mouse_droid

COLLABORATORS

	TITLE : mouse_droid	

ACTION	*NAME*	*DATE*	*SIGNATURE*
WRITTEN BY		December 13, 2023	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

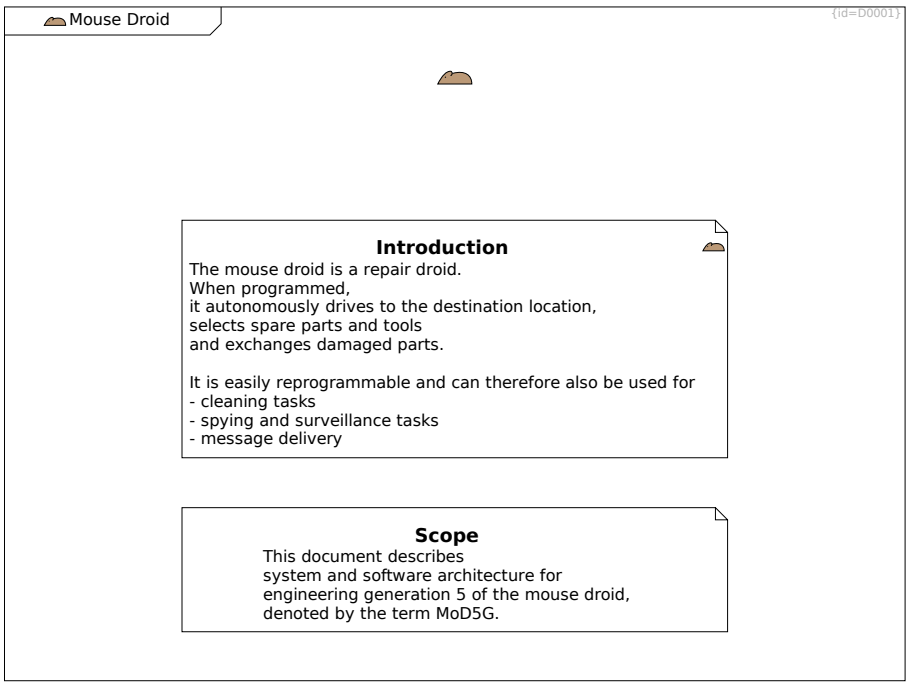
Chapter 1

Mouse Droid

Mouse Droid «mouse-droid» D0001

This document shows a system architecture and software architecture for a mouse droid. This is a small repair droid similar to the MSE-6 used on death star 1.

The document follows the arc42.org template proposed by Gernot Starke and Peter Hruschka,



Introduction «mouse-droid» C0001 *(appears in Chapter 1: Mouse Droid)*

The mouse droid is a repair droid. When programmed, it autonomously drives to the destination location, selects spare parts and tools and exchanges damaged parts.

It is easily reprogrammable and can therefore also be used for

- cleaning tasks

- spying and surveillance tasks

- message delivery

Scope C0141 *(appears in Chapter 1: Mouse Droid)*

This document describes system and software architecture for engineering generation 5 of the mouse droid, denoted by the term MoD5G.

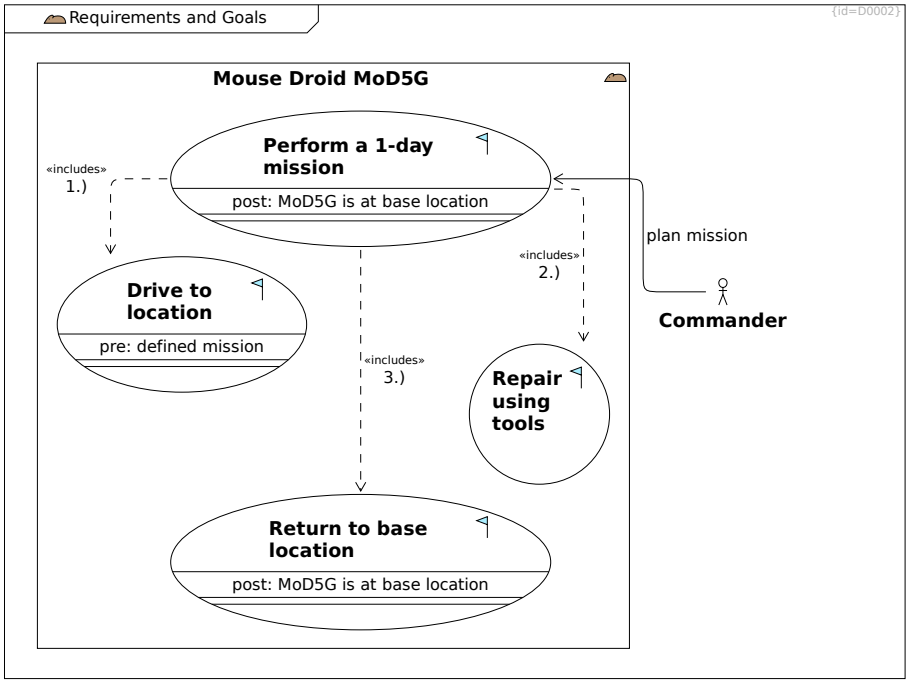
«mouse-droid» C0173 *(appears in Chapter 1: Mouse Droid)*

1.1 Requirements and Goals

Requirements and Goals «mouse-droid» D0002

This section gives a short overview on the project goals (Problem Space, System Level L1)

Primary purpose of the MoD5G is to autonomously repair mechanical things.



Mouse Droid MoD5G «mouse-droid» C0004 *(appears in Section 1.5.3.1: Requirements and Goals, Section 1.1: Requirements and Goals, Section 1.5.3.3: Scope and Context, Section 1.7: Deployment View)*

The Mouse Droid (MoD5G) is a repair droid that can be instructed to perform a mission and which then autonomously selects tactics to achieve the mission goals.

+-- Perform a 1-day mission R0001

+-- Return to base location R0283

+-- Drive to location R0003

+-- Repair using tools R0004

Commander C0005 *(appears in Section 1.1: Requirements and Goals)*

The commander instructs the MoD5G on the mission to perform.

plan mission --> Perform a 1-day mission R0002

provide mission goals and high-level strategy

Perform a 1-day mission «goal» C0006 *(appears in Section 1.1: Requirements and Goals)*

The mouse droid is able to perform a mission that takes several hours. The energy resources of the MoD5G last for up to one terrestrial day.

- The commander programs a mission
- The mouse droid drives to the first location
- The mouse droid uses tools to remove a defective part
- The mouse droid installes a spare part
- Above steps are repeated for other goals
- The mouse droid returns to its base location (see Section 1.12: Glossary)

post: MoD5G is at base location F0034

1.) ·-> Drive to location R0005

2.) ·-> Repair using tools R0006

3.) ·-> Return to base location R0284

Drive to location «goal» C0007 *(appears in Section 1.5.3.1: Requirements and Goals, Section 1.1.1: L1 Requirements View, Section 1.1: Requirements and Goals)*

The mouse droid can explore its environment and calculate a route from its actual position to the target location.

- The mouse droid explores its environment
- The mouse droid enriches an internally memorized map
- The mouse droid calculates a route
- The mouse droid drives along the calculated route
- The mouse droid re-caclulates the route in case of new environment data
- The mouse droid reaches the target location

pre: defined mission F0032

Repair using tools «goal» C0008 *(appears in Section 1.5.3.1: Requirements and Goals, Section 1.1.1: L1 Requirements View, Section 1.1: Requirements and Goals)*

The mouse droid has a couple of tools inside its chassis.

- The mouse droid uses a screw diver to untighten damaged parts
- The mouse droid uses a gripper to move the damaged part out of the way
- The mouse droid uses a gripper to put a spare part from its internal cargo bin to the target place
- The mouse droid uses a screw diver to tighten replaced parts
- The mouse droid uses a gripper to move the damaged part into its internal cargo bin.

(see Section 1.5.1: L2 Mechanics)

Return to base location «goal» C0177 *(appears in Section 1.1.1: L1 Requirements View, Section 1.1: Requirements and Goals)*

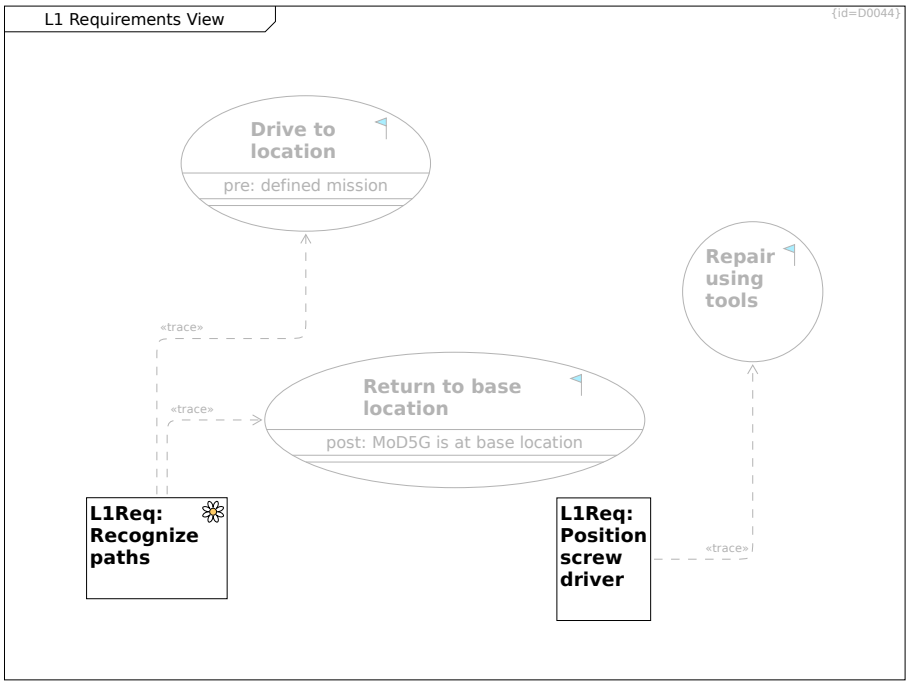
see Section 1.12: Glossary.

post: MoD5G is at base location F0033

1.1.1 L1 Requirements View

L1 Requirements View D0044

This diagram shows examples of system level L1 requirements.



