Use case analysis and functional breakdown

Docupedia Export

Author:Mukkamala Kiran (XC-AS/EDA2)
Date:29-Jan-2024 11:25

Table of Contents

- General Procedure:
  - Use cases
  - Elicit use cases (initial step)
  - Functions
  - User functions and system functions
  - Functions and atomic functions:
  - Allocating functions to logical elements
- Work in progress example for »Automated Parking Assist« (APA)
  - Use Cases
  - OneParking System Architecture

1 General Procedure:

OneParking MBSE Rollout
Proposed Requirements Structure

Stakeholder Needs

- APA base: Set of Use Cases and Scenarios
- APA enhanced: Set of Use Cases and Scenarios

Analysis (OPGI-301)

- Set of atomic User Functions

Analysis (OPGI-301)

- Mapping Table

Derived L2 Requirements (OPGI-301)

	APA base	APA enhanced	Anywhere Parking	Pull Out Control
Atomic User Function 1	x	x	x	x
Atomic User Function 2		x	x	
Atomic User Function 3			x	x

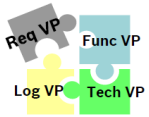
System Requirements

- SysRS "OneParking Platform"
  - Chapter "Atomic **User** Function 1"
    - Requirements related to User Function 1
  - Chapter "Atomic **User** Function 2"
    - ...
  - Chapter "Atomic **User** Function 3"
    - ...

Analysis and derived L3 Requirements (OPGI-301)

- SysRS "Subsystem X"
  - Chapter "Atomic **System** Function A"
  - Chapter "Atomic **System** Function B"

BBM MBSE Framework: Requirements Viewpoint
System Requirements - Hints



- The Stakeholder Requirements are analyzed, harmonized, refined, elaborated and translated into a set of System requirements that guides the architecture + design. This is an iterative process
- Refine the System Requirements to the extent that is necessary for the system architecture
- To analyze or derive functional requirements use scenarios, state model and models of the functional viewpoint
- To analyze or derive non functional requirements use context model and models of the logical viewpoint and technical viewpoint
- Use the model to evaluate the System Requirements technical impact and technical feasibility
- Structure the System Requirements Specification by e.g. user functions
- Prioritize system requirements and map them to a release inside the project
- Tag constraints -> important input for architecture

BOSCH

MBSE Structure Proposal

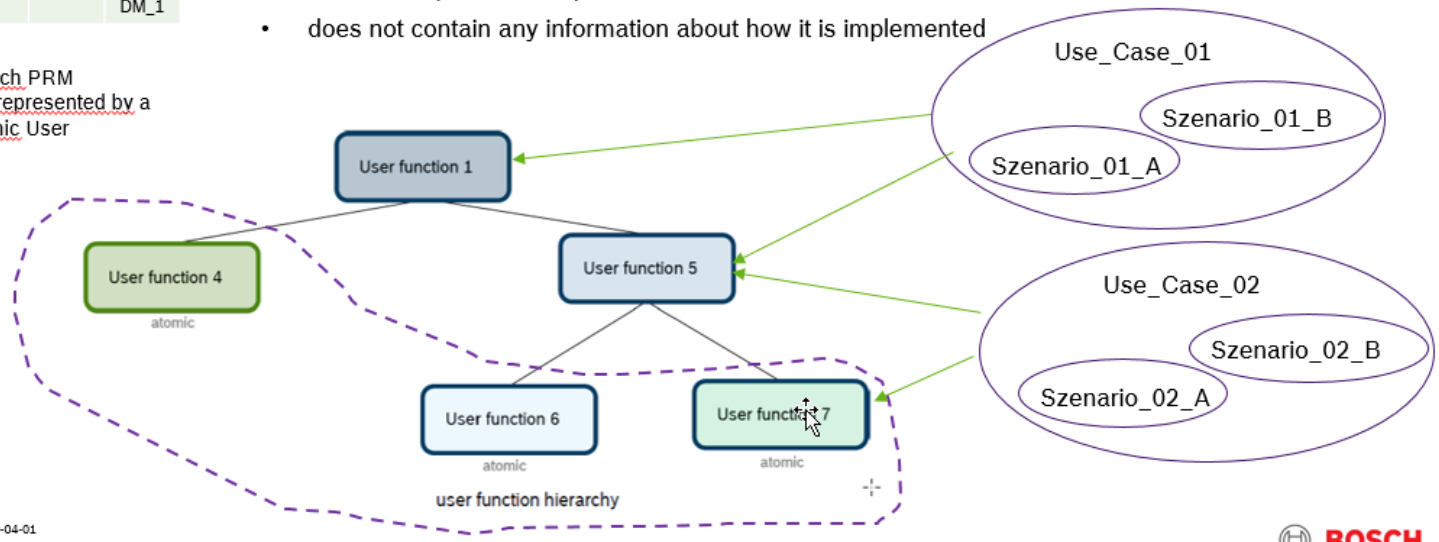
Structured by atomic User Functions

	APA	GPA	MEB	PCB	DM
aUF_A	x		x		DM_1
aUF_B		x			DM_1
aUF_C			x	x	DM_1
aUF_D	x		x		DM_1

User Functions and Atomic User Functions (aUF)

