

Use case analysis and functional breakdown

Docupedia Export

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

Table of Contents

| | |
|--|-----------|
| 1 General Procedure: | 4 |
| 1.1 Use cases | 8 |
| 1.2 Elicit use cases (initial step) | 10 |
| 1.3 Functions | 11 |
| 1.4 User functions and system functions | 12 |
| 1.5 Functions and atomic functions: | 19 |
| 1.6 Allocating functions to logical elements | 20 |
| 2 Work in progress example for »Automated Parking Assist« (APA) | 21 |
| 2.1 Use Cases | 21 |
| 2.2 OneParking System Architecture | 22 |

- 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

| | 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 |

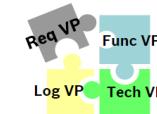
System Requirements

- ▶ SysRS “OneParking Platform”
 - ▶ Chapter “Atomic User Function 1”
 - Requirements related to User Function 1
 - ▶ Chapter “Atomic User Function 2”
 - ...
 - ▶ Chapter “Atomic User Function 3”
 - ...
- ▶ SysRS “Subsystem X”
 - ▶ Chapter “Atomic System Function A”
 - ▶ Chapter “Atomic System Function B”

Analysis and derived L3 Requirements (OPGI-301)

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

MBSE Structure Proposal

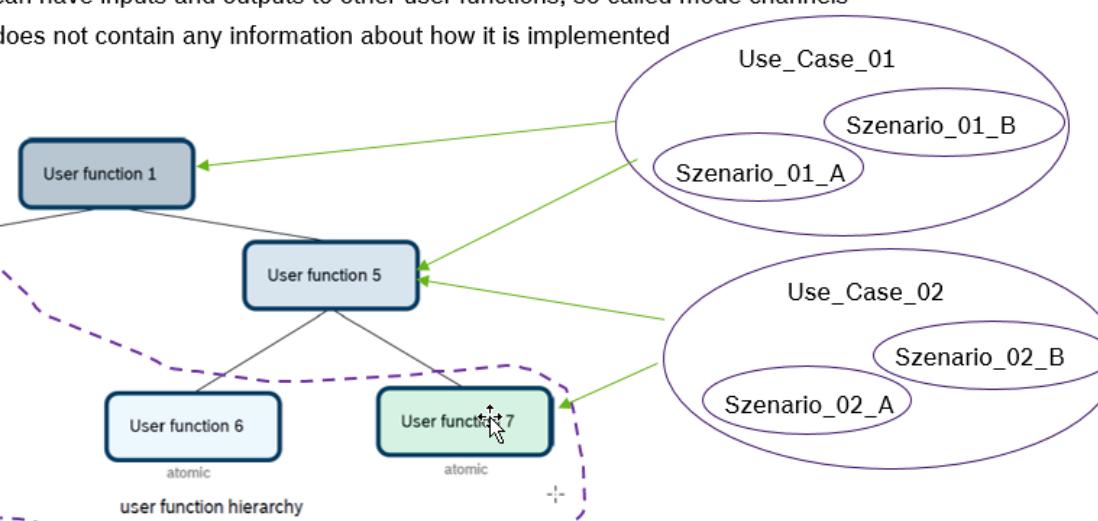
Structured by atomic User Functions

| | 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 |

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

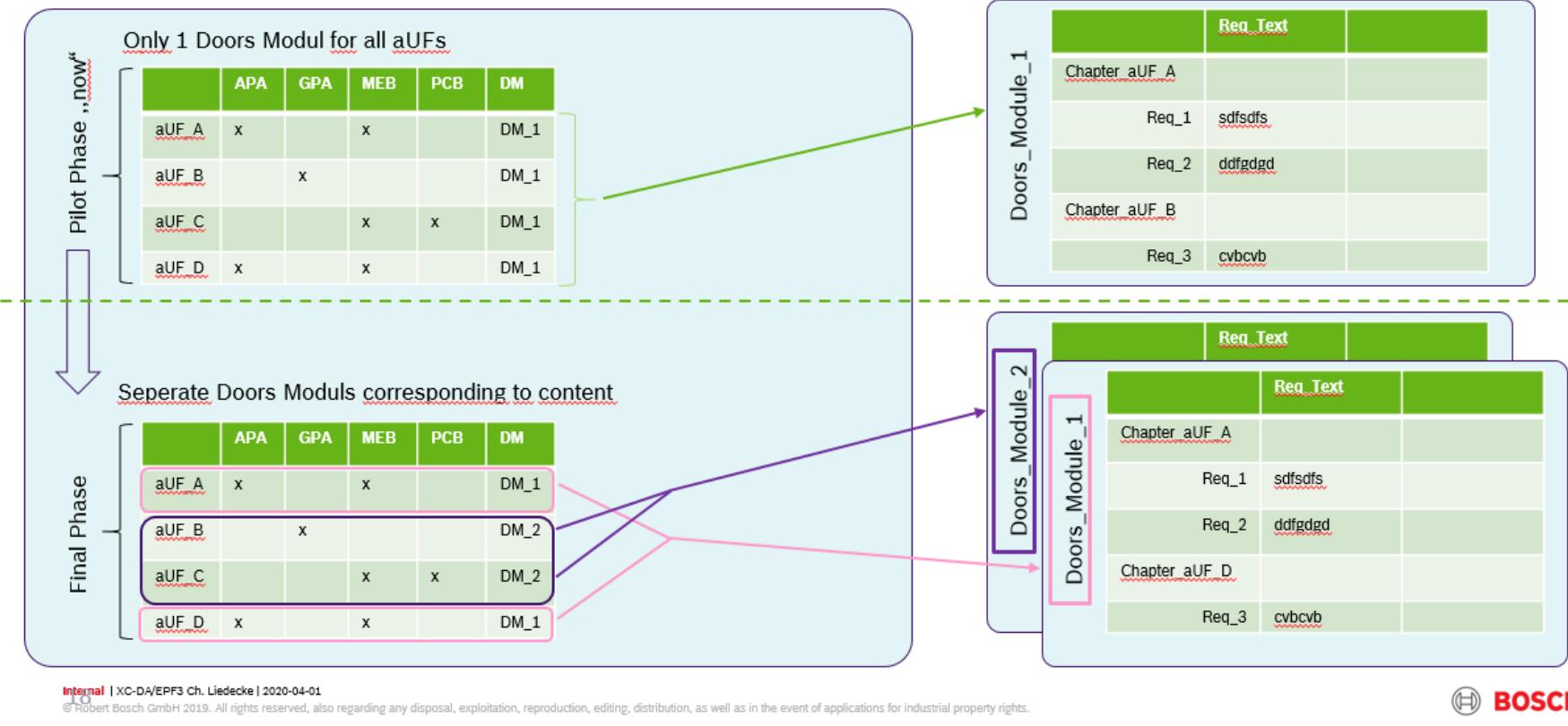
User Functions and Atomic User Functions (aUF)

- 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



MBSE Structure Proposal

Structured by atomic User Functions



1.1 Use cases

| Model Element | Intent | Procedure | Naming Convention | Necessary Description |
|---------------|---|--|---|--|
| Use Case | <p>Describe functionality from a user's or stakeholder's perspective. Consider the system as a black box.</p> <p>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</p> | <p>Start with the system boundaries, the actor or context element which triggers the use case, and the expected result.</p> <p>Then describe the use case as abstract as possible, and as specific as necessary. Add further actors and context elements as required.</p> <p>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.</p> | <p>Verb (active voice + gerund – <i>-ing</i> form) + object</p> <p>Example: Selecting Maneuver Type</p> | <ul style="list-style-type: none"> • 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 |