Drive to location «goal» C0007 *(appears in Section 1.5.3.1: Requirements and Goals, Section 1.1.1: L1 Requirements View, Section 1.1: Requirements and Goals)*

The mouse droid can explore its environment and calculate a route from its actual position to the target location.

- The mouse droid explores its environment
- The mouse droid enriches an internally memorized map
- The mouse droid calculates a route
- The mouse droid drives along the calculated route
- The mouse droid re-caclulates the route in case of new environment data
- The mouse droid reaches the target location

pre: defined mission F0032

Repair using tools «goal» C0008 *(appears in Section 1.5.3.1: Requirements and Goals, Section 1.1.1: L1 Requirements View, Section 1.1: Requirements and Goals)*

The mouse droid has a couple of tools inside its chassis.

- The mouse droid uses a screw diver to untighten damaged parts
- The mouse droid uses a gripper to move the damaged part out of the way
- The mouse droid uses a gripper to put a spare part from its internal cargo bin to the target place
- The mouse droid uses a screw diver to tighten replaced parts
- The mouse droid uses a gripper to move the damaged part into its internal cargo bin.

(see Section 1.5.1: L2 Mechanics)

L1Req: Recognize paths «env-perception» C0149 *(appears in Section 1.1.1: L1 Requirements View, Section 1.5.3.1.1: L3 Requirements)*

The MoDG5 shall use redundant sensor data to calculate paths that it can drive along.

··> Drive to location R0244

··> Return to base location R0285

L1Req: Position screw driver C0150 *(appears in Section 1.1.1: L1 Requirements View, Section 1.5.3.1.1: L3 Requirements)*

The MoDG5 shall bring the screw driver into a given 3D position.

··> Repair using tools R0245

Return to base location «goal» C0177 *(appears in Section 1.1.1: L1 Requirements View, Section 1.1: Requirements and Goals)*

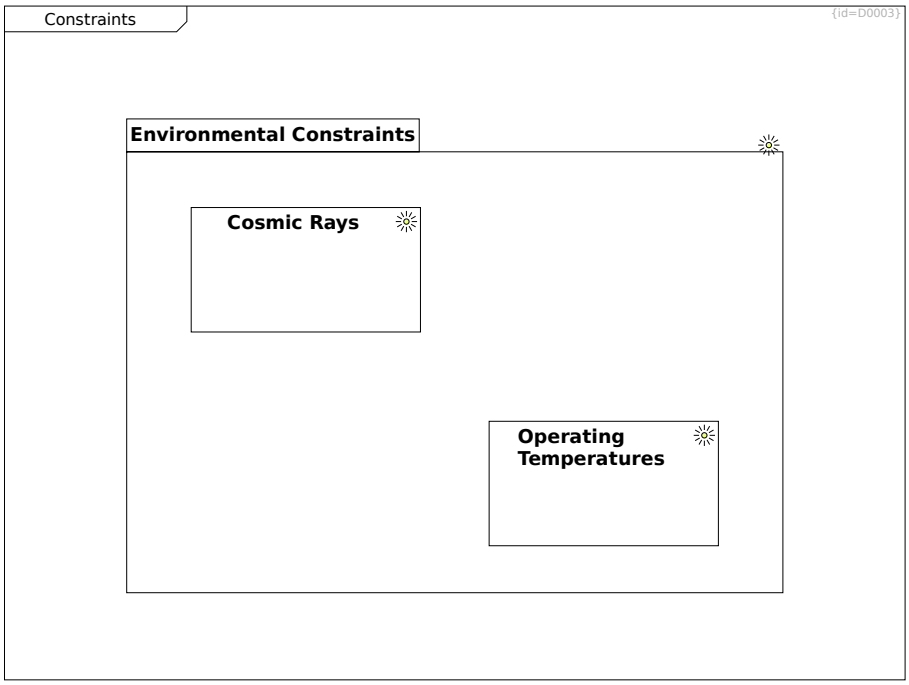
see Section 1.12: Glossary.

post: MoD5G is at base location F0033

1.2 Constraints

Constraints D0003

This section explains the major obstacles that need to be considered when designing a solution that addresses the project goals. (Problem Space, System Level L1)



Operating Temperatures «environment» C0003 *(appears in Section 1.2: Constraints)*

The droid shall be fully functional in the range 240K..360K, it shall survive temperatures from 200K to 400K.

Cosmic Rays «environment» C0002 *(appears in Section 1.2: Constraints, Section 1.9: Architecture Decisions)*

The droid shall ensure data and program integrity.

It shall continue operation after cosmic rays have interfered with data storage or program execution.

Corrupted data must not be stored permanently.

Environmental Constraints «environment» C0144 *(appears in Section 1.2: Constraints)*

This package lists technical/physical constraints imposed by the environment in which to operate.

+-- Cosmic Rays R0221

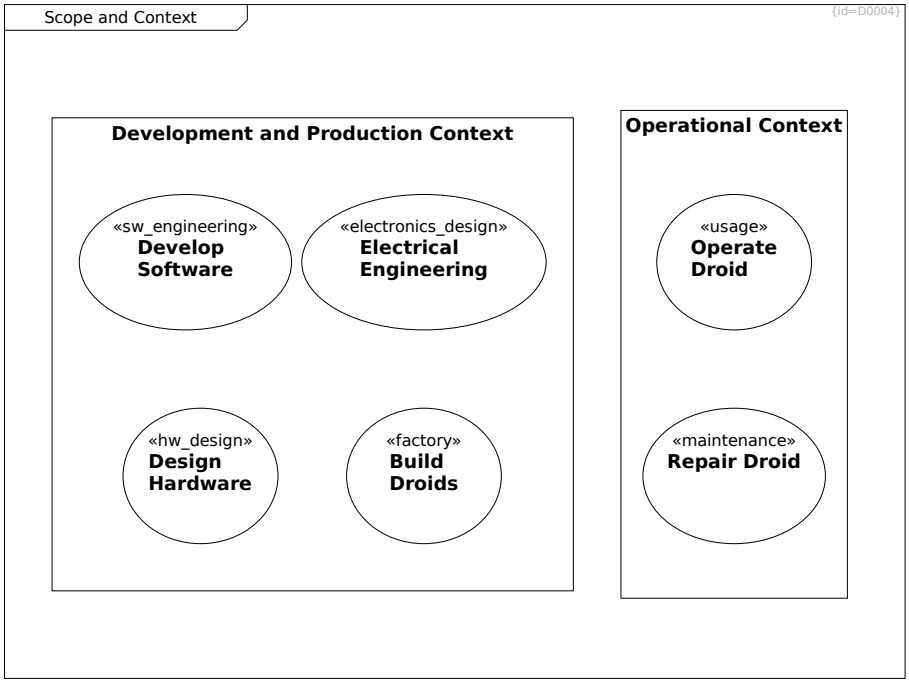
+-- Operating Temperatures R0222

Group of constratins imposed by the operation environment

1.3 Scope and Context

Scope and Context D0004

This section shows the organizational contexts of development and operational environments. (Problem Space, System Level L1)



Development and Production Context C0088 *(appears in Section 1.3: Scope and Context)*

This boundary encompasses the topics that are in scope during development and production.

+-- Electrical Engineering R0188

+-- Design Hardware R0187

+-- Develop Software R0125

+-- Build Droids R0124

Operational Context C0089 *(appears in Section 1.3: Scope and Context)*

This boundary encompasses the topics that are in scope during operation and maintenance.

+-- Repair Droid R0127

+-- Operate Droid R0126

Build Droids «factory» C0090 *(appears in Section 1.3: Scope and Context)*

At the factory, workers assemble hardware and electronic parts to mouse droids and integrate control logic and data.

Develop Software «sw_engineering» C0091 *(appears in Section 1.3: Scope and Context)*

A team of engineers designs, produces and tests the control logic and factory data of the droids.

Operate Droid «usage» C0092 *(appears in Section 1.3: Scope and Context)*

An operator/commander instructs the mouse droid which mission to perform. The logic of the mouse droid translates this mission into driving maneuvers and actions of the integrated tools.

Repair Droid «maintenance» C0093 *(appears in Section 1.3: Scope and Context)*

A service mechanic analyzes the health state of a mouse droid. Depending on the outcome, oil is refilled, logic is updated, parts are exchanged or the whole droid is disintegrated.

Design Hardware «hw_design» C0135 *(appears in Section 1.3: Scope and Context)*

A team of engineers designs, produces and tests the mechanical hardware parts of the droids.

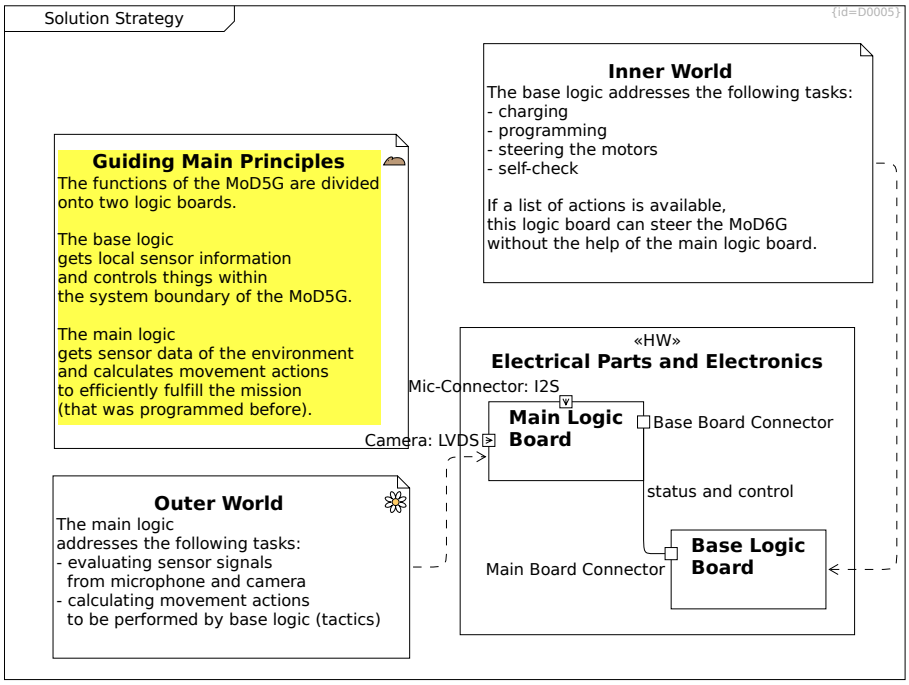
Electrical Engineering «electronics_design» C0136 *(appears in Section 1.3: Scope and Context)*

A team of engineers designs, produces and tests the electrical and electronic parts of the droids.