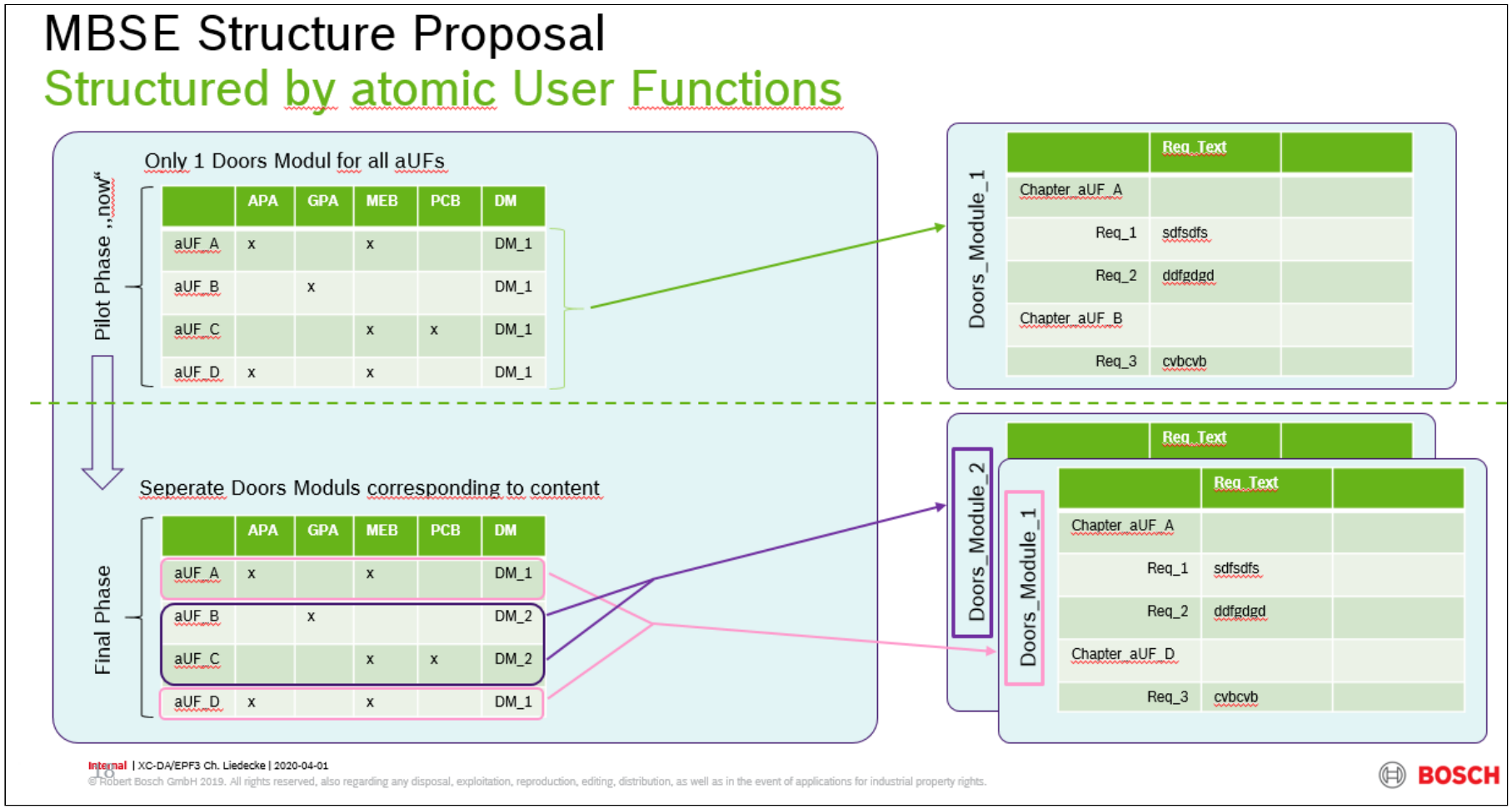
- Complete set of atomic user functions (goal is: user functions are independent from each other)
- defines a part of the system behavior observable at the system boundary
- has inputs and outputs on the system boundary (in some cases user functions can have no outputs at the system boundary, but lead to a change of a system state)
- can have inputs and outputs to other user functions, so called mode channels
- does not contain any information about how it is implemented

Mapping table: Each PRM Function shall be represented by a collections of atomic User Functions



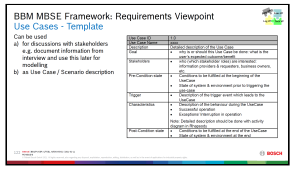
17 Internal | XC-DA/EPF3 Ch. Liedecke | 2020-04-01