BBM MBSE Framework: Requirements Viewpoint
Use Cases - Template

Can be used:
a) for discussions with stakeholders
e.g. document information from stakeholder requirements modeling
b) as Use Case / Scenario description

BBM MBSE Framework: Requirements Viewpoint
Scenario - Overview

Scenario Diagram
A graphical representation of interactions between users and neighboring systems from user perspective

BBM MBSE Framework: Requirements Viewpoint
Scenario - Hints

Support questions
What does the system do?
Who are the actors?
What does each actor do?
What does the system receive?
What does the system give?
What does the system change?
What does the system know?
What does the system need?
What does the system have?
What does the system do with what it has?
What does the system give to whom?
What does the system receive from whom?
What does the system do with what it receives?
What does the system do with what it gives?
What does the system do with what it has and what it receives?

| | | |
|---|--|---|
| <p>Use-Case</p> <ul style="list-style-type: none"> Functional Range Desired Behavior Functional System Boundaries | <p>Scene</p> <ul style="list-style-type: none"> Dynamic Elements Scenery Self-Representation | <p>Scenario</p> <ul style="list-style-type: none"> Actions & Events Goals & Values |
|---|--|---|

Source

Ulrich, 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)

BBM MBSE Framework: Requirements Viewpoint
Use Cases & Scenario Model - General

Suggested workflow:

- ⌚ Preparation: based on the context everyone thinks of use cases for the system
- ⌚ Meet as a team to define a set of use cases for the system
- ⌚ Get responsible persons for use cases, let them detail the use cases as scenarios (work parallel)
- ⌚ Give scenarios to everyone to review
- ⌚ Meet as a team to identify same/similar actions/operations in the different scenarios. Update scenarios.

