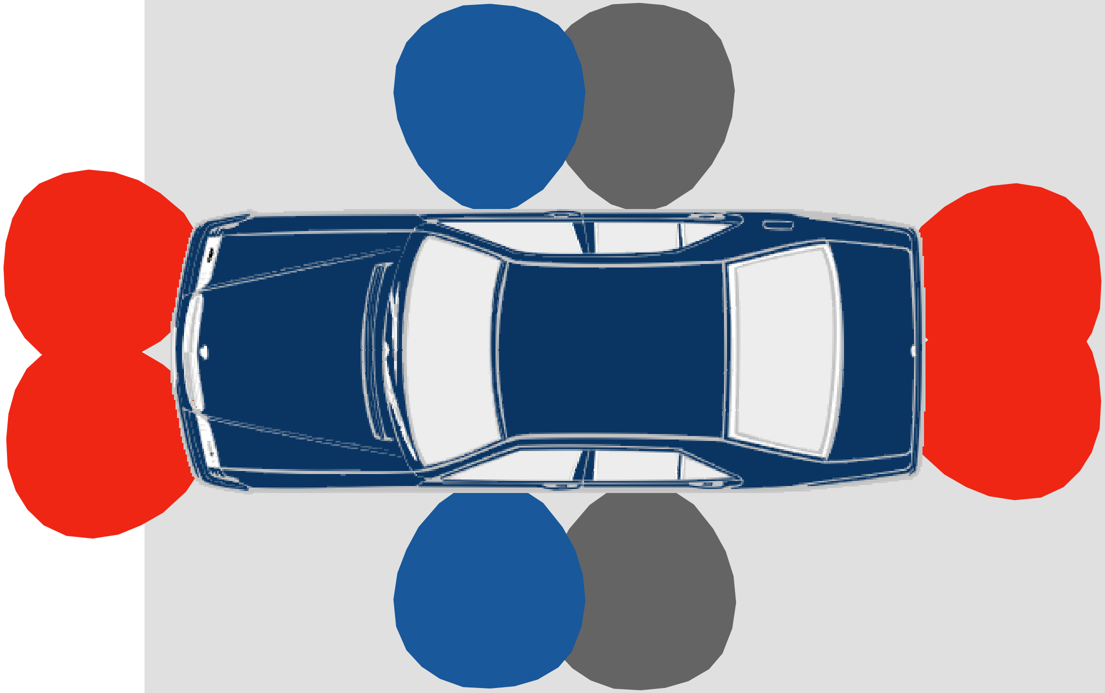
attachment spring

sensor module

sensor housing

Parking Assistance System

The CPS Product Family

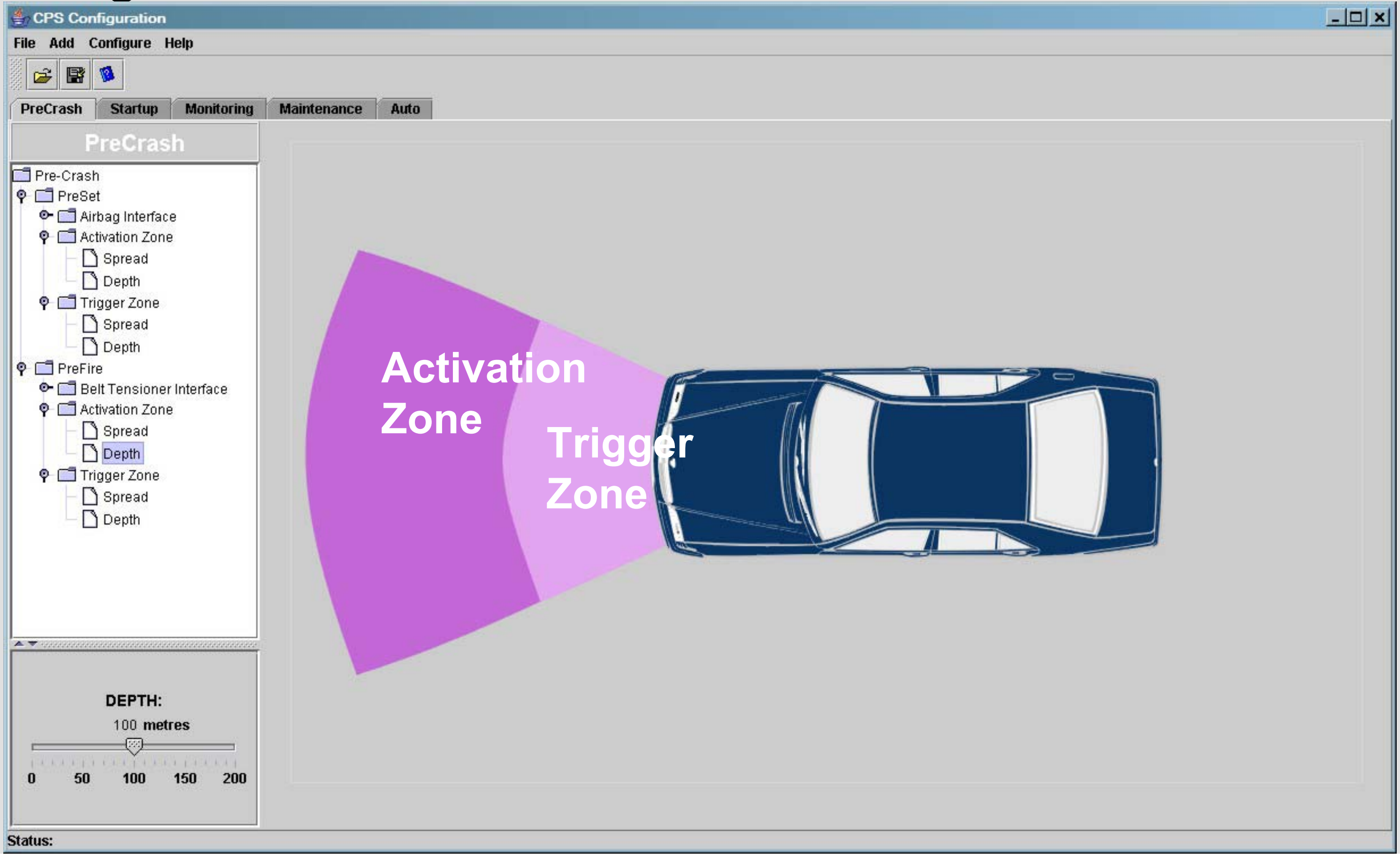


- 🟥 Parking Assistance
- 🟥 Pre-Crash Detection Front
- 🟥 Semiautomatic Go
- 🟥 ACC Stop & Go
- 🟦 Parking Spot Detection
- 🟦 Pre-Crash Detection Front & Side
- ⬛ Blind Spot Detection
- ⬛ Semi-automatic Parking
- ⬛ Pre-Crash Detection Side & Rear

CPS Products Used in the Demo

- Parking Support
  - Monitors the distance to objects while the vehicle is parking
    - Displays the distance to the object, or
    - Sounds alarms when boundaries are crossed
- PreCrash Applications: detect an imminent crash
  - Preset: Sensitises the airbag sensor
  - PreFire: Fires a (seat)belt tensioner