1.4 Solution Strategy

Solution Strategy D0005

This section shows the fundamental base principle of the system design. (Solution Space, System Level L1)



Electrical Parts and Electronics «HW» C0019 *(appears in Section 1.5.1: L2 Mechanics, Section 1.4.3: L1 Physical View, Section 1.4: Solution Strategy, Section 1.5: Building Block View, Section 1.5.2: L2 Electric Parts and Electronics)*

The main components of the electric parts are:

- a set of cables and connectors
- a set of sensors and actuators
- the energy cell
- two printed circuit boards (PCB) containing the electronic parts

+-- Base Logic Board R0048

+-- Main Logic Board R0047

Base Logic Board C0048 *(appears in Section 1.5.2.1: Base Logic Board, Section 1.5.3.1.1: L3 Requirements, Section 1.4.3: L1 Physical View, Section 1.4: Solution Strategy, Section 1.5.2: L2 Electric Parts and Electronics)*

The base logic board consists of several electronic parts shown in Section 1.5.2.1: Base Logic Board.

Main Board Connector F0031

Main Logic Board C0047 *(appears in Section 1.5.2.2: Main Board Logic, Section 1.5.3.1.1: L3 Requirements, Section 1.4.3: L1 Physical View, Section 1.4: Solution Strategy, Section 1.5.2: L2 Electric Parts and Electronics)*

The main logic board consists of several electronic parts shown in Section 1.5.2.2: Main Board Logic.

Base Board Connector F0008

Mic-Connector «I2S» F0002

Camera «LVDS» F0001

status and control -- Base Logic Board R0280

Outer World «env-perception» C0094 *(appears in Section 1.4: Solution Strategy)*

The main logic addresses the following tasks:

- evaluating sensor signals from microphone and camera

- calculating movement actions to be performed by base logic (tactics)

·-> Main Logic Board R0184

Guiding Main Principles «mouse-droid» C0116 *(appears in Section 1.4: Solution Strategy)*

The functions of the MoD5G are divided onto two logic boards.

The base logic gets local sensor information and controls things within the system boundary of the MoD5G.

The main logic gets sensor data of the environment and calculates movement actions to efficiently fulfill the mission (that was programmed before).

Inner World C0117 *(appears in Section 1.4: Solution Strategy)*

The base logic addresses the following tasks:

- charging

- programming

- steering the motors

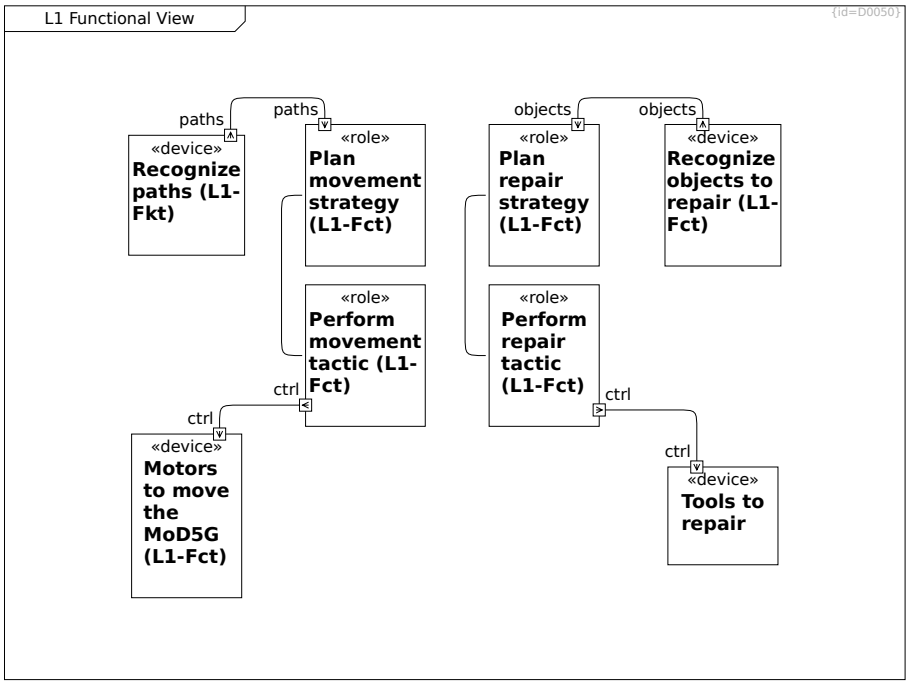
- self-check

If a list of actions is available, this logic board can steer the MoD6G without the help of the main logic board.

·-> Base Logic Board R0183

1.4.1 L1 Functional View

L1 Functional View D0050



Motors to move the MoD5G (L1-Fct) «device» C0178 *(appears in Section 1.4.1: L1 Functional View)*

ctrl F0041

Recognize paths (L1-Fkt) «device» C0179 *(appears in Section 1.4.1: L1 Functional View)*

paths F0035

-- Plan movement strategy (L1-Fct) R0286

Recognize objects to repair (L1-Fct) «device» C0180 *(appears in Section 1.4.1: L1 Functional View)*

objects F0037

-- Plan repair strategy (L1-Fct) R0287

Plan movement strategy (L1-Fct) «role» C0181 *(appears in Section 1.4.1: L1 Functional View, Section 1.4.2: L1 Logical View)*

paths F0036

-- Perform movement tactic (L1-Fct) R0288

Plan repair strategy (L1-Fct) «role» C0182 *(appears in Section 1.4.1: L1 Functional View, Section 1.4.2: L1 Logical View)*

objects F0038

-- Perform repair tactic (L1-Fct) R0289

Perform movement tactic (L1-Fct) «role» C0183 *(appears in Section 1.4.1: L1 Functional View, Section 1.4.2: L1 Logical View)*

ctrl F0042

-- Motors to move the MoD5G (L1-Fct) R0290

Perform repair tactic (L1-Fct) «role» C0184 *(appears in Section 1.4.1: L1 Functional View, Section 1.4.2: L1 Logical View)*

ctrl F0039

-- Tools to repair R0291

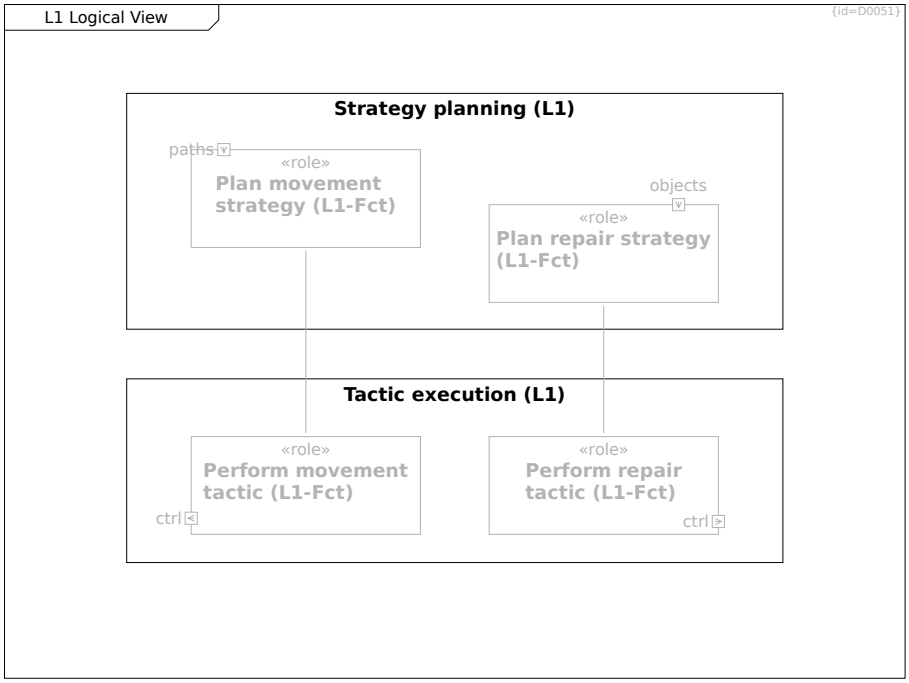
Tools to repair «device» C0185 *(appears in Section 1.4.1: L1 Functional View)*

ctrl F0040

1.4.2 L1 Logical View

L1 Logical View D0051

This view maps the functions to a form.