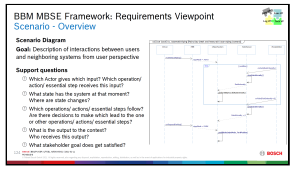
BOSCH

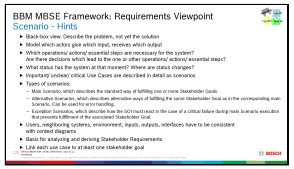


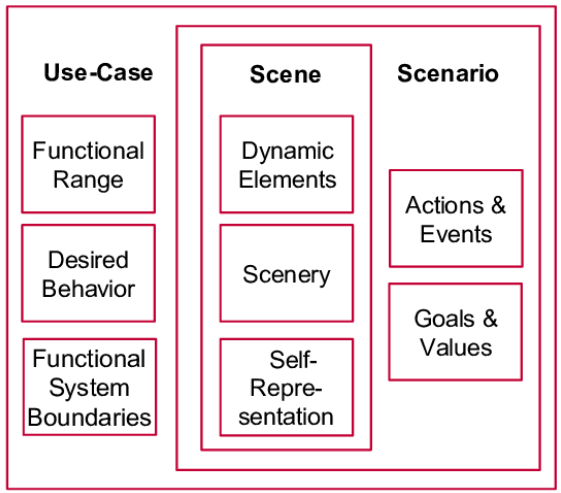
1.1 Use cases

Model Element	Intent	Procedure	Naming Convention	Necessary Description
Use Case	Describe functionality from a user's or stakeholder's perspective. Consider the **system** as a **black box**. A use case can be used as a basis for clarifying existing or generating new stakeholder requirements. It shall not contain or anticipate the system's internal design	Start with the system boundaries, the actor or context element which **triggers the use case**, and the **expected result**. Then describe the use case as abstract as possible, and as specific as necessary. Add further actors and context elements as required. Describe the use case's scenarios (main, alternative, and exception scenarios) – and in case of critical or ambiguous use cases, make your description more detailed.	**Verb** (active voice + gerund – *-ing* form) **+ object** Example: Selecting Maneuver Type	• Author and contact person • Use case name and unique identifier (tbd.) • Short description of the use case, the scene, and the relevant scenarios • System boundary • Relevant actors and stakeholders, context elements





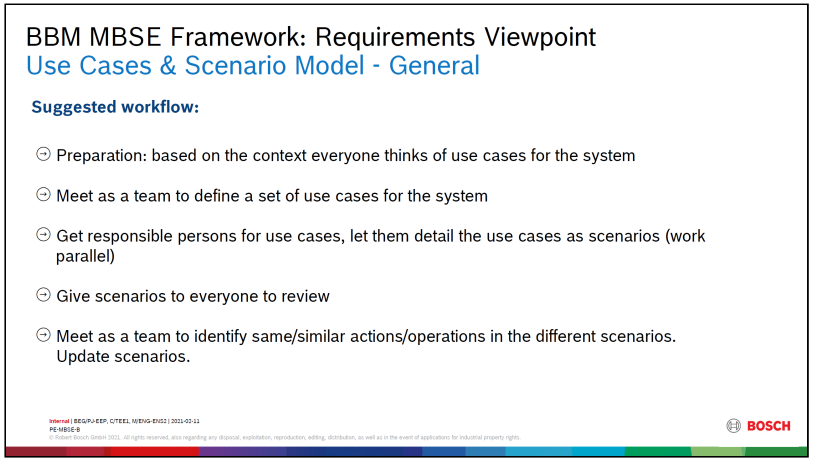


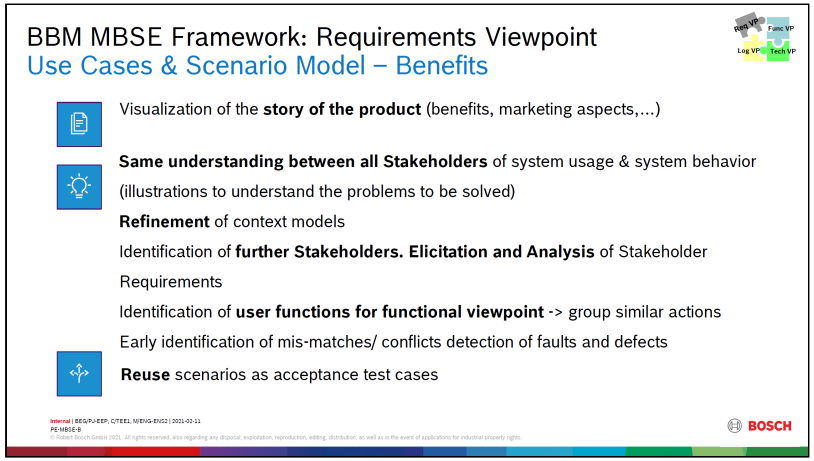


Source

Ulbrich, Simon & Menzel, Till & Reschka, Andreas & Schuldt, Fabian & Maurer, Markus. (2015). Definition der Begriffe Szene, Situation und Szenario für das automatisierte Fahren. (Download Link)

1.2 Elicit use cases (initial step)





Relevant sources

- PRM Modular Kit
- PRM Feature Lists
- USS Gen. 6 Feature List
- NRCS2 Feature List

More on Use Cases and Scenarios:

- BBM MBSE Methodology Documentation

Good Practice Example:

- Garage Parking Use Case Development and Requirements Derivation

1.3 Functions

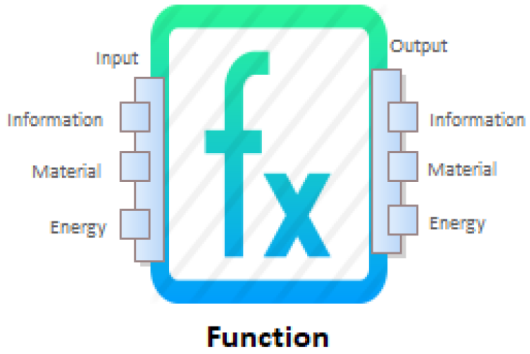
BBM MBSE Framework: Functional Viewpoint
General

▶ What is a function?