Configuration Front-End for PreCrash



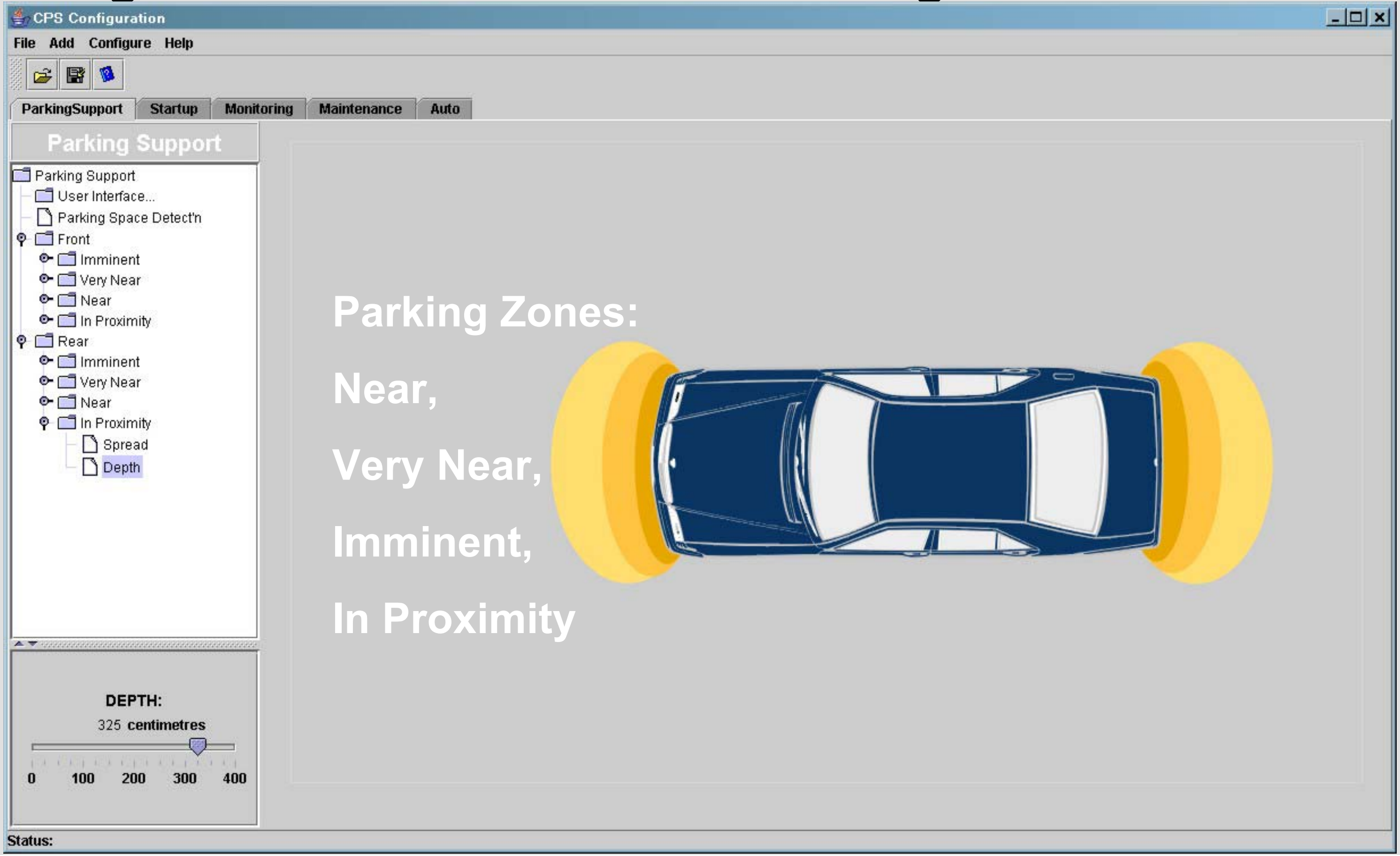
- Pre-Crash
  - PreSet
    - Airbag Interface
    - Activation Zone
      - Spread
      - Depth
    - Trigger Zone
      - Spread
      - Depth
  - PreFire
    - Belt Tensioner Interface
    - Activation Zone
      - Spread
      - Depth
    - Trigger Zone
      - Spread
      - Depth

DEPTH:

100 metres

0 50 100 150 200

Configuration Front-End for Parking Assistance



DEPTH:
325 centimetres

Parking Zones:

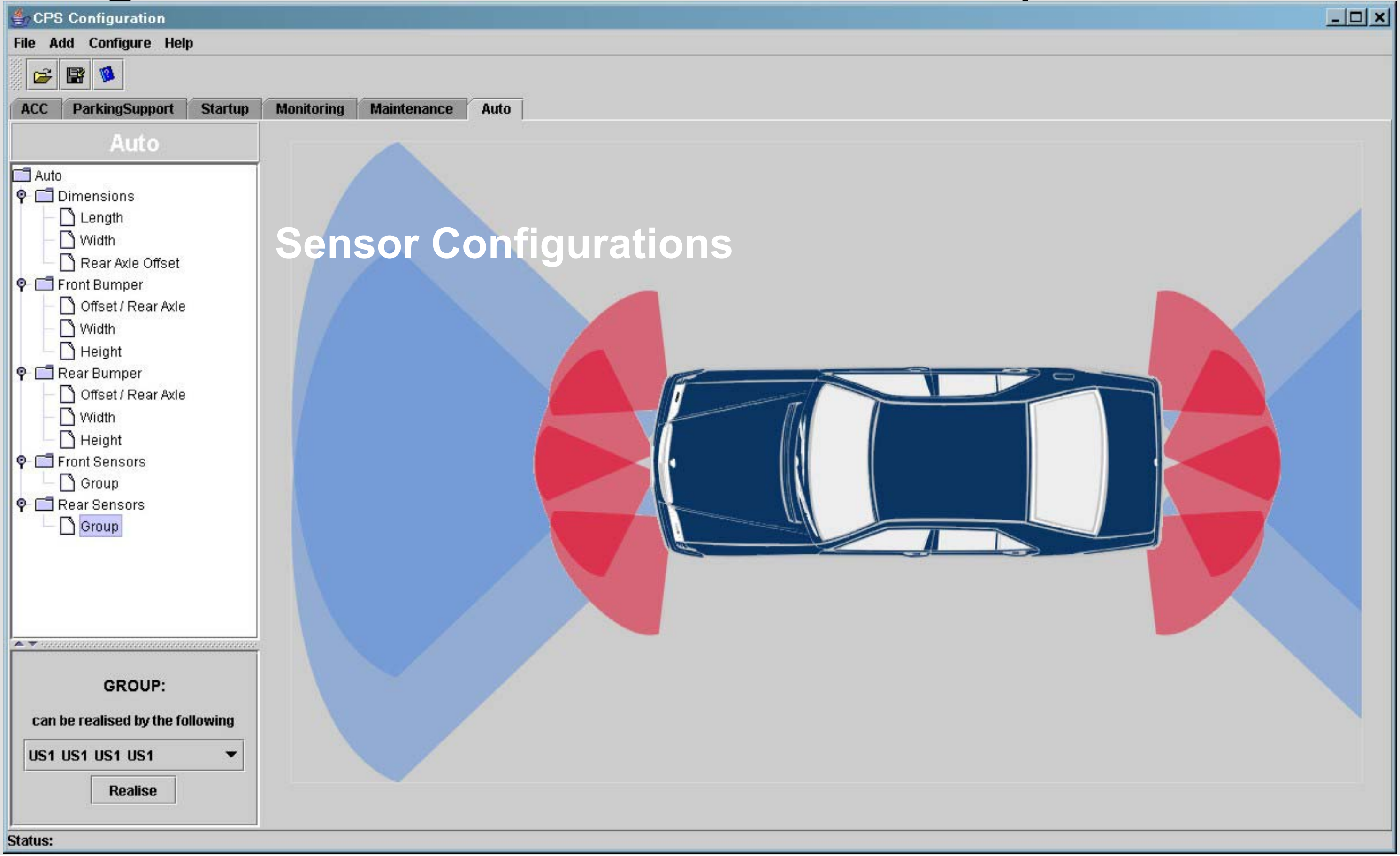
Near,

Very Near,

Imminent,

In Proximity

Configuration Front-End for Solution Space



- Auto
  - Dimensions
    - Length
    - Width
    - Rear Axle Offset
  - Front Bumper
    - Offset / Rear Axle
    - Width
    - Height
  - Rear Bumper
    - Offset / Rear Axle
    - Width
    - Height
  - Front Sensors
    - Group
  - Rear Sensors
    - Group

GROUP:

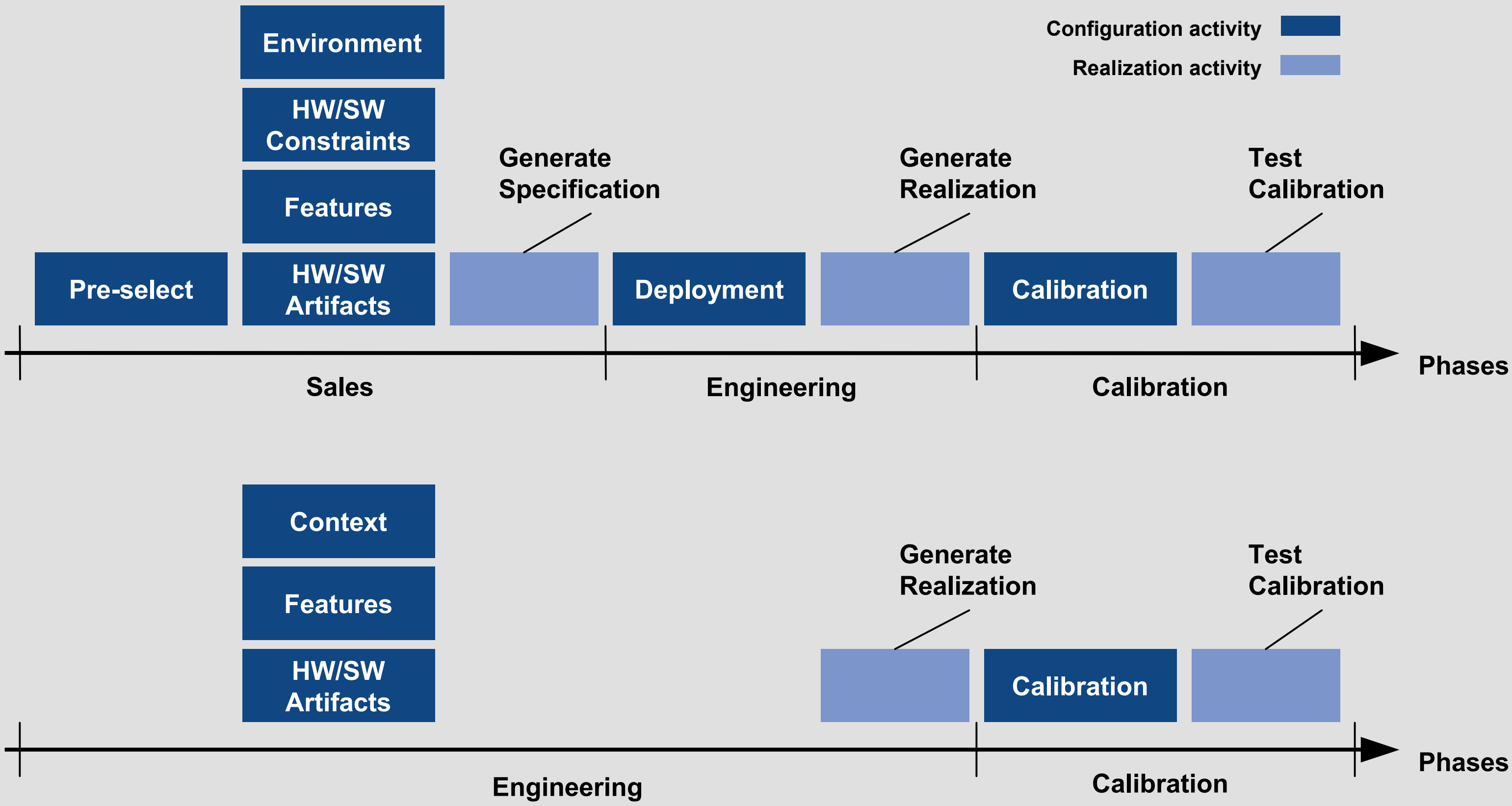
can be realised by the following

US1 US1 US1 US1

Realise

Status:

Bosch's Instantiation of the ConIPF Process



Configuration activity 🟦

Realization activity ▭

Environment

HW/SW Constraints

Features

Pre-select

HW/SW Artifacts

Generate Specification

Deployment

Generate Realization

Calibration

Test Calibration

Sales

Engineering

Calibration

Phases

Context

Features

HW/SW Artifacts

Generate Realization

Calibration

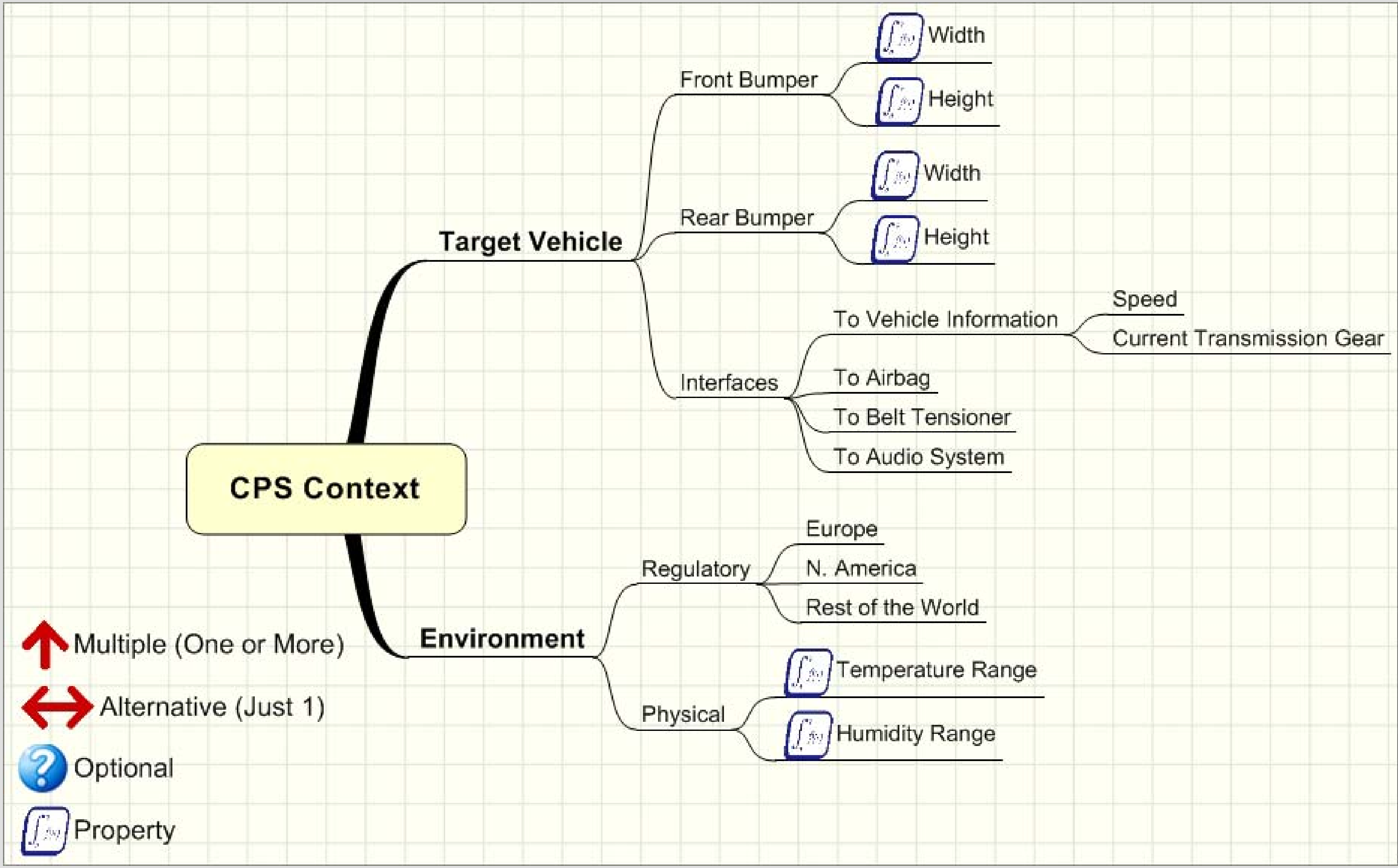
Test Calibration

Engineering

Calibration

Phases

III. Configuration Models



CPS Context

Target Vehicle

Front Bumper

Width

Height

Rear Bumper

Width

Height

Interfaces

To Vehicle Information

Speed

Current Transmission Gear

To Airbag

To Belt Tensioner

To Audio System

Environment

Regulatory

Europe

N. America

Rest of the World

Physical

Temperature Range

Humidity Range

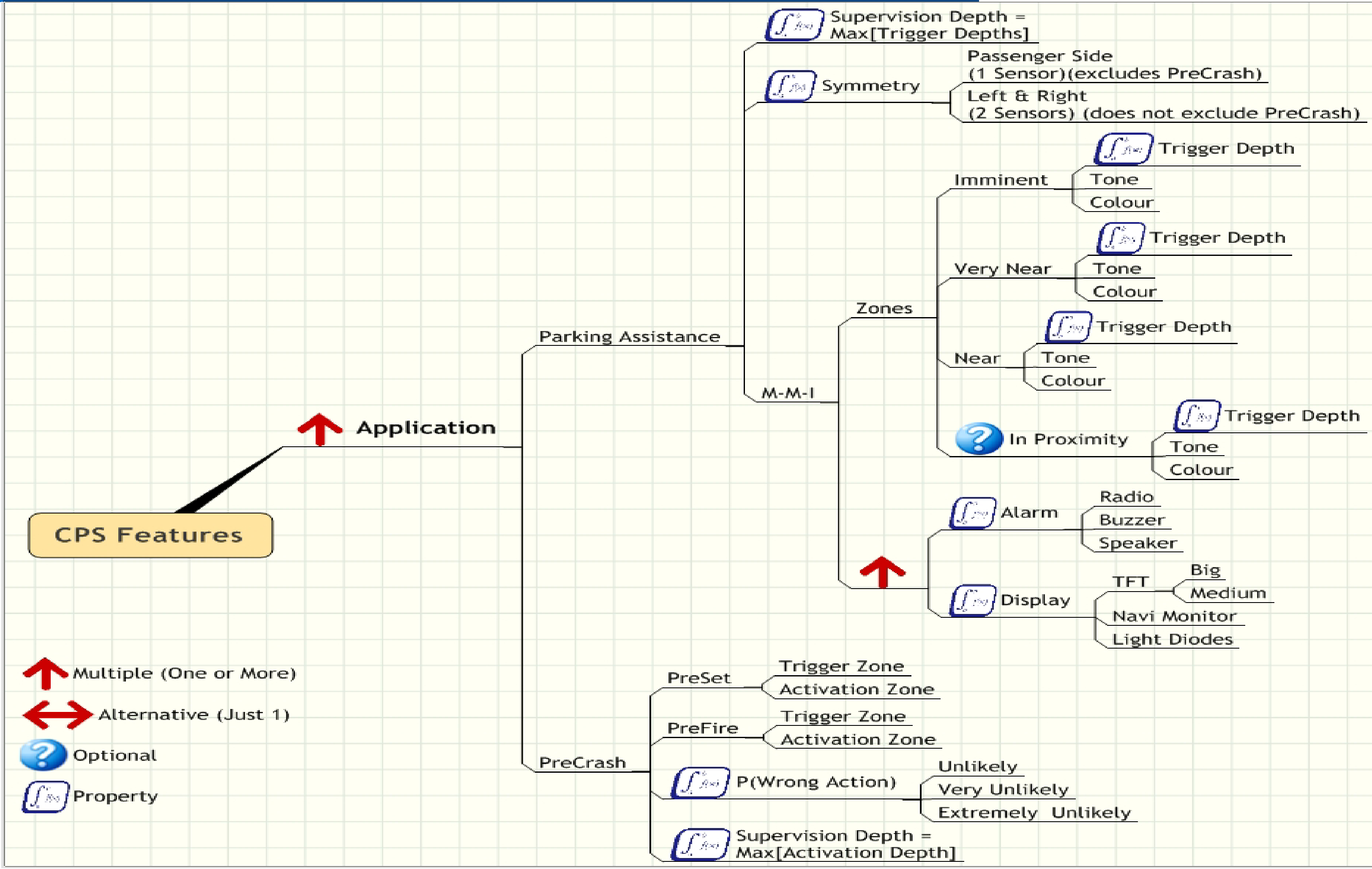
Multiple (One or More)

Alternative (Just 1)

Optional

Property

III. Configuration Models



- CPS Features
  - ↑ Application
    - Parking Assistance
      - ∫ Supervision Depth = Max[Trigger Depths]
      - ∫ Symmetry
        - Passenger Side (1 Sensor)(excludes PreCrash)
        - Left & Right (2 Sensors) (does not exclude PreCrash)
      - M-M-I
        - Zones
          - Imminent
            - ∫ Trigger Depth
            - Tone
            - Colour
          - Very Near
            - ∫ Trigger Depth
            - Tone
            - Colour
          - Near
            - ∫ Trigger Depth
            - Tone
            - Colour
          - ? In Proximity
            - ∫ Trigger Depth
            - Tone
            - Colour
        - ↑
          - ∫ Alarm
            - Radio
            - Buzzer
            - Speaker
          - ∫ Display
            - TFT
              - Big
              - Medium
            - Navi Monitor
            - Light Diodes
    - PreCrash
      - PreSet
        - Trigger Zone
        - Activation Zone
      - PreFire
        - Trigger Zone
        - Activation Zone
      - ∫ P(Wrong Action)
        - Unlikely
        - Very Unlikely
        - Extremely Unlikely
      - ∫ Supervision Depth = Max[Activation Depth]

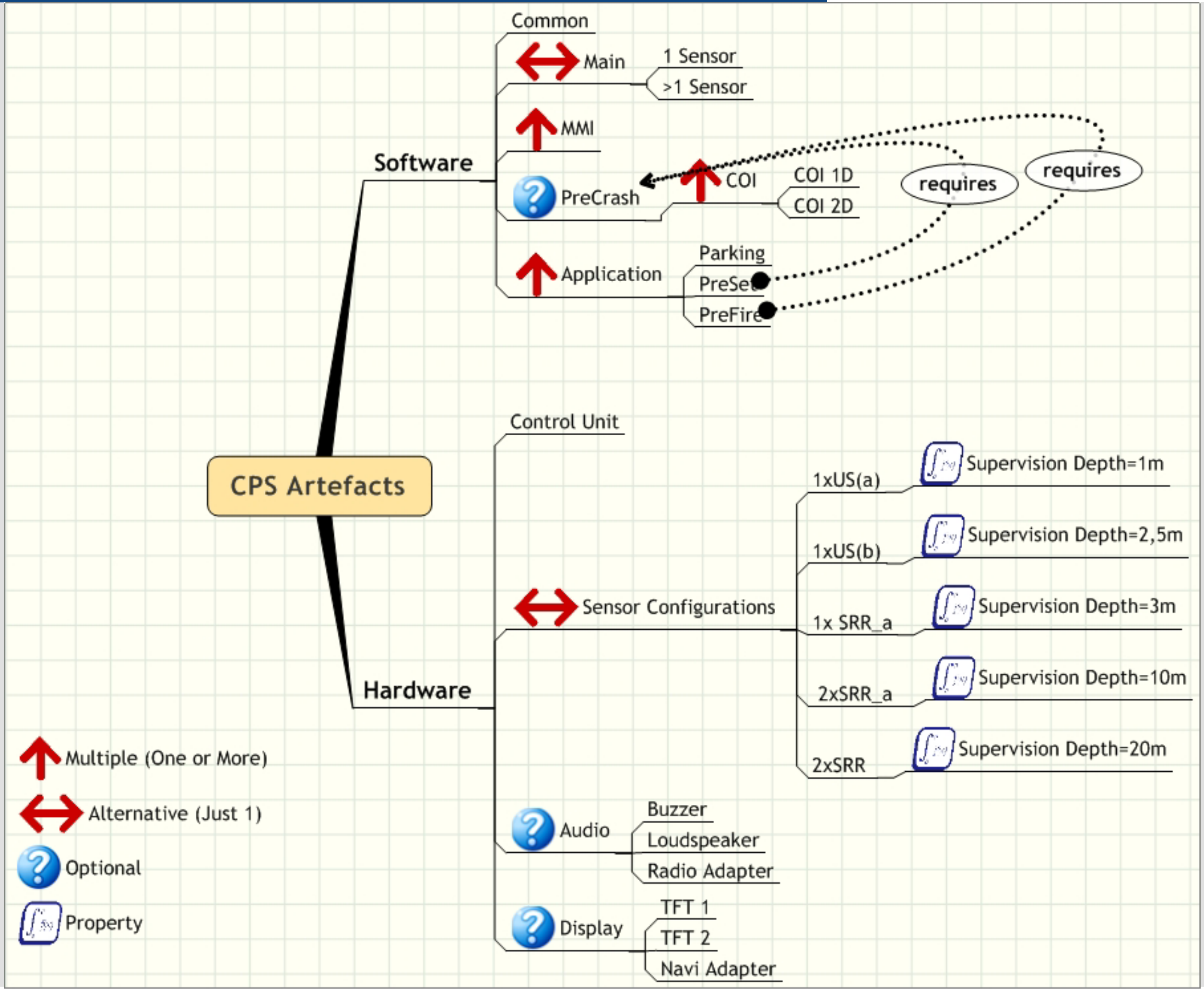
↑ Multiple (One or More)