Plan movement strategy (L1-Fct) «role» C0181 *(appears in Section 1.4.1: L1 Functional View, Section 1.4.2: L1 Logical View)*

paths F0036

-- Perform movement tactic (L1-Fct) R0288

Plan repair strategy (L1-Fct) «role» C0182 *(appears in Section 1.4.1: L1 Functional View, Section 1.4.2: L1 Logical View)*

objects F0038

-- Perform repair tactic (L1-Fct) R0289

Perform repair tactic (L1-Fct) «role» C0184 *(appears in Section 1.4.1: L1 Functional View, Section 1.4.2: L1 Logical View)*

ctrl F0039

Perform movement tactic (L1-Fct) «role» C0183 *(appears in Section 1.4.1: L1 Functional View, Section 1.4.2: L1 Logical View)*

ctrl F0042

Strategy planning (L1) C0186 *(appears in Section 1.4.2: L1 Logical View, Section 1.4.3: L1 Physical View)*

+-- Plan movement strategy (L1-Fct) R0292

+-- Plan repair strategy (L1-Fct) R0293

Tactic execution (L1) C0187 *(appears in Section 1.4.2: L1 Logical View, Section 1.4.3: L1 Physical View)*

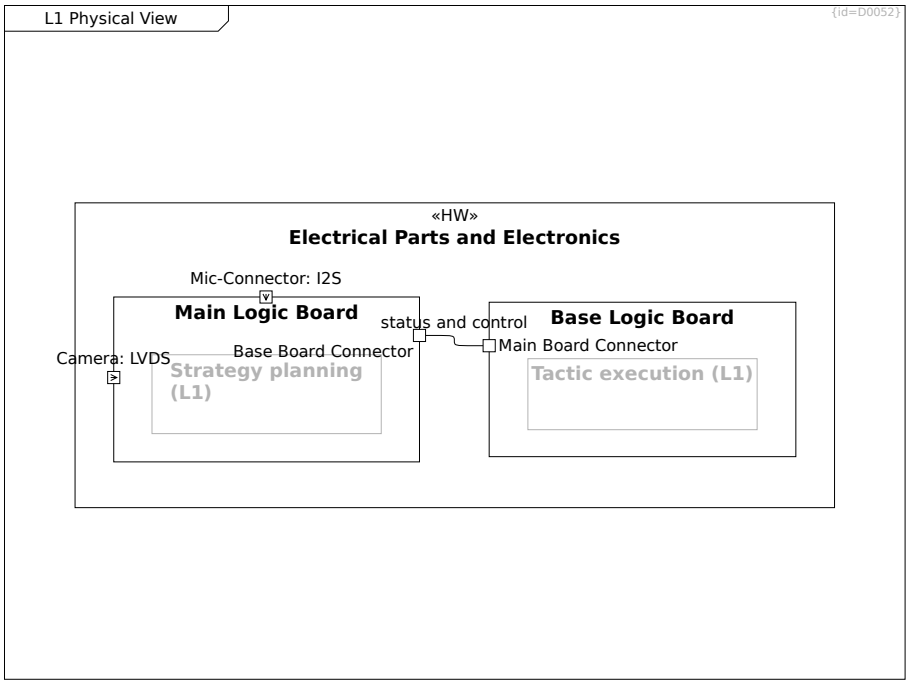
+-- Perform movement tactic (L1-Fct) R0294

+-- Perform repair tactic (L1-Fct) R0295

1.4.3 L1 Physical View

L1 Physical View D0052

This view maps the logical form to physical artifacts.



Electrical Parts and Electronics «HW» C0019 *(appears in Section 1.5.1: L2 Mechanics, Section 1.4.3: L1 Physical View, Section 1.4: Solution Strategy, Section 1.5: Building Block View, Section 1.5.2: L2 Electric Parts and Electronics)*

The main components of the electric parts are:

- a set of cables and connectors
- a set of sensors and actuators
- the energy cell
- two printed circuit boards (PCB) containing the electronic parts

+-- Base Logic Board R0048

+-- Main Logic Board R0047

Base Logic Board C0048 *(appears in Section 1.5.2.1: Base Logic Board, Section 1.5.3.1.1: L3 Requirements, Section 1.4.3: L1 Physical View, Section 1.4: Solution Strategy, Section 1.5.2: L2 Electric Parts and Electronics)*

The base logic board consists of several electronic parts shown in Section 1.5.2.1: Base Logic Board.

Main Board Connector F0031

+-- Tactic execution (L1) R0297

Main Logic Board C0047 *(appears in Section 1.5.2.2: Main Board Logic, Section 1.5.3.1.1: L3 Requirements, Section 1.4.3: L1 Physical View, Section 1.4: Solution Strategy, Section 1.5.2: L2 Electric Parts and Electronics)*

The main logic board consists of several electronic parts shown in Section 1.5.2.2: Main Board Logic.

Base Board Connector F0008

Mic-Connector «I2S» F0002

Camera «LVDS» F0001

status and control -- Base Logic Board R0280

+-- Strategy planning (L1) R0296

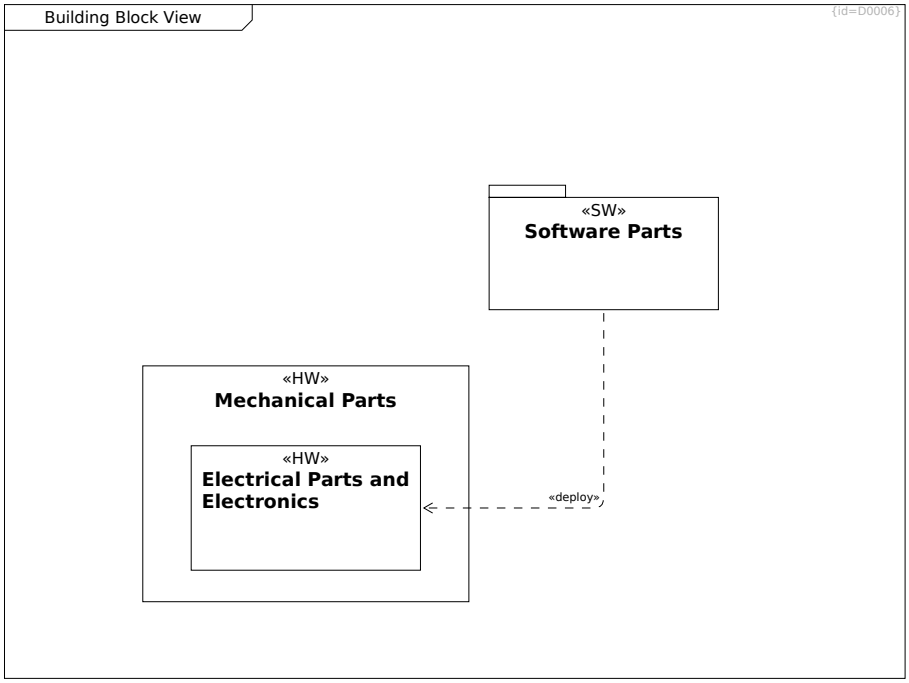
Tactic execution (L1) C0187 *(appears in Section 1.4.2: L1 Logical View, Section 1.4.3: L1 Physical View)*

Strategy planning (L1) C0186 *(appears in Section 1.4.2: L1 Logical View, Section 1.4.3: L1 Physical View)*

1.5 Building Block View

Building Block View D0006

This section shows the parts of the MoD5G system (Solution Space, System Level L1)



Mechanical Parts «HW» C0018 *(appears in Section 1.5.1: L2 Mechanics, Section 1.5: Building Block View)*

The mechanical parts encompass all parts of which the mouse droid consists. From outside, the chassis and wheels are the most obvious parts. Inside, a skeleton frame provides stability of the assembly.

Important charactereistics are weight, operating temperature range, durability, stability.

+-- Electrical Parts and Electronics R0013

Electrical Parts and Electronics «HW» C0019 *(appears in Section 1.5.1: L2 Mechanics, Section 1.4.3: L1 Physical View, Section 1.4: Solution Strategy, Section 1.5: Building Block View, Section 1.5.2: L2 Electric Parts and Electronics)*

The main components of the electric parts are:

- a set of cables and connectors

- a set of sensors and actuators

- the energy cell

- two printed circuit boards (PCB) containing the electronic parts

Software Parts «SW» C0020 *(appears in Section 1.5: Building Block View, Section 1.5.3.4: Solution Strategy, Section 1.5.3.5: Building Block View)*

The software consists of control logic, initial data that was integrated at the factory and learned data that is aggregated during operation.

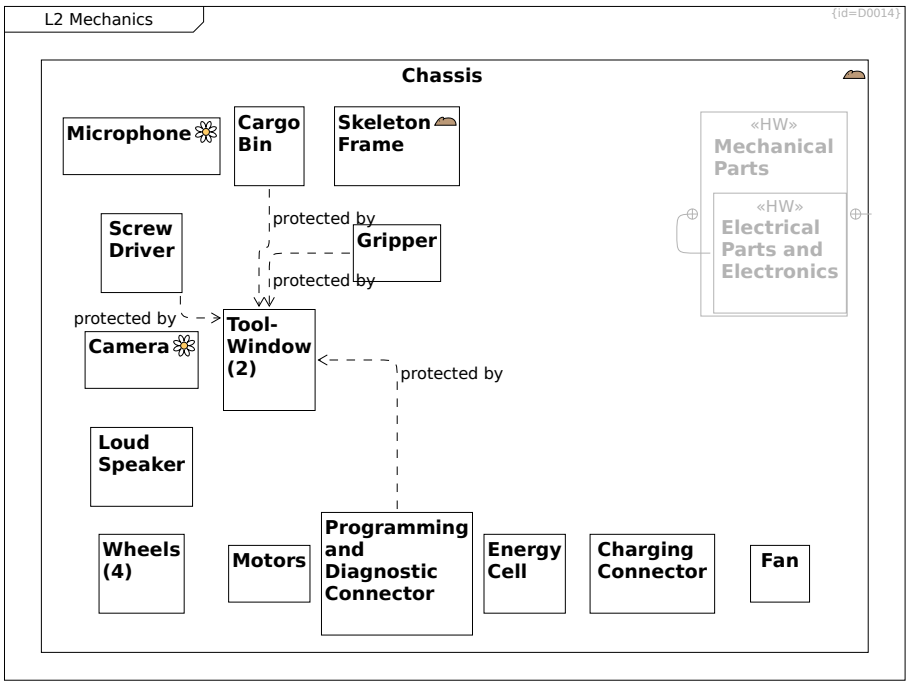
These software items are split over two logical boards and subdivided into independent execution partitions.

·-> Electrical Parts and Electronics R0243

1.5.1 L2 Mechanics

L2 Mechanics D0014

This section shows the mechanical parts of the MoD5G system (Solution Space, System Level L2)



Chassis «mouse-droid» C0035 *(appears in Section 1.5.1: L2 Mechanics)*

+-- Skeleton Frame R0228

+-- Tool-Window (2) R0065

+-- Gripper R0064

+-- Cargo Bin R0034

+-- Microphone R0040

+-- Energy Cell R0044

+-- Programming and Diagnostic Connector R0042

+-- Motors R0038

+-- Loud Speaker R0041

+-- Charging Connector R0043

+-- Camera R0039

+-- Wheels (4) R0036

+-- Fan R0037

+-- Screw Driver R0035

+-- Electrical Parts and Electronics R0046

Cargo Bin C0036 *(appears in Section 1.5.1: L2 Mechanics)*

protected by ·-> Tool-Window (2) R0224

Screw Driver C0037 *(appears in Section 1.5.1: L2 Mechanics, Section 1.5.3.1.1: L3 Requirements)*

protected by ·-> Tool-Window (2) R0225

Wheels (4) C0038 *(appears in Section 1.5.1: L2 Mechanics)*

Fan C0039 *(appears in Section 1.5.1: L2 Mechanics)*

A fan prevents overheating in hot environment conditions.