A **FUNCTION** is a *common* and *intended* relationship between input and output of a system or component that fulfills some required task.
Common means:
- the relationship can be described in abstract way

intended means:
- function **is not** a malfunction

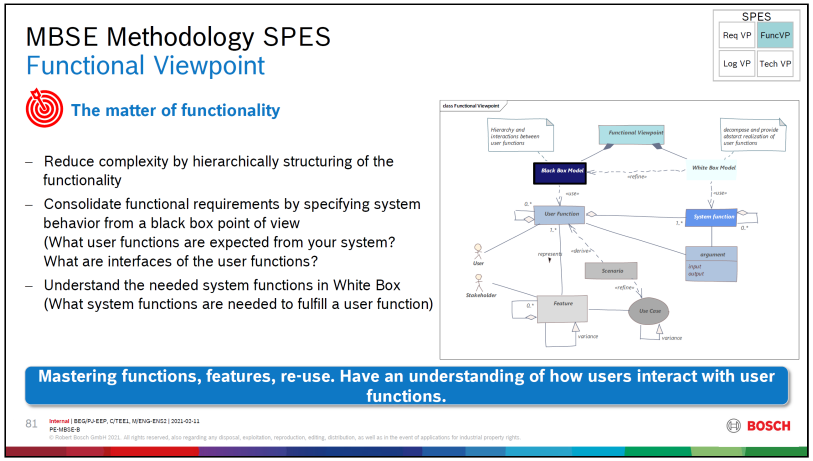
SeBOK
A function is defined by the transformation of input flows to output flows, with defined performance.

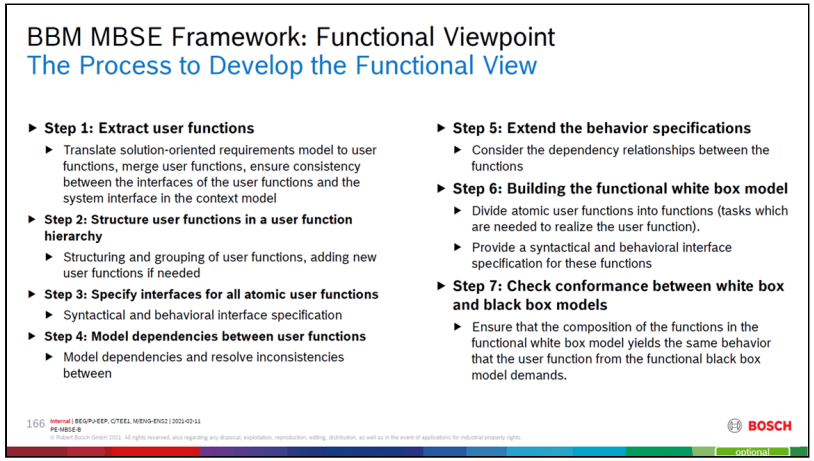


165 Internal | BEG/PJ-EEP, C/TEE1, M/ENG-ENS2 | 2021-02-11
PE-MBSE-B

BOSCH

1.4 User functions and system functions





In principle, user functions and system functions exist on every level. If a given function qualifies as a user or a system function depends on your perspective. Proposal: Instead of using the terms user function and system function it might be less confusing to consider them as *Black Box* / *White Box* functions on the level of interest.

Scope	User Function / Black Box	System Function / White Box
PRM Feature	MBSE L1 (Vehicle Level)	MBSE L2 (ADAS)
OneParking Top Level System Requirements	MBSE L2 (ADAS)	MBSE L3 (ECU/Sensor/Subsystem Level)
OneParking Subsystem Requirements	MBSE L3 (ECU/Sensor/Subsystem Level)	MBSE L4 (Engineering Domain Hardware/Software)

Further information on:

- User Functions and Functional Black Box Models– from BBM System Development Kit (Link)
- System Functions and Functional White Box Models – from BBM System Development Kit (Link)

Click here for an overview of OneParking-relevant levels

MBSE Level	Name	Objects of Interest	Engineering Category	Requirements, Requirements View	Functional View	Logical View	Technical View
0	**World**	World	not applicable	not applicable	not applicable	System Context	not applicable
1	**Vehicle**	• Ego Vehicle • Other Vehicles • Pedestrians • Roads • Parking Spaces • Obstacles • Vegetation • Static Objects • Potentially moving Objects	Vehicle E/E	Enterprise Architect (Platform Repo): • PRM Features • Use Cases • Use Case Scenarios DOORS: • ADAS / OneParking Stakeholder Requirements Modules	optional	Enterprise Architect (Platform Repo): System Context	System Context of • Reference Projects • Customer Projects

2	**Vehicle Domain**	• ADAS • Powertrain System • Braking System • Steering System • Cockpit HMI System • ...	System (ADAS, OneParking)	DOORS: • ADAS / OneParking System Level Requirements Modules	Enterprise Architect (Platform Repo): Engineering Functions and corresponding behavioral models of the ADAS, e.g. • Activities • Interactions • State Machines • ... Example ADAS Functions: • “Find usable Parking Space” • "Detect Objects" • “Control automated Parking Maneuver” • ...	Enterprise Architec (Platform Repo): Advanced Driver Assistance System (ADAS) Neighboring Systems of the ADAS	System Architecture of • Reference Projects • Customer Projects

3	**Control Unit**	OneParking Subsystems as Black Boxes • Ultrasonic Sensing • Vehicle Sensing • Radar Sensing • Near Field Model • Motion Control • Parking Space Detection • …	System (ADAS, OneParking)	DOORS: • OneParking System Requirements (Subsystem Level) • Separate DOORS modules for each subsystem • Specifications for Subsystems from external Suppliers (e.g., 3rd Party Video Perception, 3rd Party Camera Heads)	Enterprise Architect (Platform Repo): Behavioral models of Subsystems, e.g. • Activities • Interactions • State Machines • … Example Functions of Subsystems: • “Provide static Objects” • “Calculate Target Position” • “Perform Parkable Check” • “Arbitrate Parking Functions” • “Provide Deceleration Request” • …	Enterprise Architect (Platform Repo): • Ultrasonic Sensing Subsystems • Video Sensing Subsystems • Sensing Hypothesis Subsystems • Environmental Mapping Subsystems • Situation Interpretation Subsystems • Parking Function Subsystems • Motion Control Subsystem	System Archtiecture of • Reference Projects • Customer Projects