↔ Alternative (Just 1)

? Optional

∫ Property

III. Configuration Models



CPS Artefacts

Software

Common

↔ Main

1 Sensor

>1 Sensor

↑ MMI

? PreCrash

↑ COI

COI 1D

COI 2D

↑ Application

Parking

PreSet

PreFire

requires

requires

Hardware

Control Unit

↔ Sensor Configurations

1xUS(a)

Supervision Depth=1m

1xUS(b)

Supervision Depth=2,5m

1x SRR_a

Supervision Depth=3m

2xSRR_a

Supervision Depth=10m

2xSRR

Supervision Depth=20m

? Audio

Buzzer

Loudspeaker

Radio Adapter

? Display

TFT 1

TFT 2

Navi Adapter

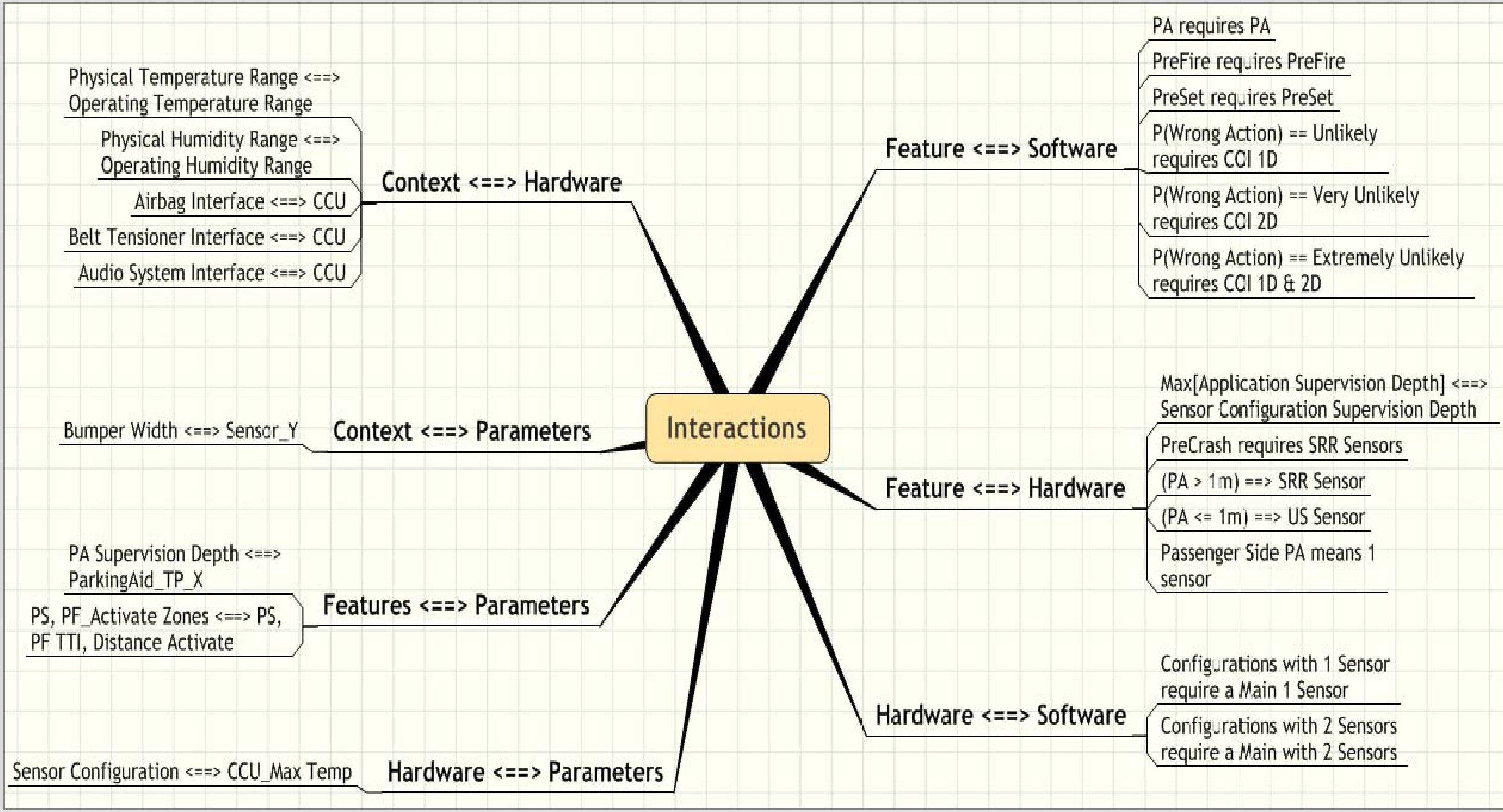
↑ Multiple (One or More)

↔ Alternative (Just 1)