Motors C0040 *(appears in Section 1.5.1: L2 Mechanics, Section 1.5.2: L2 Electric Parts and Electronics)*

Camera «env-perception» C0041 *(appears in Section 1.5.1: L2 Mechanics, Section 1.5.3.1.1: L3 Requirements, Section 1.5.2: L2 Electric Parts and Electronics)*

Microphone «env-perception» C0042 *(appears in Section 1.5.1: L2 Mechanics, Section 1.5.2: L2 Electric Parts and Electronics)*

Loud Speaker C0043 *(appears in Section 1.5.1: L2 Mechanics, Section 1.5.2: L2 Electric Parts and Electronics)*

Programming and Diagnostic Connector C0044 *(appears in Section 1.5.1: L2 Mechanics, Section 1.5.2: L2 Electric Parts and Electronics)*

protected by ·-> Tool-Window (2) R0226

Charging Connector C0045 *(appears in Section 1.5.1: L2 Mechanics, Section 1.5.2: L2 Electric Parts and Electronics)*

Energy Cell C0046 *(appears in Section 1.5.1: L2 Mechanics, Section 1.5.2: L2 Electric Parts and Electronics)*

Mechanical Parts «HW» C0018 *(appears in Section 1.5.1: L2 Mechanics, Section 1.5: Building Block View)*

The mechanical parts encompass all parts of which the mouse droid consists. From outside, the chassis and wheels are the most obvious parts. Inside, a skeleton frame provides stability of the assembly.

Important charactereistics are weight, operating temperature range, durability, stability.

+-- Electrical Parts and Electronics R0013

+-- Chassis R0045

Electrical Parts and Electronics «HW» C0019 *(appears in Section 1.5.1: L2 Mechanics, Section 1.4.3: L1 Physical View, Section 1.4: Solution Strategy, Section 1.5: Building Block View, Section 1.5.2: L2 Electric Parts and Electronics)*

The main components of the electric parts are:

- a set of cables and connectors
- a set of sensors and actuators
- the energy cell
- two printed circuit boards (PCB) containing the electronic parts

Gripper C0054 *(appears in Section 1.5.1: L2 Mechanics)*

A tool that allows to grab objects and move them.

protected by ·-> Tool-Window (2) R0223

Tool-Window (2) C0055 *(appears in Section 1.5.1: L2 Mechanics)*

A tool window is a flap in the chassis that protects screw driver, gripper and cargo bin when unused. It can be opened and closed by a motor.

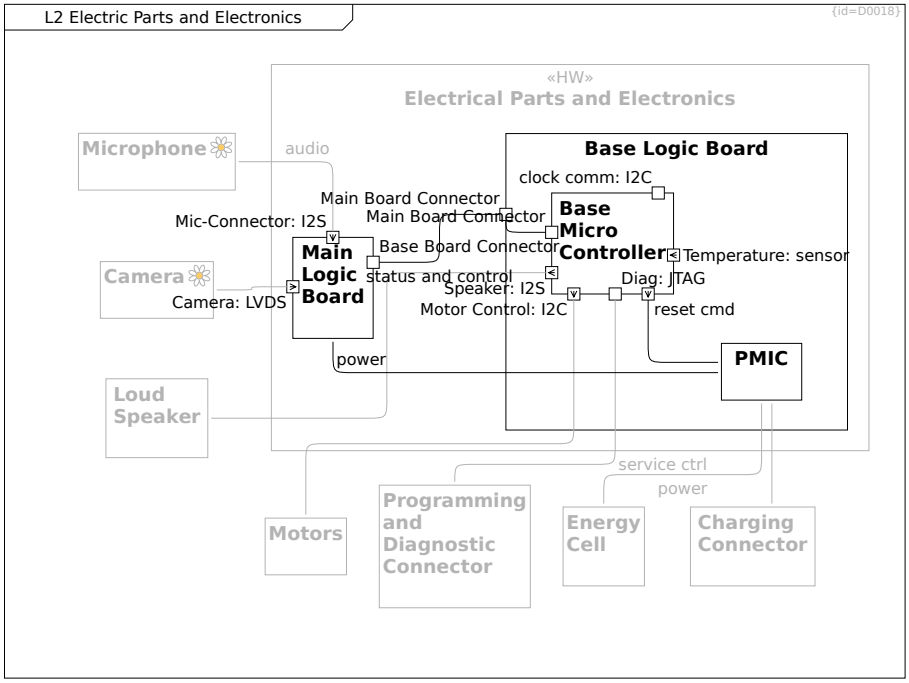
Skeleton Frame «mouse-droid» C0146 *(appears in Section 1.5.1: L2 Mechanics)*

The skeleton frame provides stability to the assembly of parts. Anchorage points latch the assembled parts at their positions.

1.5.2 L2 Electric Parts and Electronics

L2 Electric Parts and Electronics D0018

This section shows the electric parts and electronics of the MoD5G system (Solution Space, System Level L2)



Electrical Parts and Electronics «HW» C0019 *(appears in Section 1.5.1: L2 Mechanics, Section 1.4.3: L1 Physical View, Section 1.4: Solution Strategy, Section 1.5: Building Block View, Section 1.5.2: L2 Electric Parts and Electronics)*

The main components of the electric parts are:

- a set of cables and connectors
- a set of sensors and actuators
- the energy cell
- two printed circuit boards (PCB) containing the electronic parts

+-- Base Logic Board R0048

+-- Main Logic Board R0047

Energy Cell C0046 *(appears in Section 1.5.1: L2 Mechanics, Section 1.5.2: L2 Electric Parts and Electronics)*

power -- PMIC R0050

Loud Speaker C0043 *(appears in Section 1.5.1: L2 Mechanics, Section 1.5.2: L2 Electric Parts and Electronics)*

Camera «env-perception» C0041 *(appears in Section 1.5.1: L2 Mechanics, Section 1.5.3.1.1: L3 Requirements, Section 1.5.2: L2 Electric Parts and Electronics)*

-- Main Logic Board R0051

Microphone «env-perception» C0042 *(appears in Section 1.5.1: L2 Mechanics, Section 1.5.2: L2 Electric Parts and Electronics)*

audio -- Main Logic Board R0056

Motors C0040 *(appears in Section 1.5.1: L2 Mechanics, Section 1.5.2: L2 Electric Parts and Electronics)*

Charging Connector C0045 *(appears in Section 1.5.1: L2 Mechanics, Section 1.5.2: L2 Electric Parts and Electronics)*

-- PMIC R0057

Programming and Diagnostic Connector C0044 *(appears in Section 1.5.1: L2 Mechanics, Section 1.5.2: L2 Electric Parts and Electronics)*

service ctrl -- Base Micro Controller R0061

Main Logic Board C0047 *(appears in Section 1.5.2.2: Main Board Logic, Section 1.5.3.1.1: L3 Requirements, Section 1.4.3: L1 Physical View, Section 1.4: Solution Strategy, Section 1.5.2: L2 Electric Parts and Electronics)*

The main logic board consists of several electronic parts shown in Section 1.5.2.2: Main Board Logic.

Base Board Connector F0008

Mic-Connector «I2S» F0002

Camera «LVDS» F0001

status and control -- Base Logic Board R0280

Base Logic Board C0048 *(appears in Section 1.5.2.1: Base Logic Board, Section 1.5.3.1.1: L3 Requirements, Section 1.4.3: L1 Physical View, Section 1.4: Solution Strategy, Section 1.5.2: L2 Electric Parts and Electronics)*

The base logic board consists of several electronic parts shown in Section 1.5.2.1: Base Logic Board.

Main Board Connector F0031

+-- Base Micro Controller R0053

+-- PMIC R0049

-- Base Micro Controller R0279

PMIC C0049 *(appears in Section 1.5.2.1: Base Logic Board, Section 1.5.2: L2 Electric Parts and Electronics)*

Power Management Integrated Circuit

power -- Main Logic Board R0063

Base Micro Controller C0051 *(appears in Section 1.5.2.1: Base Logic Board, Section 1.5.3.1: Requirements and Goals, Section 1.5.2: L2 Electric Parts and Electronics, Section 1.5.3: L2 Software [subsystem])*

The base micro controller consists of

- logic unit
- data storage
- persistent data+logic storage
- self supervision (by ECC and lockstep-cores)
- HW watchdog
- clock
- temperature sensor
- io ports

This block is optimized for providing a reliable execution of algorithms, see Section 1.10: Quality Requirements.