						• ...	
4	**Engineering Domain**	OneParking Subsystems as White Boxes Engineering Domains Hardware/Software	Hardware Software System (partly)	Sphinx Needs: • Requirements for Logical L4 Blocks	Enterprise Architect (Platform Repo): Behavioral models of Subsystems' L4 Logical Blocks Example Functions: • "Redefine Depth Reference" • "Perform Target-based Camera Calibration" • ...	Enterprise Architect (Platform Repo): Examples of Logical L4 Blocks: • "Video Optical Flow" • "Ultrasonic Echo Tracing" • "Parking Space Merger" • "Localization Manager" • ...	Software Architecture in Sphinx Needs Hardware Architecture in Enterprise Architect
5	**Component**	Electronics, Mechanics (examples tbd.), Software	Hardware Software	(out of MBSE scope)	(out of MBSE scope)	(out of MBSE scope)	(out of MBSE scope)

6	**Module**	Hardware Parts: • System on Chip (SoC) • Controller Chips • System ASICs Software Modules: • Source and Header Files	Hardware Software	(out of MBSE scope)	(out of MBSE scope)	(out of MBSE scope)	(out of MBSE scope)
7	**Element**	Hardware: • Interfaces (e.g., SPI, UART) • “Sub-Components” of Hardware Parts (e.g., RAM, Firmware, Package) Software: • Units	Hardware Software	(out of MBSE scope)	(out of MBSE scope)	(out of MBSE scope)	(out of MBSE scope)

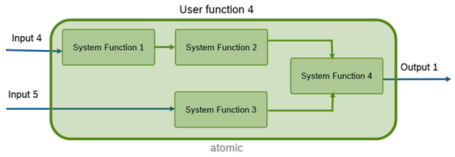
Info

This chapter contains a reusable documentation segment (PageWithExcerpt = Use case analysis and functional breakdown | MultiExcerptName = OneParking_MBSE_Levels)

BBM MBSE Framework: Functional Viewpoint
Functional White Box Model - Overview

Goal:

Managing complexity by hierarchically structuring the functionality of a system (black box to white box perspective). Decomposition of atomic user functions into a set of system functions



system function =

- defines a part of the system behavior not observable at the system boundary
- has inputs and/or outputs, that are not on the system boundary
- is a usage of a user function of the lower system level

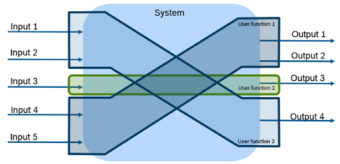
175 Internal | BEG/PJ-EEP, CITEEL MENG-EN02 | 2021-02-11
PE-MBSE-B

BOSCH

BBM MBSE Framework: Functional Viewpoint
Functional Black Box Model - Overview

Goal:

- Specifying the system behavior from a black box perspective
- Structures **all user functions** of a system and describes their **dependencies** with each other



user function =

- defines a part of the system behavior observable at the system boundary
- has inputs and outputs on the system boundary (in some cases user functions can have no outputs at the system boundary, but lead to a change of a system state)
- can have inputs and outputs to other user functions, so called mode channels
- does not contain any information about how it is implemented

168 Internal | BEG/PJ-EEP, CITEEL MENG-EN02 | 2021-02-11
PE-MBSE-B

BOSCH

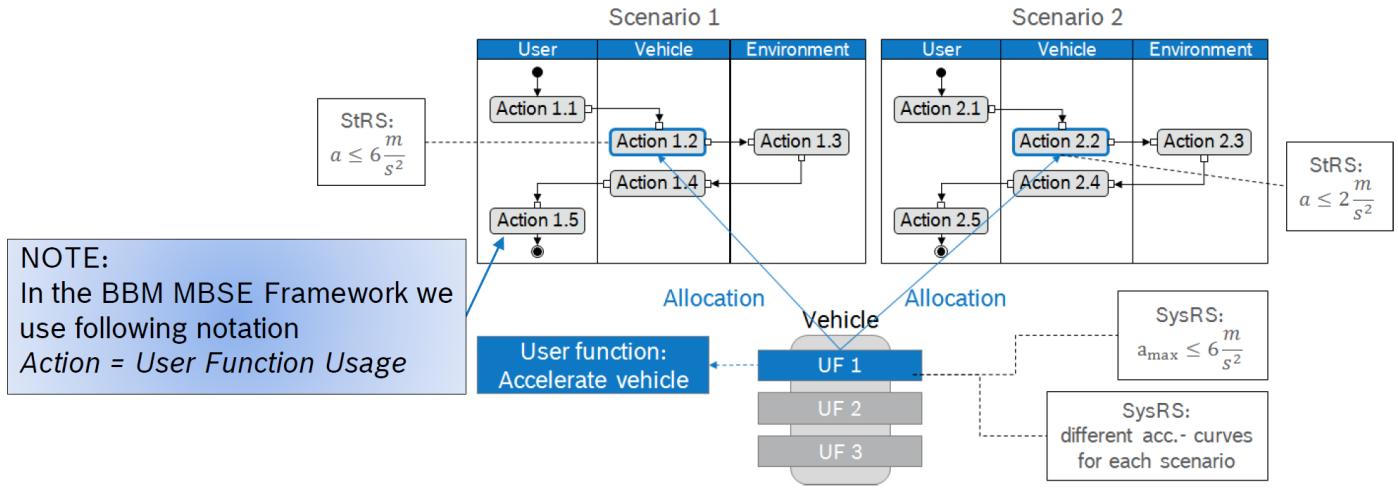
1.5 Functions and atomic functions:

BBM MBSE Framework: Functional Viewpoint

Functional Black Box Model - Hints

How to derive user functions:

User functions depict the consolidated version of system actions used in the description of use case scenarios that have the same target condition. For example, in scenario 1 and scenario 2 there are system actions which describe the acceleration of the vehicle. For the vehicle a user function "Accelerate vehicle" consolidates these system actions.



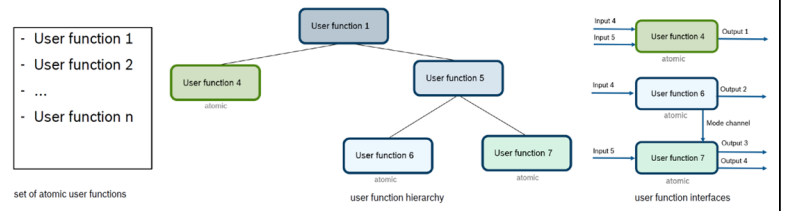
171 Internal | BEG/PJ-EEP, C/TEE1, M/ENG-ENS2 | 2021-02-11
PE-MBSE-B

BOSCH

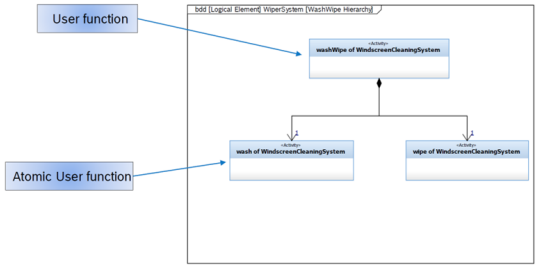
BBM MBSE Framework: Functional Viewpoint
Functional Black Box Model - Hints

Hints: A functional black-box model shows

- Complete set of atomic user functions (goal is: user functions are independent from each other)
- User function hierarchy (optional, if needed)
- User function interfaces



BBM MBSE Framework: Functional Viewpoint
Functional Black Box model - SysML

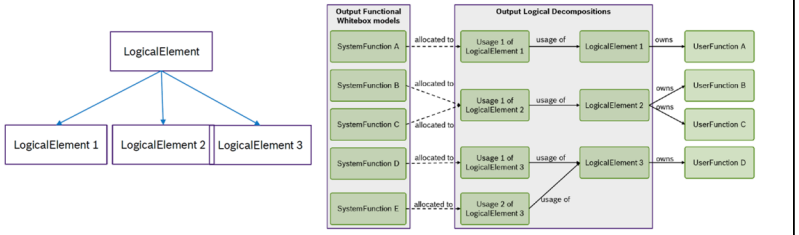


1.6 Allocating functions to logical elements

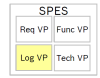
BBM MBSE Framework: Logical Viewpoint
Connection Functional to LogicalViewpoint

White Box & Logical Decomposition

- Logical decomposition of system into subsystems
- Find the right granularity level for system functions, so that they can be allocated to subsystems

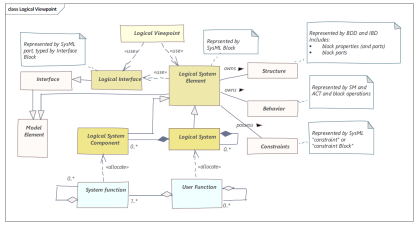


MBSE Methodology SPES
Logical Viewpoint



Master complexity

- Describe the internal logic of the system
  - divide the system into communicating logical components
- Allocate desired functions to logical components
- Support the reuse of existing logical components
- Facilitate future reuse of logical components
- Define the total behavior of the system
  - as opposed to the partial behavior specifications in the models of the functional viewpoint
- Enable the complete simulation of the entire system



System decomposition with total system behavior to specify system interfaces, enable reuse and early verification

2 Work in progress example for »Automated Parking Assist« (APA)

Goals of the related OnePlanning Ticket:

OPGI-301 - Establish functional breakdown for each OneParking system function CLOSED

1. All OneParking-relevant functions and use cases on MBSE Level 1 (vehicle), Level 2 (ADAS), and Level 3 (subsystems) are:
    - elicited and analyzed (sources: PRM Modular Kit, state of the art in USS Gen 6 and NRCS2, etc.)
    - applicable constraints and performance criteria have been derived
    - broken down to match the granularity necessary for allocating them to one single OneParking subsystem
    - System requirements towards affected OneParking subsystems have been derived
2. Initial baselines for all OneParking-relevant functions are available and reviewed
    - System requirements for OneParking Functions
    - System architecture elements (use cases, activities, high level functional interfaces)
3. Created artifacts are enablers for:
    - establishing function-specific cause/effect chains
    - describing the function-specific dynamic behavior
4. BBM MBSE principles are considered during creation of system requirements and system architecture elements

For the collection of content, this Excel Template is being used.