? Optional

∫ Property

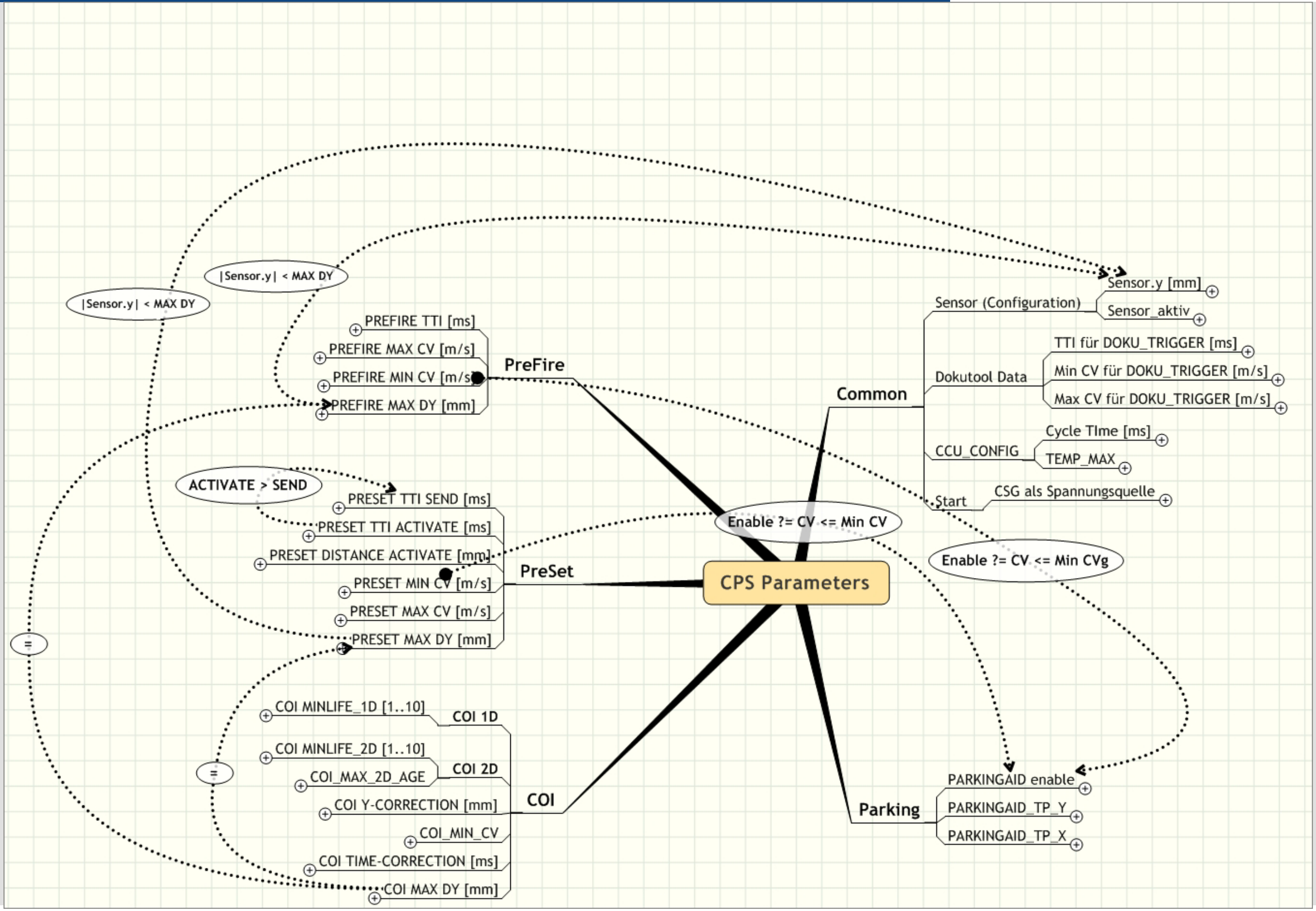
III. Configuration Models



Interactions

- Context <==> Hardware
  - Physical Temperature Range <==> Operating Temperature Range
  - Physical Humidity Range <==> Operating Humidity Range
  - Airbag Interface <==> CCU
  - Belt Tensioner Interface <==> CCU
  - Audio System Interface <==> CCU
- Context <==> Parameters
  - Bumper Width <==> Sensor_Y
- Features <==> Parameters
  - PA Supervision Depth <==> ParkingAid_TP_X
  - PS, PF_Activate Zones <==> PS, PF TTI, Distance Activate
- Hardware <==> Parameters
  - Sensor Configuration <==> CCU_Max Temp
- Feature <==> Software
  - PA requires PA
  - PreFire requires PreFire
  - PreSet requires PreSet
  - P(Wrong Action) == Unlikely requires COI 1D
  - P(Wrong Action) == Very Unlikely requires COI 2D
  - P(Wrong Action) == Extremely Unlikely requires COI 1D & 2D
- Feature <==> Hardware
  - Max[Application Supervision Depth] <==> Sensor Configuration Supervision Depth
  - PreCrash requires SRR Sensors
  - (PA > 1m) ==> SRR Sensor
  - (PA <= 1m) ==> US Sensor
  - Passenger Side PA means 1 sensor
- Hardware <==> Software
  - Configurations with 1 Sensor require a Main 1 Sensor
  - Configurations with 2 Sensors require a Main with 2 Sensors

III Configuration Models



CPS Parameters

PreFire
- PREFIRE TTI [ms]
- PREFIRE MAX CV [m/s]
- PREFIRE MIN CV [m/s]
- PREFIRE MAX DY [mm]

PreSet
- PRESET TTI SEND [ms]
- PRESET TTI ACTIVATE [ms]
- PRESET DISTANCE ACTIVATE [mm]
- PRESET MIN CV [m/s]
- PRESET MAX CV [m/s]
- PRESET MAX DY [mm]

COI
- COI 1D
  - COI MINLIFE_1D [1..10]
- COI 2D
  - COI MINLIFE_2D [1..10]
  - COI_MAX_2D_AGE
- COI Y-CORRECTION [mm]
- COI_MIN_CV
- COI TIME-CORRECTION [ms]
- COI MAX DY [mm]

Common
- Sensor (Configuration)
  - Sensor.y [mm]
  - Sensor_aktiv
- Dokutool Data
  - TTI für DOKU_TRIGGER [ms]
  - Min CV für DOKU_TRIGGER [m/s]
  - Max CV für DOKU_TRIGGER [m/s]
- CCU_CONFIG
  - Cycle TIme [ms]
  - TEMP_MAX
- Start
  - CSG als Spannungsquelle

Parking
- PARKINGAID enable
- PARKINGAID_TP_Y
- PARKINGAID_TP_X

|Sensor.y| < MAX DY

|Sensor.y| < MAX DY

ACTIVATE > SEND

=

=

Enable ?= CV <= Min CV

Enable ?= CV <= Min CVg

Demo Target Platform

→ Realistic - Compatible with „real“ applications

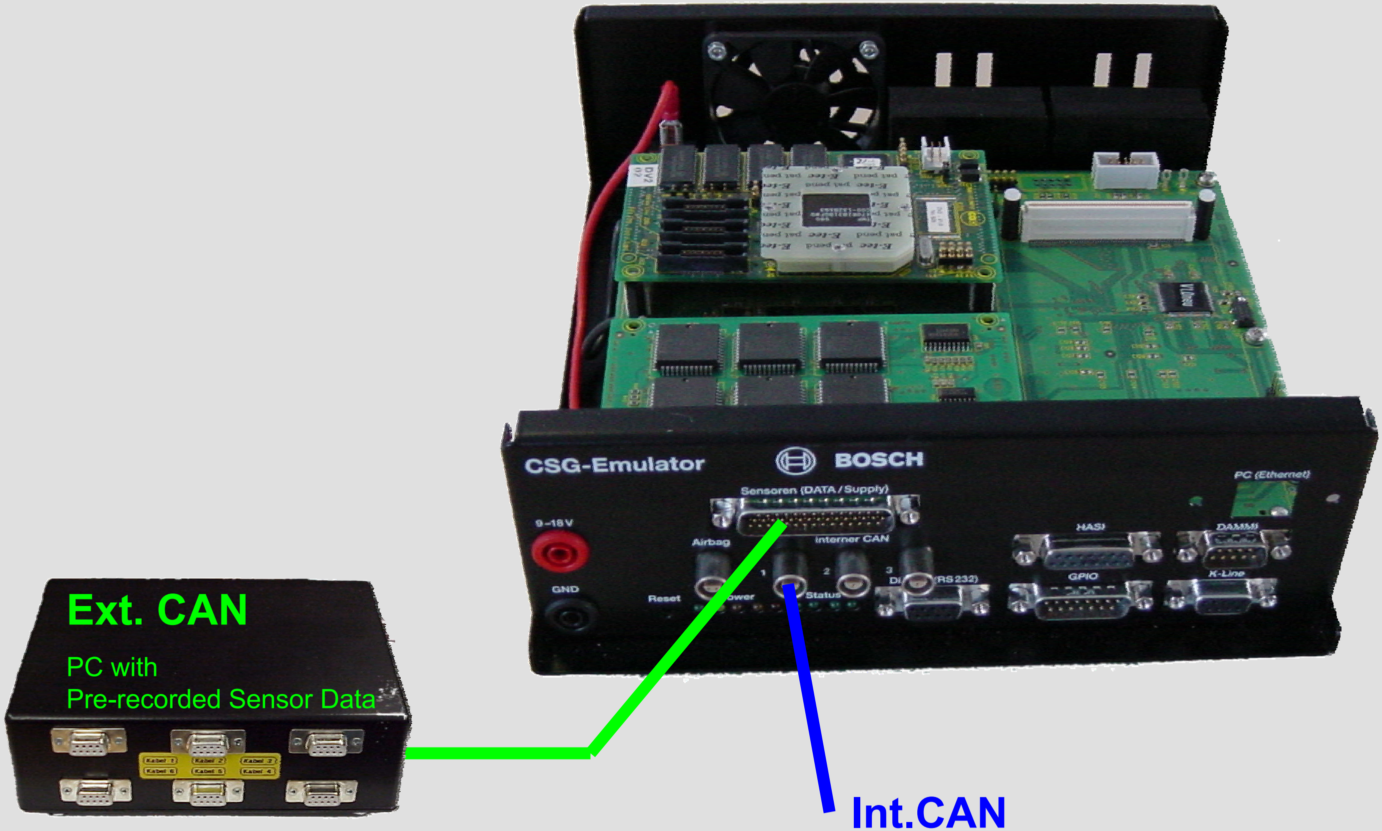
→ Hardware Platform:

- Control Unit running with reduced CPU frequency
- PC with pre-recorded sensor signals as input
- PC with Labview Simulation of MMI as output for Parking Support