Temperature «sensor» F0010

clock comm «I2C» F0007

Main Board Connector F0009

Speaker «I2S» F0004

Motor Control «I2C» F0003

Diag «JTAG» F0005

reset cmd F0006

-- Motors R0059

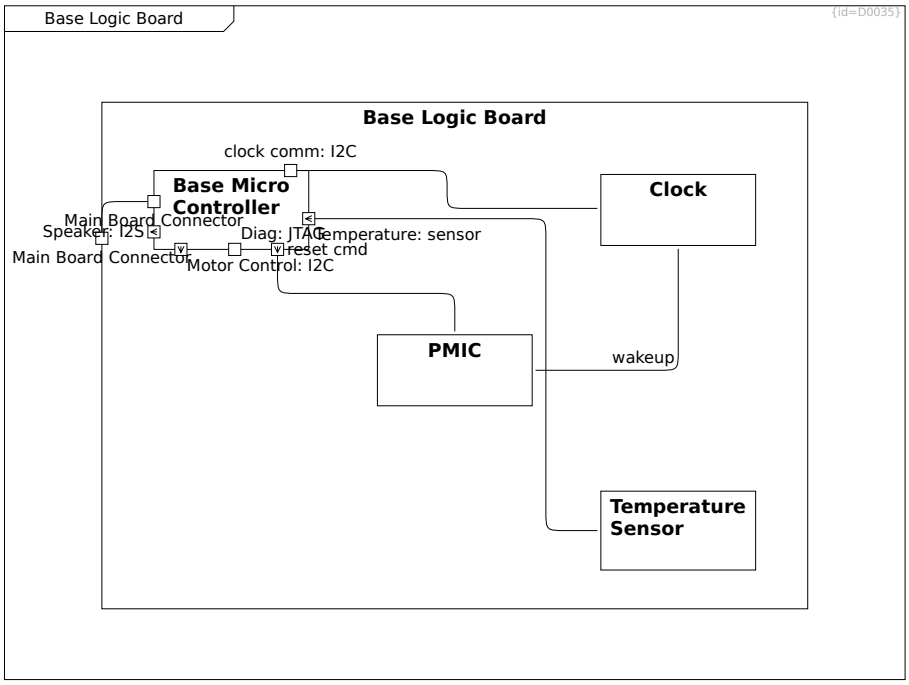
-- Loud Speaker R0060

-- PMIC R0062

1.5.2.1 Base Logic Board

Base Logic Board D0035

This section shows the base board logic of the MoD5G system (Solution Space, System Level L3)



Base Logic Board C0048 *(appears in Section 1.5.2.1: Base Logic Board, Section 1.5.3.1.1: L3 Requirements, Section 1.4.3: L1 Physical View, Section 1.4: Solution Strategy, Section 1.5.2: L2 Electric Parts and Electronics)*

The base logic board consists of several electronic parts shown in Section 1.5.2.1: Base Logic Board.

Main Board Connector F0031

+-- Base Micro Controller R0053

+-- Clock R0055

+-- PMIC R0049

+-- Temperature Sensor R0095

-- Base Micro Controller R0279

Base Micro Controller C0051 *(appears in Section 1.5.2.1: Base Logic Board, Section 1.5.3.1: Requirements and Goals, Section 1.5.2: L2 Electric Parts and Electronics, Section 1.5.3: L2 Software [subsystem])*

The base micro controller consists of

- logic unit
- data storage
- persistent data+logic storage
- self supervision (by ECC and lockstep-cores)
- HW watchdog
- clock
- temperature sensor
- io ports

This block is optimized for providing a reliable execution of algorithms, see Section 1.10: Quality Requirements.

Temperature «sensor» F0010

clock comm «I2C» F0007

Main Board Connector F0009

Speaker «I2S» F0004

Motor Control «I2C» F0003

Diag «JTAG» F0005

reset cmd F0006

-- PMIC R0062

-- Clock R0066

Clock C0053 *(appears in Section 1.5.2.1: Base Logic Board)*

wakeup -- PMIC R0067

PMIC C0049 *(appears in Section 1.5.2.1: Base Logic Board, Section 1.5.2: L2 Electric Parts and Electronics)*
Power Management Integrated Circuit

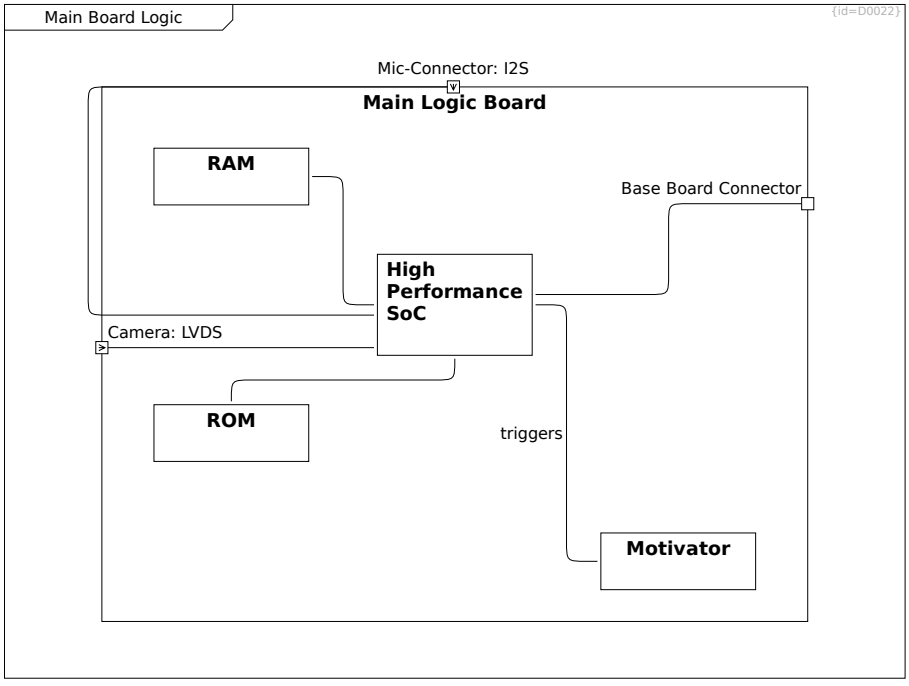
Temperature Sensor C0071 *(appears in Section 1.5.2.1: Base Logic Board)*

-- Base Micro Controller R0096

1.5.2.2 Main Board Logic

Main Board Logic D0022

This section shows the main board logic of the MoD5G system (Solution Space, System Level L3)



Main Logic Board C0047 *(appears in Section 1.5.2.2: Main Board Logic, Section 1.5.3.1.1: L3 Requirements, Section 1.4.3: L1 Physical View, Section 1.4: Solution Strategy, Section 1.5.2: L2 Electric Parts and Electronics)*

The main logic board consists of several electronic parts shown in Section 1.5.2.2: Main Board Logic.

Base Board Connector F0008

Mic-Connector «I2S» F0002

Camera «LVDS» F0001

+-- RAM R0070

+-- ROM R0071

+-- High Performance SoC R0069

+-- Motivator R0077

-- High Performance SoC R0074

-- High Performance SoC R0076

High Performance SoC C0056 *(appears in Section 1.5.2.2: Main Board Logic, Section 1.5.3.1: Requirements and Goals, Section 1.5.3: L2 Software [subsystem])*

This block provides an execution environment for algorithms that is optimized for high performance.

-- RAM R0072

-- ROM R0073

-- Main Logic Board R0075

RAM C0057 *(appears in Section 1.5.2.2: Main Board Logic)*

ROM C0058 *(appears in Section 1.5.2.2: Main Board Logic)*

Motivator C0059 *(appears in Section 1.5.2.2: Main Board Logic)*

A motivator is a basic component needed to keep going on.

triggers -- High Performance SoC R0078

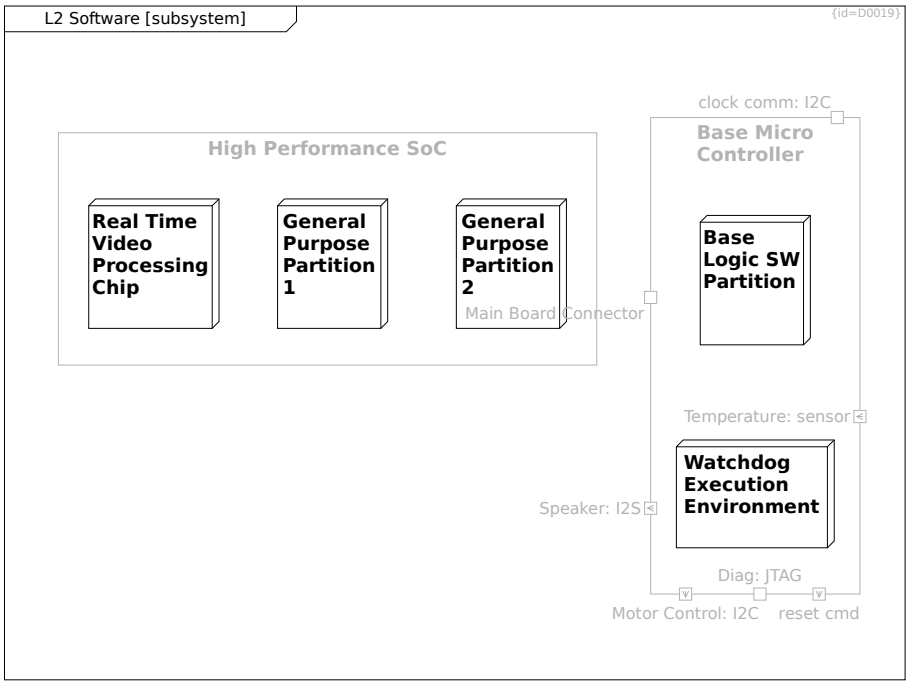
1.5.3 L2 Software [subsystem]

L2 Software [subsystem] D0019

This diagram shows the virtual machines and specialized (non-versatile) execution environments (Solution Space, System Level L2)

These are deployed onto the logic boards shown in Section 1.5.2: L2 Electric Parts and Electronics.

In this section, this view is further detailed to software elements, their relations and interactions.



High Performance SoC C0056 *(appears in Section 1.5.2.2: Main Board Logic, Section 1.5.3.1: Requirements and Goals, Section 1.5.3: L2 Software [subsystem])*

This block provides an execution environment for algorithms that is optimized for high performance.

+-- Real Time Video Processing Chip R0079

+-- General Purpose Partition 1 R0081

+-- General Purpose Partition 2 R0082

Base Micro Controller C0051 *(appears in Section 1.5.2.1: Base Logic Board, Section 1.5.3.1: Requirements and Goals, Section 1.5.2: L2 Electric Parts and Electronics, Section 1.5.3: L2 Software [subsystem])*

The base micro controller consists of

- logic unit
- data storage
- persistent data+logic storage
- self supervision (by ECC and lockstep-cores)
- HW watchdog
- clock
- temperature sensor
- io ports

This block is optimized for providing a reliable execution of algorithms, see Section 1.10: Quality Requirements.