Internal | BSG/P-IEP-CITEEL-NVHQ-ENG1 | 2021-03-11
PE-MBSE-B
© Robert Bosch GmbH 2021. All rights reserved, also regarding any disposal, exploration, reproduction, editing, distribution, as well as in the event of applications for industrial property rights.



BBM MBSE Framework: Requirements Viewpoint
Use Cases & Scenario Model – Benefits



- Visualization of the **story of the product** (benefits, marketing aspects,...)
- Same understanding between all Stakeholders** of system usage & system behavior (illustrations to understand the problems to be solved)
- Refinement** of context models
- Identification of **further Stakeholders**. **Elicitation and Analysis** of Stakeholder Requirements
- Identification of **user functions for functional viewpoint** -> group similar actions
- Early identification of mis-matches/ conflicts detection of faults and defects
- Reuse** scenarios as acceptance test cases

Internal | BSG/P-IEP-CITEEL-NVHQ-ENG1 | 2021-03-11
PE-MBSE-B
© Robert Bosch GmbH 2021. All rights reserved, also regarding any disposal, exploration, reproduction, editing, distribution, as well as in the event of applications for industrial property rights.



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:

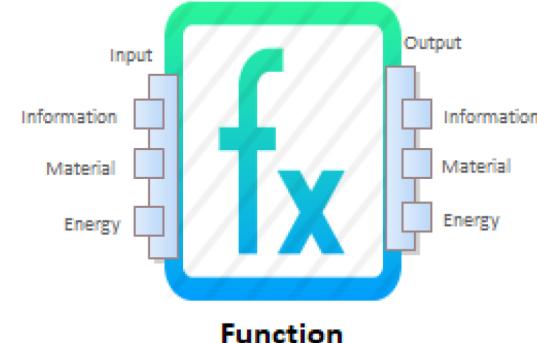
- the relationship can be described in abstract way

intended means:

- function is not a malfunction

SeBOK

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



1.4 User functions and system functions

MBSE Methodology SPES Functional Viewpoint

The matter of functionality

- Reduce complexity by hierarchically structuring of the functionality
- Consolidate functional requirements by specifying system behavior from a black box point of view
(What user functions are expected from your system?
What are interfaces of the user functions?)
- Understand the needed system functions in White Box
(What system functions are needed to fulfill a user function)

Mastering functions, features, re-use. Have an understanding of how users interact with user functions.

01 Internal | DEEP-DEP-CTEEL-NWBG-ENG2 | 2021-02-11
PE-MBSE-B
© Robert Bosch GmbH 2021. All rights reserved, also regarding any disposal, exploitation, reproduction, editing, distribution, as well as in the event of applications for industrial property rights.



BOSCH

BBM MBSE Framework: Functional Viewpoint The Process to Develop the Functional View

► Step 1: Extract user functions

- Translate solution-oriented requirements model to user functions, merge user functions, ensure consistency between the interfaces of the user functions and the system interface in the context model

► Step 2: Structure user functions in a user function hierarchy

- Structuring and grouping of user functions, adding new user functions if needed

► Step 3: Specify interfaces for all atomic user functions

- Syntactical and behavioral interface specification

► Step 4: Model dependencies between user functions

- Model dependencies and resolve inconsistencies between

► Step 5: Extend the behavior specifications

- Consider the dependency relationships between the functions

► Step 6: Building the functional white box model

- Divide atomic user functions into functions (tasks which are needed to realize the user function).
- Provide a syntactical and behavioral interface specification for these functions

► Step 7: Check conformance between white box and black box models

- Ensure that the composition of the functions in the functional white box model yields the same behavior that the user function from the functional black box model demands.

166 Internal | DEEP-DEP-CTEEL-NWBG-ENG2 | 2021-02-12
PE-MBSE-B
© Robert Bosch GmbH 2021. All rights reserved, also regarding any disposal, exploitation, reproduction, editing, distribution, as well as in the event of applications for industrial property rights.