2.1 Use Cases

Sources of Use Cases

- PRM Modular Kit
- NRCS Function List
- USS Gen. 6 Feature List

Collect use cases which are relevant for a given "Feature", e.g.:

- APA Standard Use Cases
- APA Extended/Special/Corner Use Cases)

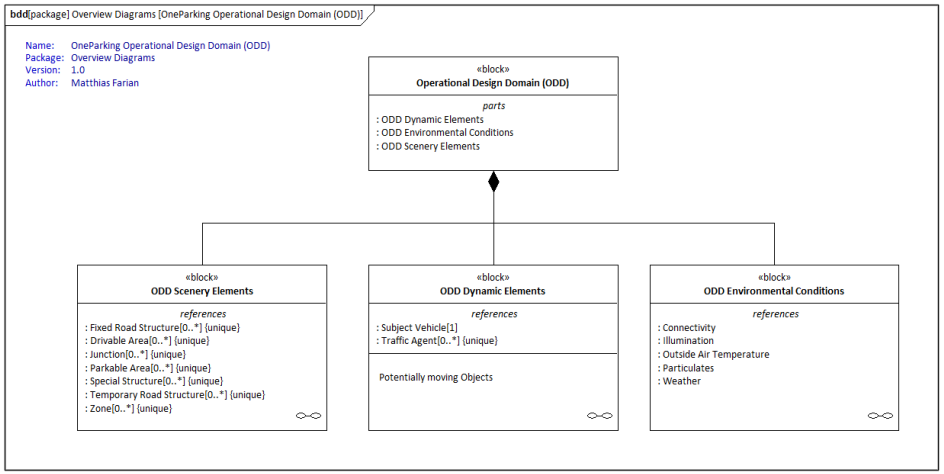
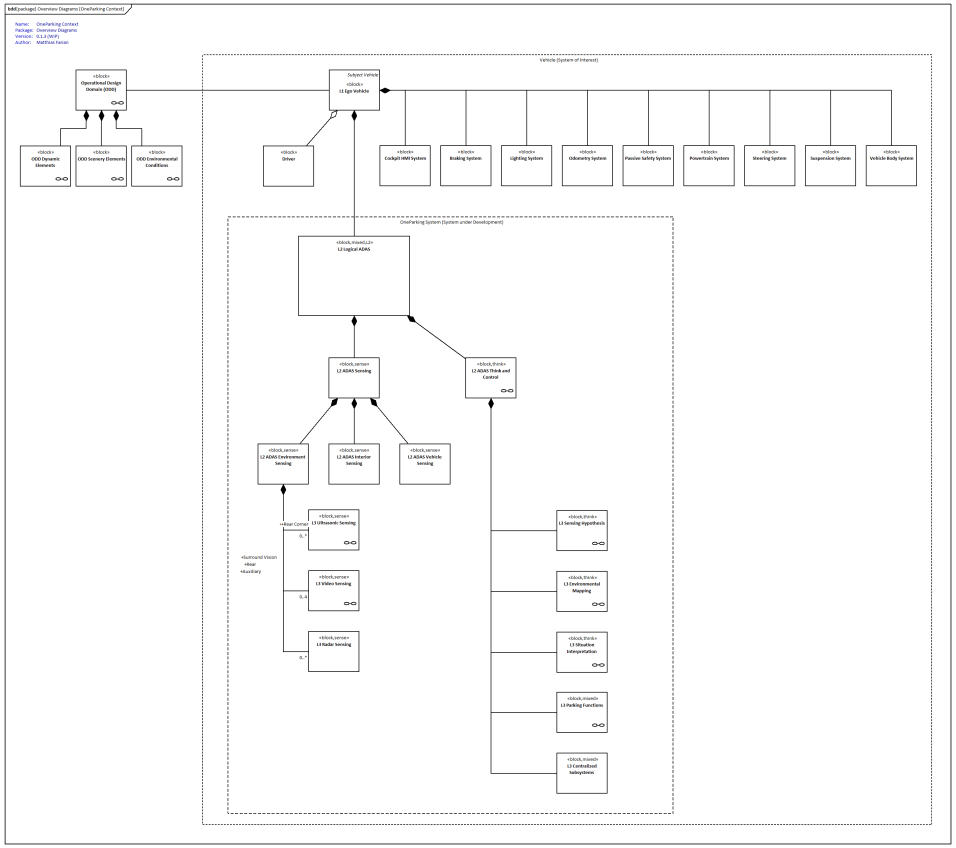
More on Use Cases and Scenarios:

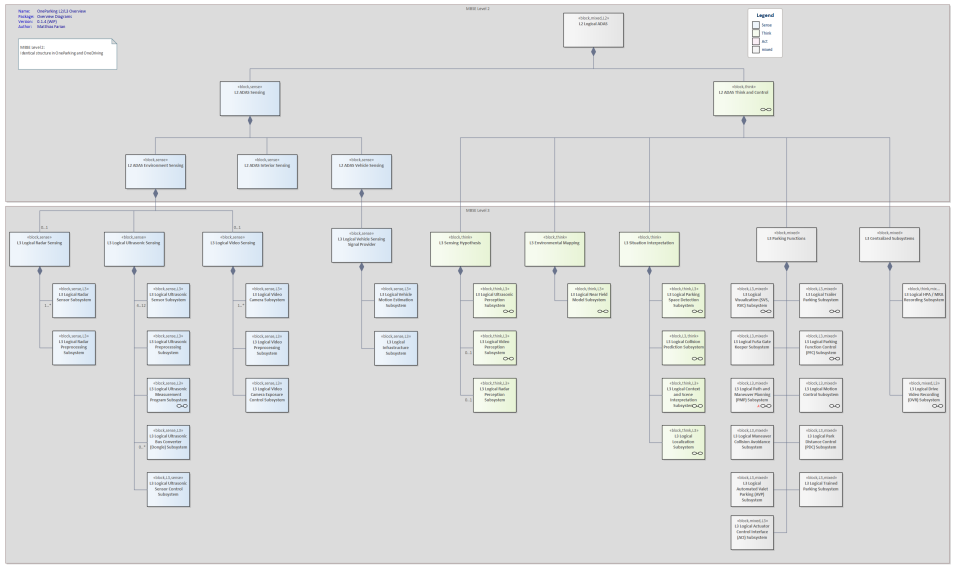
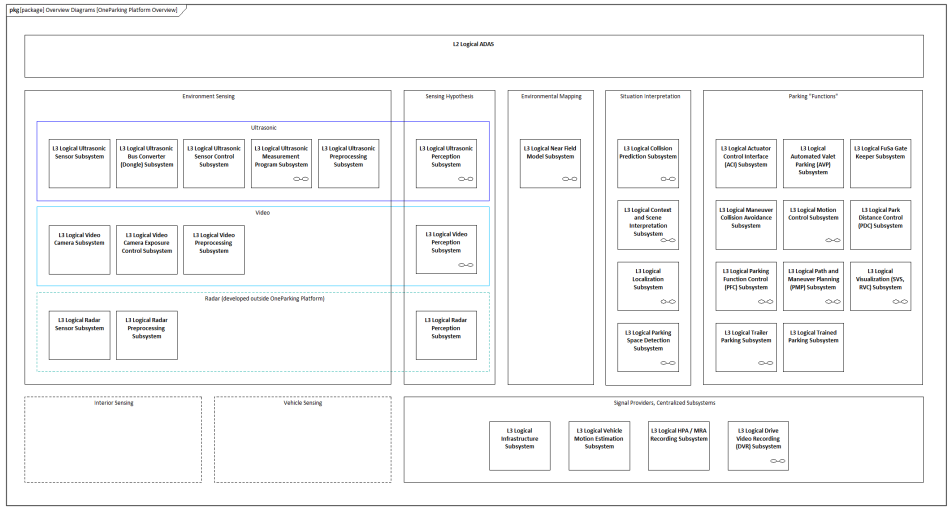
- BBM MBSE Methodology Documentation

Good Practice Example:

- Garage Parking Use Case Development and Requirements Derivation

2.2 OneParking System Architecture

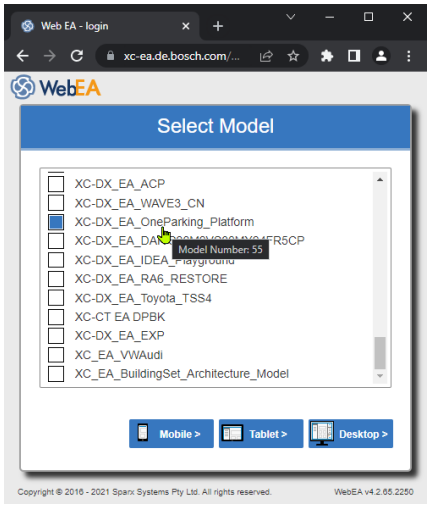
OneParking Operational Design Domain (ODD)	OneParking Context Model
	
Last updated: 2023-05-23. Unsynchronized screenshot of diagram, current version is only available via WebEA, see link below.	Last updated: 2022-09-29. Unsynchronized screenshot of diagram, current version is only available via WebEA, see link below,
Original diagram: OneParking Operational Design Domain (WebEA)	Original diagram: OneParking Context (WebEA)

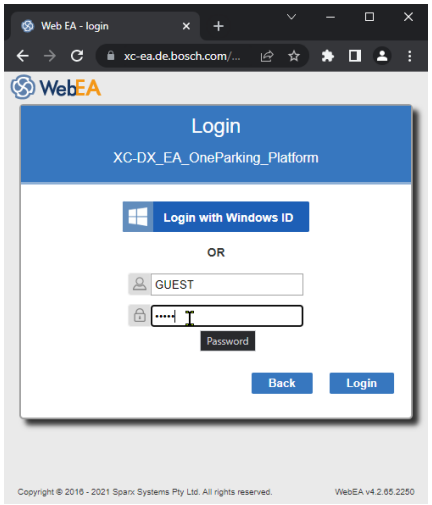
OneParking: Logical Architecture, MBSE Levels 2 and 3	OneParking Platform Overview
	
Last updated: 2023-05-23. Unsynchronized screenshot of diagram, current version is only available via WebEA, see link below,	Last updated: 2023-05-23. Unsynchronized screenshot of diagram, current version is only available via WebEA, see link below,
Original diagram: OneParking L2/L3 Overview (WebEA)	Original diagram: OneParking Platform Overview (WebEA)

Info

This chapter contains a reusable documentation segment (PageWithExcerpt = Use case analysis and functional breakdown | MultiExcerptName = OverviewDiagrams_Logical)

Enterprise Architect Model for the OneParking Platform, accessible via WebEA





- URL: https://xc-ea.de.bosch.com/WebEA/login.php
- Model Number: 55
- Model Name: XC-DX_EA_OneParking_Platform

Read-only access is available for everyone

- User: GUEST
- Password: GUEST

Registered users can also use "Login with Windows ID"

Info

This chapter contains a reusable documentation segment (PageWithExcerpt = Use case analysis and functional breakdown | MultiExcerptName = WebEA_Access)