Temperature «sensor» F0010

clock comm «I2C» F0007

Main Board Connector F0009

Speaker «I2S» F0004

Motor Control «I2C» F0003

Diag «JTAG» F0005

reset cmd F0006

+-- Base Logic SW Partition R0083

+-- Watchdog Execution Environment R0080

Real Time Video Processing Chip C0060 *(appears in Section 1.5.3.7: Deployment View, Section 1.5.3: L2 Software [subsystem])*

Watchdog Execution Environment C0061 *(appears in Section 1.5.3: L2 Software [subsystem])*

General Purpose Partition 1 C0062 *(appears in Section 1.5.3.7: Deployment View, Section 1.5.3: L2 Software [subsystem])*

General Purpose Partition 2 C0063 *(appears in Section 1.5.3.7: Deployment View, Section 1.5.3: L2 Software [subsystem])*

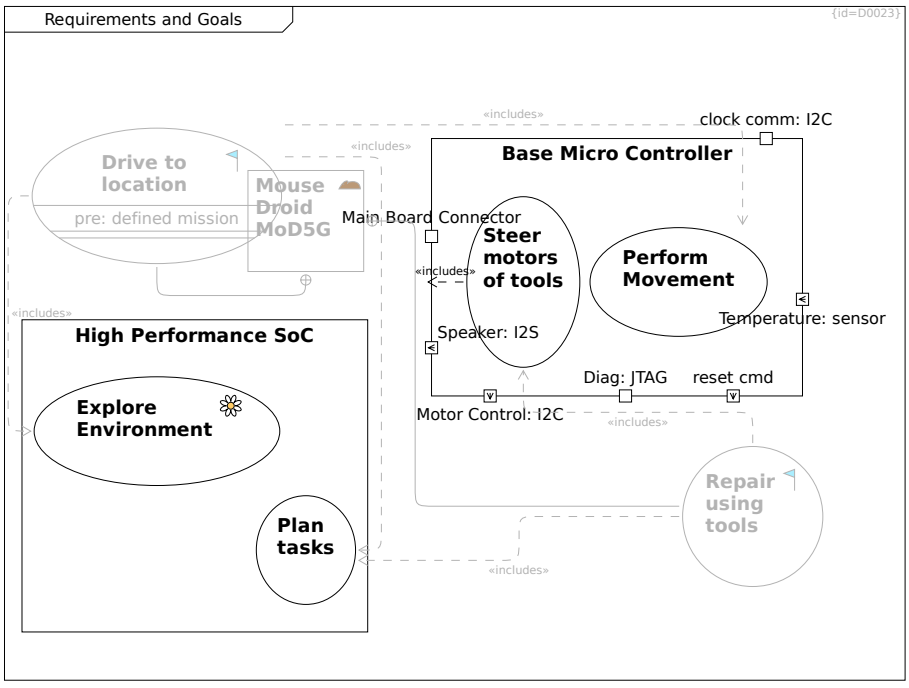
Base Logic SW Partition C0064 *(appears in Section 1.5.3.7: Deployment View, Section 1.5.3: L2 Software [subsystem])*

1.5.3.1 Requirements and Goals

Requirements and Goals D0023

This section shows the goals of the software development for the MoD5G (Problem Space, Software Level L3)

In gray, the use cases on system level L1 are repeated from Section 1.1: Requirements and Goals to show the refinement to software-only use cases shown in black.



Mouse Droid MoD5G «mouse-droid» C0004 *(appears in Section 1.5.3.1: Requirements and Goals, Section 1.1: Requirements and Goals, Section 1.5.3.3: Scope and Context, Section 1.7: Deployment View)*

The Mouse Droid (MoD5G) is a repair droid that can be instructed to perform a mission and which then autonomously selects tactics to achieve the mission goals.

+-- Drive to location R0003

+-- Repair using tools R0004

Drive to location «goal» C0007 *(appears in Section 1.5.3.1: Requirements and Goals, Section 1.1.1: L1 Requirements View, Section 1.1: Requirements and Goals)*

The mouse droid can explore its environment and calculate a route from its actual position to the target location.

- The mouse droid explores its environment
- The mouse droid enriches an internally memorized map
- The mouse droid calculates a route
- The mouse droid drives along the calculated route
- The mouse droid re-caclulates the route in case of new environment data
- The mouse droid reaches the target location

pre: defined mission F0032

·-> Plan tasks R0091

·-> Explore Environment R0234

·-> Perform Movement R0235

Repair using tools «goal» C0008 *(appears in Section 1.5.3.1: Requirements and Goals, Section 1.1.1: L1 Requirements View, Section 1.1: Requirements and Goals)*

The mouse droid has a couple of tools inside its chassis.

- The mouse droid uses a screw diver to untighten damaged parts
- The mouse droid uses a gripper to move the damaged part out of the way
- The mouse droid uses a gripper to put a spare part from its internal cargo bin to the target place
- The mouse droid uses a screw diver to tighten replaced parts
- The mouse droid uses a gripper to move the damaged part into its internal cargo bin.

(see Section 1.5.1: L2 Mechanics)

·-> Plan tasks R0090

·-> Steer motors of tools R0233

Explore Environment «env-perception» C0067 *(appears in Section 1.5.3.1: Requirements and Goals, Section 1.5.3.1.1: L3 Requirements)*

When the mouse droid is missing relevant data on the environment, it plans a list of actions that suits the purpose of gaining the missing knowledge.

Perform Movement C0068 *(appears in Section 1.5.3.1: Requirements and Goals)*

Steer motors of tools C0069 *(appears in Section 1.5.3.1: Requirements and Goals, Section 1.5.3.1.1: L3 Requirements)*

·-> Base Micro Controller R0255

Plan tasks C0070 *(appears in Section 1.5.3.1: Requirements and Goals)*

The mouse droid creates a list of actions to fulfill the given mission. If data on the environment is missing, it plans an explortion task and re-plans the action list later.

High Performance SoC C0056 *(appears in Section 1.5.2.2: Main Board Logic, Section 1.5.3.1: Requirements and Goals, Section 1.5.3: L2 Software [subsystem])*

This block provides an execution environment for algorithms that is optimized for high performance.

+-- Explore Environment R0258

+-- Plan tasks R0259

Base Micro Controller C0051 *(appears in Section 1.5.2.1: Base Logic Board, Section 1.5.3.1: Requirements and Goals, Section 1.5.2: L2 Electric Parts and Electronics, Section 1.5.3: L2 Software [subsystem])*

The base micro controller consists of

- logic unit
- data storage
- persistent data+logic storage
- self supervision (by ECC and lockstep-cores)
- HW watchdog
- clock
- temperature sensor
- io ports

This block is optimized for providing a reliable execution of algorithms, see Section 1.10: Quality Requirements.

Temperature «sensor» F0010

clock comm «I2C» F0007

Main Board Connector F0009

Speaker «I2S» F0004

Motor Control «I2C» F0003

Diag «JTAG» F0005

reset cmd F0006

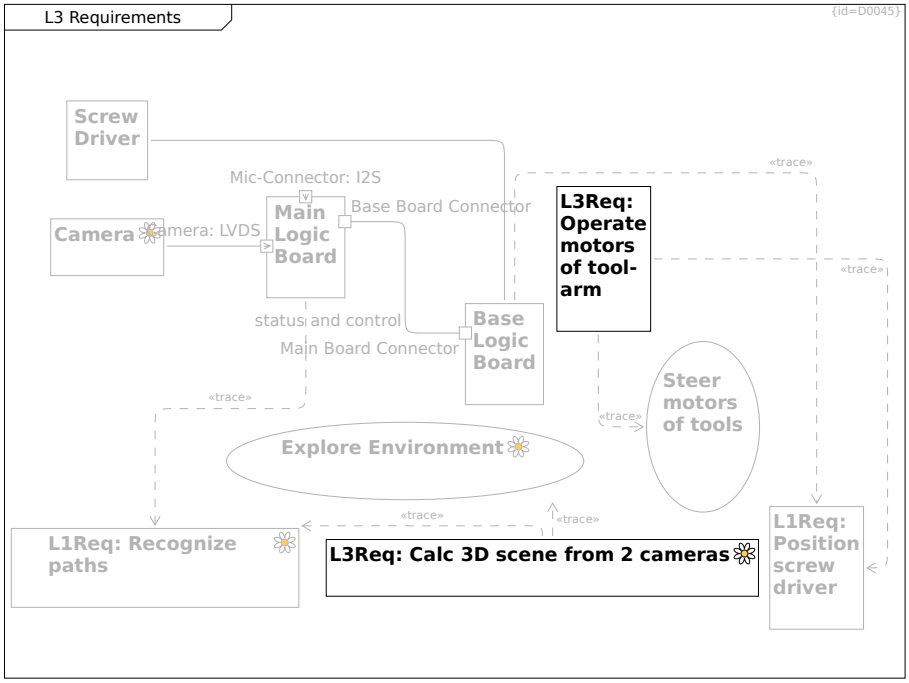
+-- Steer motors of tools R0256

+-- Perform Movement R0257

1.5.3.1.1 L3 Requirements

L3 Requirements D0045

This diagram shows examples of system level L3 requirements.



L1Req: Recognize paths «env-perception» C0149 *(appears in Section 1.1.1: L1 Requirements View, Section 1.5.3.1.1: L3 Requirements)*

The MoDG5 shall use redundant sensor data to calculate paths that it can drive along.

L1Req: Position screw driver C0150 *(appears in Section 1.1.1: L1 Requirements View, Section 1.5.3.1.1: L3 Requirements)*

The MoDG5 shall bring the screw driver into a given 3D position.

Explore Environment «env-perception» C0067 *(appears in Section 1.5.3.1: Requirements and Goals, Section 1.5.3.1.1: L3 Requirements)*

When the mouse droid is missing relevant data on the environment, it plans a list of actions that suits the purpose of gaining the missing knowledge.

Steer motors of tools C0069 *(appears in Section 1.5.3.1: Requirements and Goals, Section 1.5.3.1.1: L3 Requirements)*

L3Req: Calc 3D scene from 2 cameras «env-perception» C0151 *(appears in Section 1.5.3.1.1: L3 Requirements, Section 1.5.3.5.1: Environment Capture)*

The Main Logic Board shall create a 3D model of the environment based on 2 camera images.