BOSCH

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 | <ul style="list-style-type: none"> • Ego Vehicle • Other Vehicles • Pedestrians • Roads • Parking Spaces • Obstacles • Vegetation • Static Objects • Potentially moving Objects | Vehicle E/E | Enterprise Architect (Platform Repo): <ul style="list-style-type: none"> • PRM Features • Use Cases • Use Case Scenarios DOORS: <ul style="list-style-type: none"> • ADAS / OneParking Stakeholder Requirements Modules | optional | Enterprise Architect (Platform Repo): System Context | System Context of <ul style="list-style-type: none"> • Reference Projects • Customer Projects |

| | | | | | | | |
|---|-----------------------|---|---------------------------------|---|---|--|---|
| 2 | Vehicle Domain | <ul style="list-style-type: none"> • ADAS • Powertrain System • Braking System • Steering System • Cockpit HMI System • ... | System (ADAS, OneParking) | <p>DOORS:</p> <ul style="list-style-type: none"> • ADAS / OneParking System Level Requirements Modules | <p>Enterprise Architect (Platform Repo):</p> <p>Engineering Functions and corresponding behavioral models of the ADAS, e.g.</p> <ul style="list-style-type: none"> • Activities • Interactions • State Machines • ... <p>Example ADAS Functions:</p> <ul style="list-style-type: none"> • "Find usable Parking Space" • "Detect Objects" • "Control automated Parking Maneuver" • ... | <p>Enterprise Architec (Platform Repo):</p> <p>Advanced Driver Assistance System (ADAS)</p> <p>Neighboring Systems of the ADAS</p> | <p>System Architecture of</p> <ul style="list-style-type: none"> • Reference Projects • Customer Projects |
|---|-----------------------|---|---------------------------------|---|---|--|---|

| | | | | | | | |
|---|---------------------|--|----------------------------------|--|--|--|--|
| 3 | Control Unit | <p>OneParking Subsystems as Black Boxes</p> <ul style="list-style-type: none"> • Ultrasonic Sensing • Vehicle Sensing • Radar Sensing • Near Field Model • Motion Control • Parking Space Detection • ... | <p>System (ADAS, OneParking)</p> | <p>DOORS:</p> <ul style="list-style-type: none"> • 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) | <p>Enterprise Architect (Platform Repo): Behavioral models of Subsystems, e.g.</p> <ul style="list-style-type: none"> • Activities • Interactions • State Machines • ... <p>Example Functions of Subsystems:</p> <ul style="list-style-type: none"> • “Provide static Objects” • “Calculate Target Position” • “Perform Parkable Check” • “Arbitrate Parking Functions” • “Provide Deceleration Request” • ... | <p>Enterprise Architect (Platform Repo):</p> <ul style="list-style-type: none"> • Ultrasonic Sensing Subsystems • Video Sensing Subsystems • Sensing Hypotheses Subsystems • Environmental Mapping Subsystems • Situation Interpretation Subsystems • Parking Function Subsystems • Motion Control Subsystem | <p>System Archtiecture of</p> <ul style="list-style-type: none"> • Reference Projects • Customer Projects |
|---|---------------------|--|----------------------------------|--|--|--|--|

| | | | | | | | |
|----------|---------------------------|---|---|---|---|---|--|
| | | | | | | • ... | |
| 4 | Engineering Domain | OneParking Subsystems as White Boxes Engineering Domains Hardware/Software | Hardware Software System (partly) | Sphinx Needs: <ul style="list-style-type: none">• Requirements for Logical L4 Blocks | Enterprise Architect (Platform Repo): Behavioral models of Subsystems' L4 Logical Blocks Example Functions: <ul style="list-style-type: none">• "Redefine Depth Reference"• "Perform Target-based Camera Calibration"• ... | Enterprise Architect (Platform Repo): Examples of Logical L4 Blocks: <ul style="list-style-type: none">• "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 | <p>Hardware Parts:</p> <ul style="list-style-type: none"> • System on Chip (SoC) • Controller Chips • System ASICs <p>Software Modules:</p> <ul style="list-style-type: none"> • Source and Header Files | Hardware Software | (out of MBSE scope) |
| 7 | Element | <p>Hardware:</p> <ul style="list-style-type: none"> • Interfaces (e.g., SPI, UART) • “Sub-Components” of Hardware Parts (e.g., RAM, Firmware, Package) <p>Software:</p> <ul style="list-style-type: none"> • Units | Hardware Software | (out of MBSE scope) |

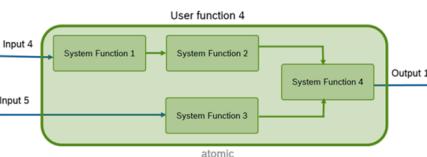
Info

i 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 | DESP/EEP_CTEEL_NEN0-EN02 | 2021-02-11
PE-MBSE-B



© Robert Bosch GmbH 2021. All rights reserved, also regarding any disposal, exploration, reproduction, editing, distribution, as well as in the event of applications for industrial property rights.