→ Software Platform

- SRR Sensor Platform
- Parking Assistance, PreSet, Prefire

Target Control Unit



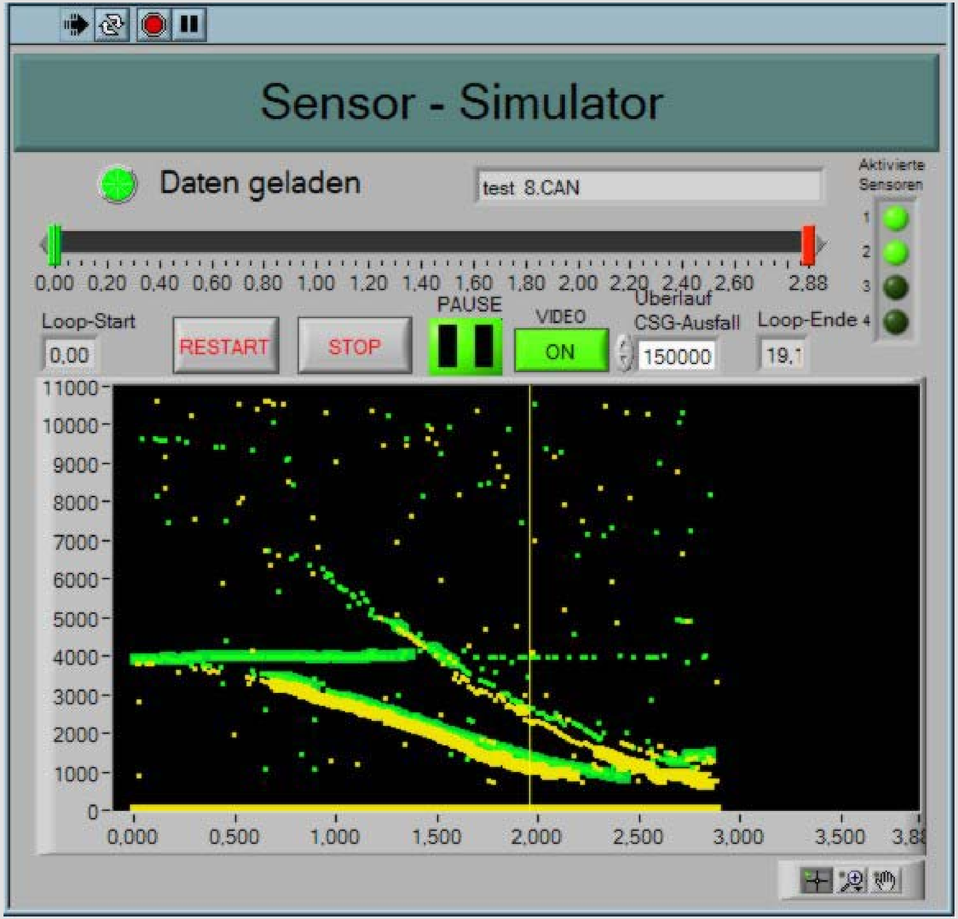
Ext. CAN

PC with
Pre-recorded Sensor Data

Int.CAN

PC with MMI in Labview

Pre-recorded Sensor Data



Sensor - Simulator

Daten geladen

test 8.CAN

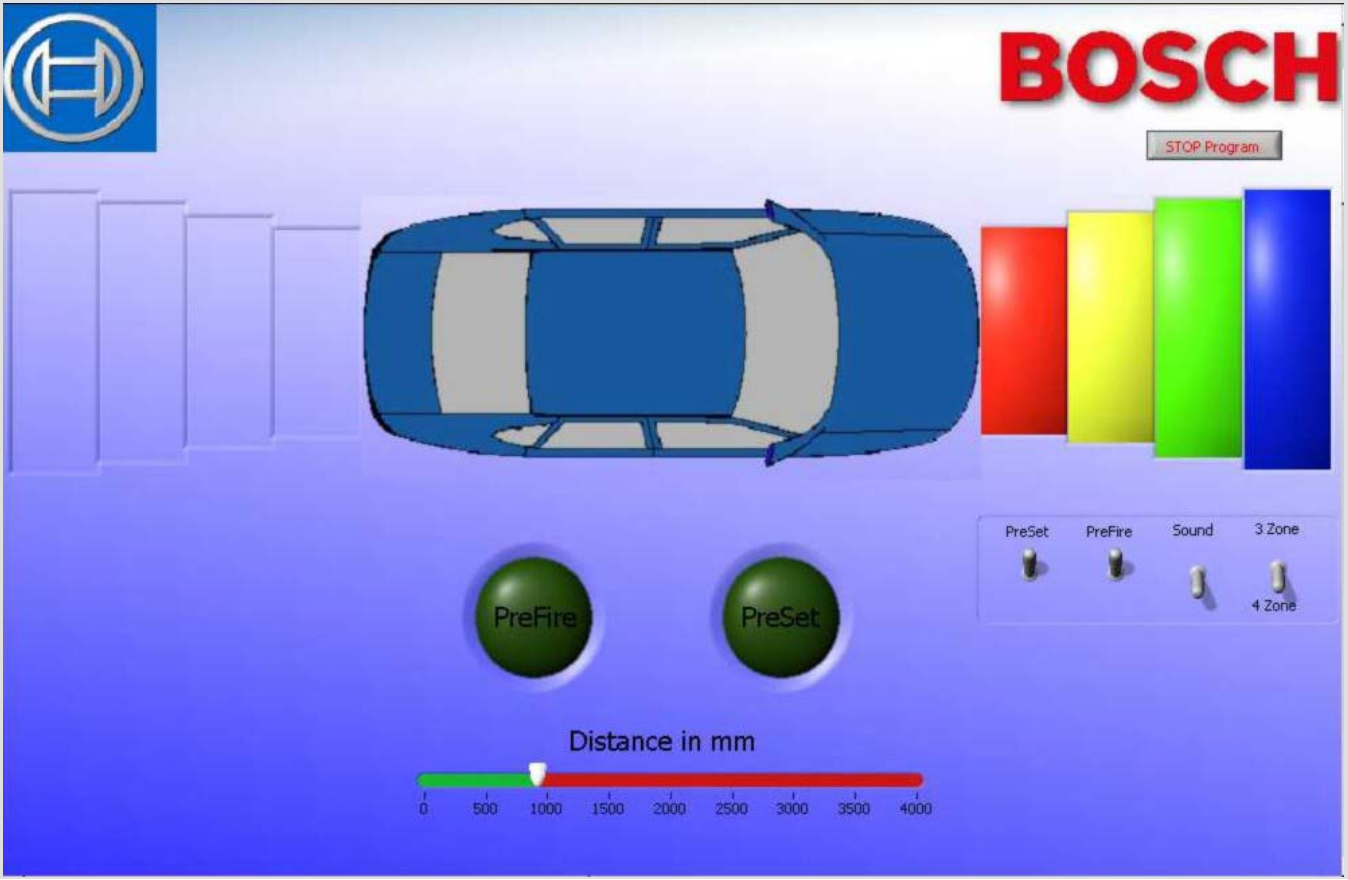
Loop-Start: 0,00

Überlauf CSG-Ausfall: 150000

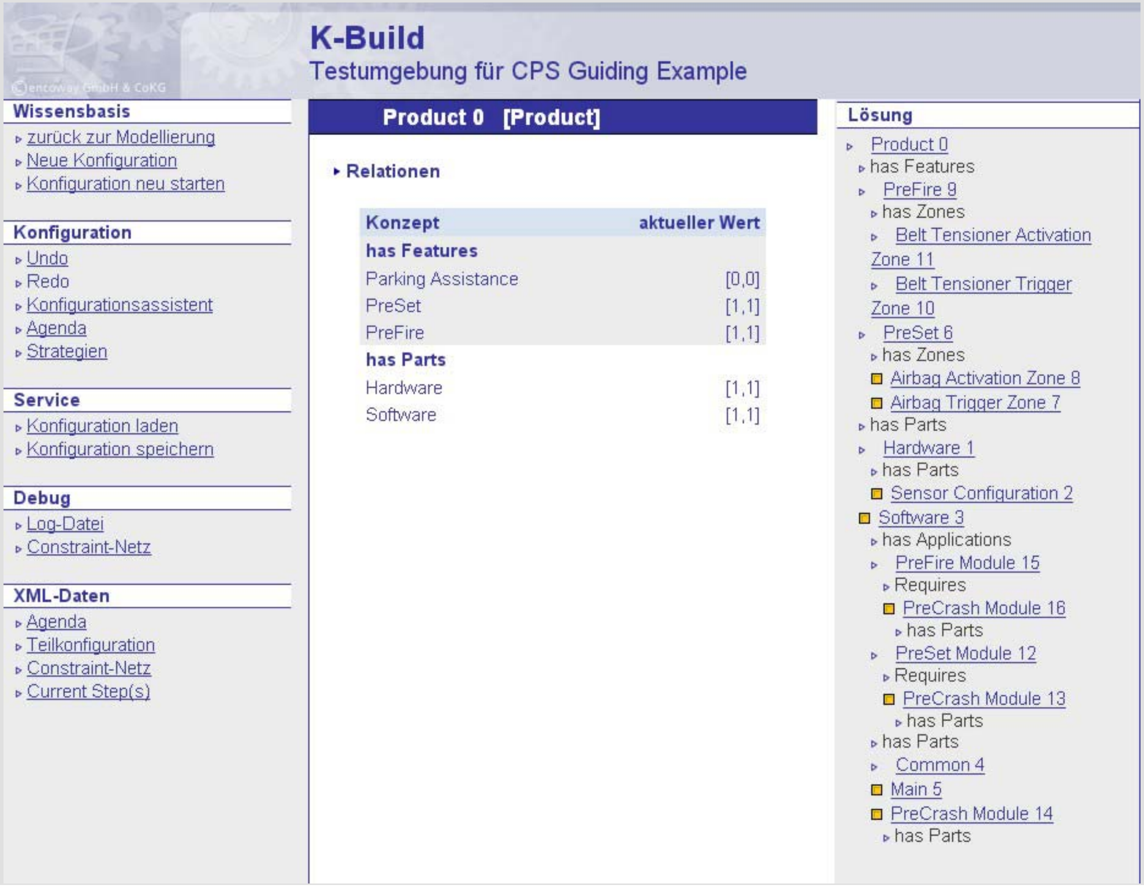
Loop-Ende: 19,1

VIDEO: ON

MMI Simulation



EngCon - Running Configuration



K-Build

Testumgebung für CPS Guiding Example

Wissensbasis

- zurück zur Modellierung
- Neue Konfiguration
- Konfiguration neu starten

Konfiguration

- Undo
- Redo
- Konfigurationsassistent
- Agenda
- Strategien

Service

- Konfiguration laden
- Konfiguration speichern

Debug

- Log-Datei
- Constraint-Netz

XML-Daten

- Agenda
- Teilkonfiguration
- Constraint-Netz
- Current Step(s)

Product 0 [Product]

▸ **Relationen**

Konzept	aktueller Wert
has Features	
Parking Assistance	[0,0]
PreSet	[1,1]
PreFire	[1,1]
has Parts	
Hardware	[1,1]
Software	[1,1]

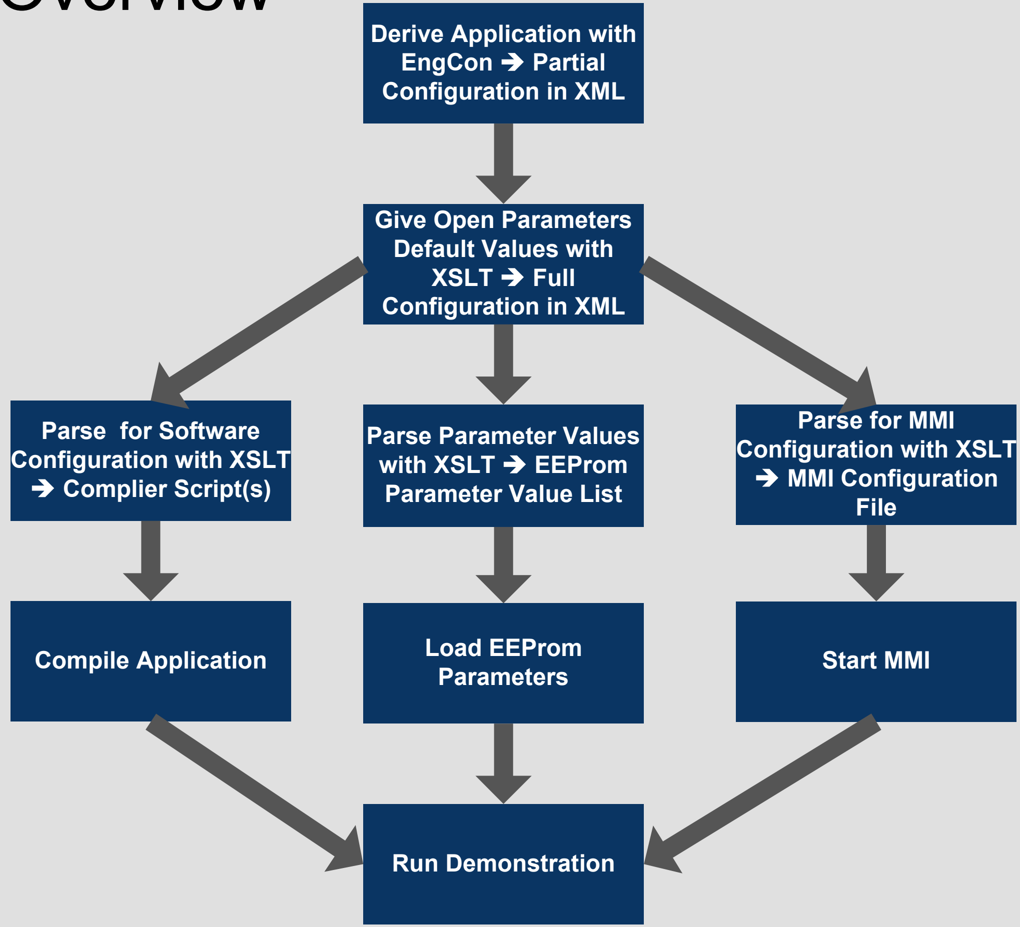
Lösung

- Product 0
  - has Features
    - PreFire 9
      - has Zones
        - Belt Tensioner Activation Zone 11
        - Belt Tensioner Trigger Zone 10
    - PreSet 6
      - has Zones
        - □ Airbag Activation Zone 8
        - □ Airbag Trigger Zone 7
  - has Parts
    - Hardware 1
      - has Parts
        - □ Sensor Configuration 2
    - □ Software 3
      - has Applications
        - PreFire Module 15