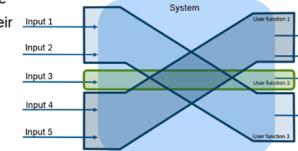
BBM MBSE Framework: Functional Viewpoint Functional Black Box Model - Overview

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 | DESP/EEP_CTEEL_NEN0-EN02 | 2021-02-11
PE-MBSE-B



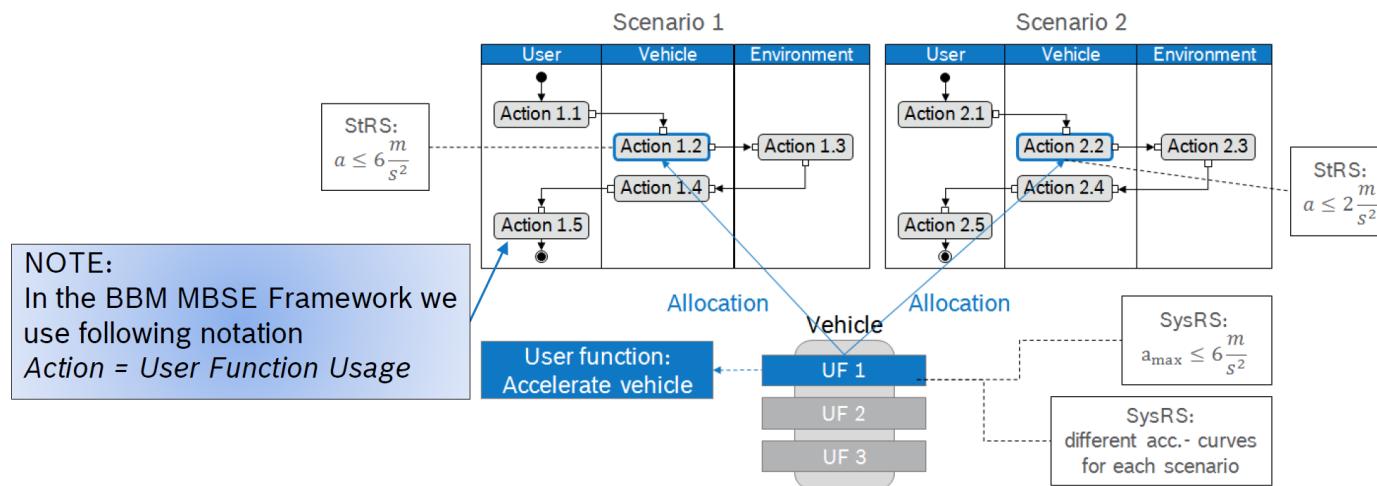
© Robert Bosch GmbH 2021. All rights reserved, also regarding any disposal, exploration, reproduction, editing, distribution, as well as in the event of applications for industrial property rights.

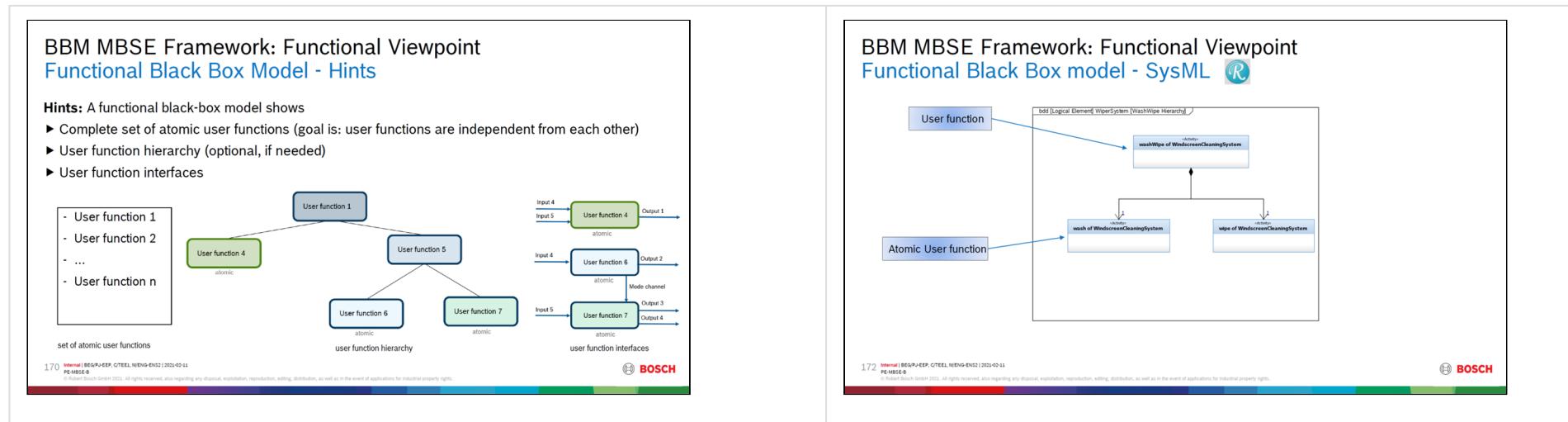
1.5 Functions and atomic functions:

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

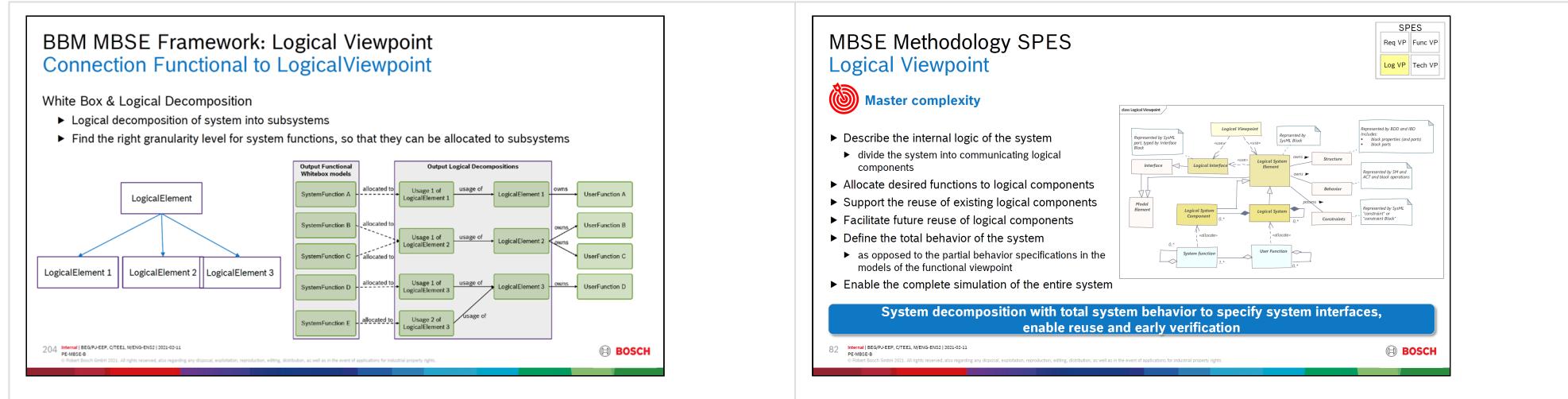
How to derive user functions:

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.





1.6 Allocating functions to logical elements



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:

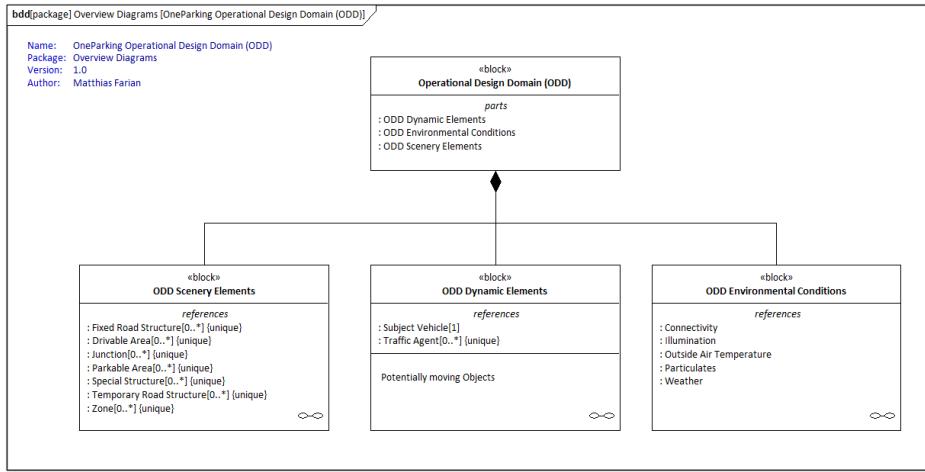
- [BBM MBSE Methodology Documentation](#)

Good Practice Example:

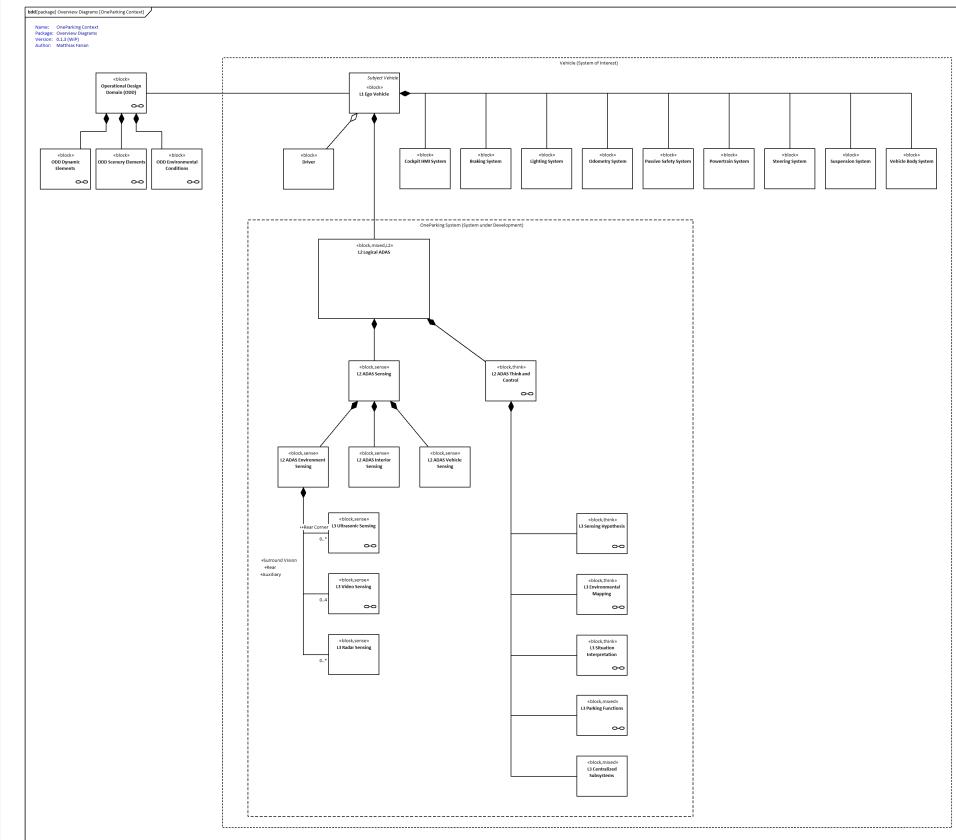
- [Garage Parking Use Case Development and Requirements Derivation](#)
-

2.2 OneParking System Architecture

OneParking Operational Design Domain (ODD)



OneParking Context Model



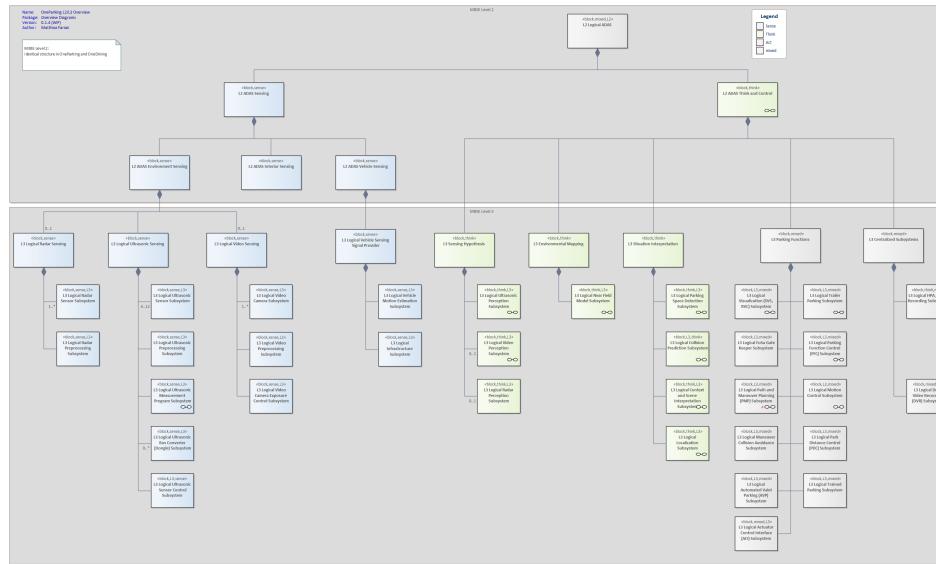
Last updated: 2023-05-23.

Unsynchronized screenshot of diagram, current version is only available via WebEA,
 see link below.

Original diagram: [OneParking Operational Design Domain \(WebEA\)](#)

Original diagram: [OneParking Context \(WebEA\)](#)

OneParking: Logical Architecture, MBSE Levels 2 and 3



Last updated: 2023-05-23.

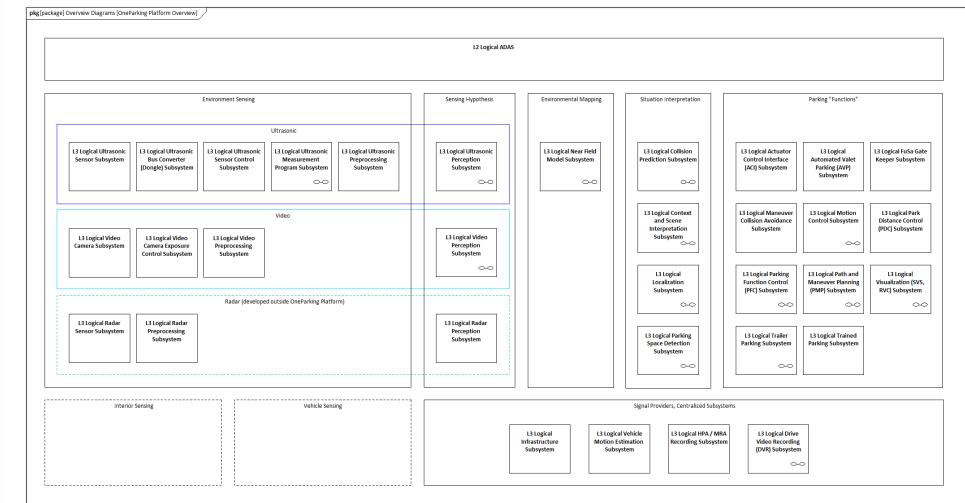
Unsynchronized screenshot of diagram, current version is only available via WebEA,
see link below,

Original diagram: [OneParking L2/L3 Overview \(WebEA\)](#)

Info

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

OneParking Platform Overview

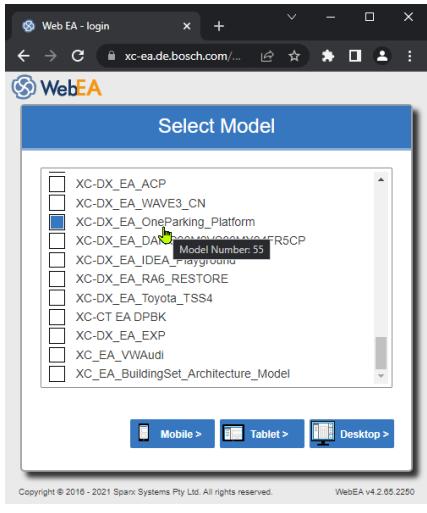
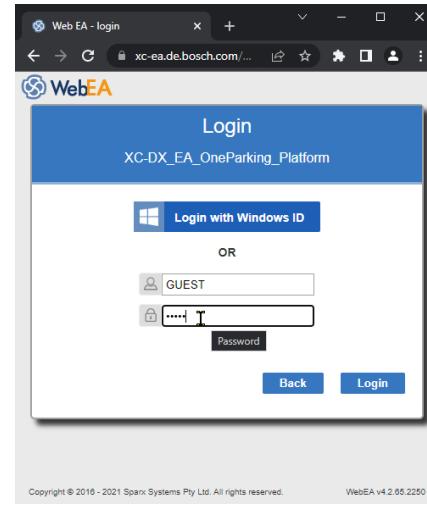


Last updated: 2023-05-23.

Unsynchronized screenshot of diagram, current version is only available via WebEA,
see link below,

Original diagram: [OneParking Platform Overview \(WebEA\)](#)

Enterprise Architect Model for the OneParking Platform, accessible via WebEA

| | |
|---|--|
|  <p>The screenshot shows the 'Select Model' page of WebEA. A list of models is displayed in a sidebar, with 'XC-DX_EA_OneParking_Platform' highlighted in blue and labeled 'Model Number: 55'. Below the sidebar are three navigation buttons: 'Mobile >', 'Tablet >', and 'Desktop >'. At the bottom, there is a copyright notice: 'Copyright © 2010 - 2021 Sparx Systems Pty Ltd. All rights reserved.' and 'WebEA v4.2.05.2250'.</p> |  <p>The screenshot shows the 'Login' page for the model 'XC-DX_EA_OneParking_Platform'. It features a 'Login with Windows ID' button, a 'GUEST' user field, a password field, and 'Back' and 'Login' buttons. The page includes a copyright notice at the bottom: 'Copyright © 2010 - 2021 Sparx Systems Pty Ltd. All rights reserved.' and 'WebEA v4.2.05.2250'.</p> |
| <ul style="list-style-type: none">• URL: https://xc-ea.de.bosch.com/WebEA/login.php• Model Number: 55• Model Name: XC-DX_EA_OneParking_Platform | <p>Read-only access is available for everyone</p> <ul style="list-style-type: none">• User: GUEST• Password: GUEST <p>Registered users can also use "Login with Windows ID"</p> |

Info

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