          - Requires
            - □ PreCrash Module 16
              - has Parts
        - PreSet Module 12
          - Requires
            - □ PreCrash Module 13
              - has Parts
      - has Parts
        - Common 4
        - □ Main 5
        - □ PreCrash Module 14
          - has Parts

Software Tools

- iSystem WinIdea:
  - Development IDE for the Control Unit
- Texas Instruments Compiler
  - C / C++
- Saxon
  - XSLT Processor
- XML Spy
  - General XML Editor
- LabView
  - Interface to MMI Simulation
  - Platform for Sensor Data Simulation

Demonstration Overview



Derive Application with EngCon ➔ Partial Configuration in XML

Give Open Parameters Default Values with XSLT ➔ Full Configuration in XML

Parse for Software Configuration with XSLT ➔ Complier Script(s)

Parse Parameter Values with XSLT ➔ EEProm Parameter Value List

Parse for MMI Configuration with XSLT ➔ MMI Configuration File

Compile Application

Load EEProm Parameters

Start MMI

Run Demonstration

Demonstration Notes:

- Transition from Context Configuration to Feature Configuration is fluid

- Current solution for calibration:
  - Parameter values are derived explicitly or set during initial configuration process
  - The resulting (partial) configuration is stored as the base
  - Calibration continues by further setting or testing the values as needed
  - Backtracking accomplished by going back to the base configuration and reconfiguring

- Better solutions are being investigated.

Thank-you for your Attention!