·-> Explore Environment R0248

·-> L1Req: Recognize paths R0249

Base Logic Board C0048 *(appears in Section 1.5.2.1: Base Logic Board, Section 1.5.3.1.1: L3 Requirements, Section 1.4.3: L1 Physical View, Section 1.4: Solution Strategy, Section 1.5.2: L2 Electric Parts and Electronics)*

The base logic board consists of several electronic parts shown in Section 1.5.2.1: Base Logic Board.

Main Board Connector F0031

·-> L1Req: Position screw driver R0247

-- Screw Driver R0252

Main Logic Board C0047 *(appears in Section 1.5.2.2: Main Board Logic, Section 1.5.3.1.1: L3 Requirements, Section 1.4.3: L1 Physical View, Section 1.4: Solution Strategy, Section 1.5.2: L2 Electric Parts and Electronics)*

The main logic board consists of several electronic parts shown in Section 1.5.2.2: Main Board Logic.

Base Board Connector F0008

Mic-Connector «I2S» F0002

Camera «LVDS» F0001

·-> L1Req: Recognize paths R0246

status and control -- Base Logic Board R0280

L3Req: Operate motors of tool-arm C0152 *(appears in Section 1.5.3.1.1: L3 Requirements, Section 1.5.3.5.2: System Control)*

The Base Logic Board shall steer the motors of the tool-arm to a given 3D position.

·-> L1Req: Position screw driver R0250

·-> Steer motors of tools R0251

Screw Driver C0037 *(appears in Section 1.5.1: L2 Mechanics, Section 1.5.3.1.1: L3 Requirements)*

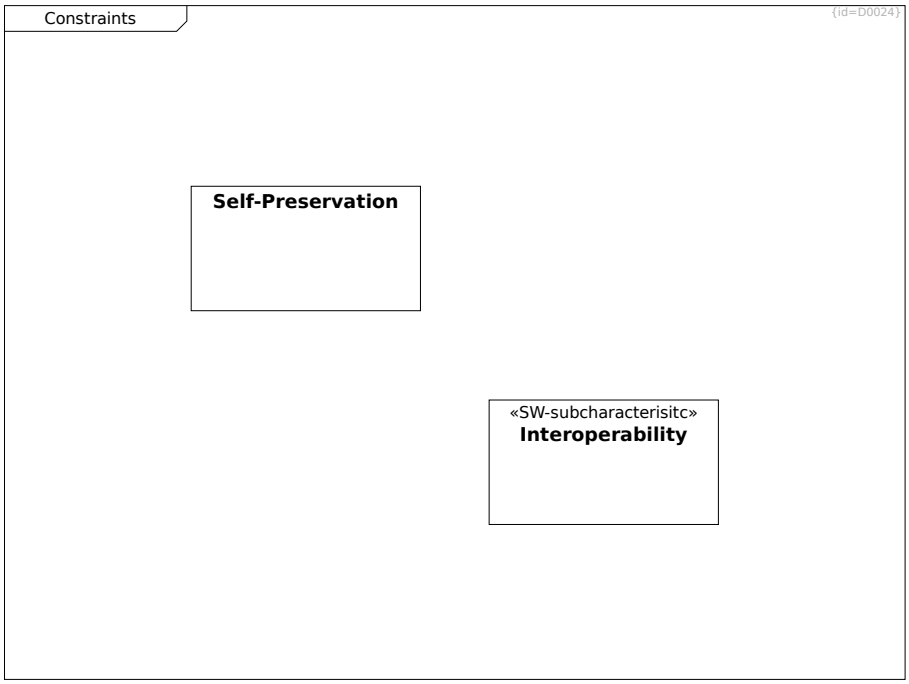
Camera «env-perception» C0041 *(appears in Section 1.5.1: L2 Mechanics, Section 1.5.3.1.1: L3 Requirements, Section 1.5.2: L2 Electric Parts and Electronics)*

-- Main Logic Board R0051

1.5.3.2 Constraints

Constraints D0024

This section explains the major obstacles, that need to be considered when designing a solution to reach the project goals. (Problem Space, Software Level L3)



Self-Preservation C0081 *(appears in Section 1.5.3.2: Constraints, Section 1.5.3.9: Architecture Decisions)*

In case a wookiee growls at the MoD5G, it shall flee for self-preservation

Interoperability «SW-subcharacterisitc» C0123 *(appears in Section 1.5.3.10.1: Quality Tree, Section 1.5.3.2: Constraints, Section 1.5.3.10.2: Quality Scenarios, Section 1.5.3.10: Quality Requirements)*

The programming and charging interfaces of the MoD5G shall be compatible to

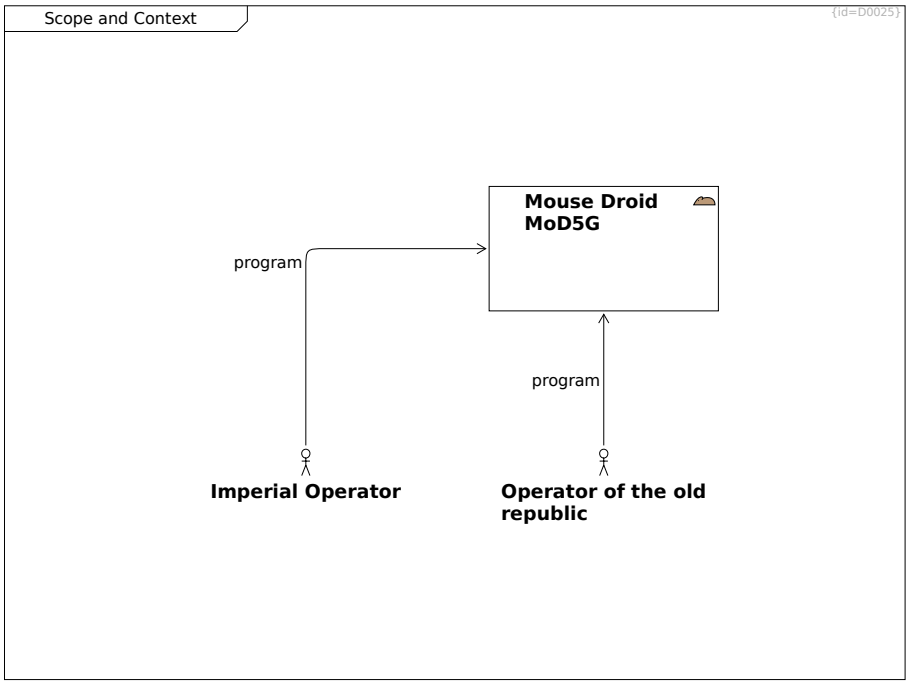
- old republic terminals

- imperial terminals

1.5.3.3 Scope and Context

Scope and Context D0025

This section shows the organizational contexts of development and operational environments. (Problem Space, Software Level L3)



Imperial Operator C0082 *(appears in Section 1.5.3.3: Scope and Context)*

program --> Mouse Droid MoD5G R0113

Operator of the old republic C0083 *(appears in Section 1.5.3.3: Scope and Context)*

program --> Mouse Droid MoD5G R0112

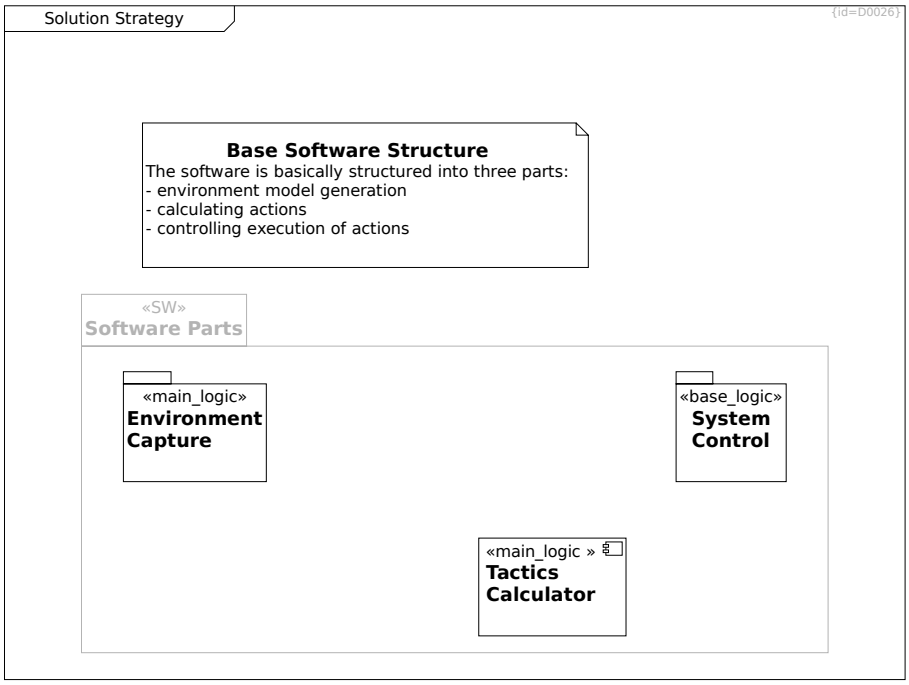
Mouse Droid MoD5G «mouse-droid» C0004 *(appears in Section 1.5.3.1: Requirements and Goals, Section 1.1: Requirements and Goals, Section 1.5.3.3: Scope and Context, Section 1.7: Deployment View)*

The Mouse Droid (MoD5G) is a repair droid that can be instructed to perform a mission and which then autonomously selects tactics to achieve the mission goals.

1.5.3.4 Solution Strategy

Solution Strategy D0026

This section shows the most fundamental principles of the software design. (Solution Space, Software Level L3)



Environment Capture «main_logic» C0118 *(appears in Section 1.5.3.5.1: Environment Capture, Section 1.5.3.4: Solution Strategy, Section 1.5.3.5: Building Block View, Section 1.5.3.7: Deployment View)*

Tactics Calculator «main_logic » C0108 *(appears in Section 1.5.3.6.1: Environment Model Composer Sequence, Section 1.5.3.4: Solution Strategy, Section 1.5.3.5: Building Block View, Section 1.5.3.7: Deployment View)*

Calculate tactics based on given strategy and current situation model

System Control «base_logic» C0119 *(appears in Section 1.5.3.5.2: System Control, Section 1.5.3.4: Solution Strategy, Section 1.5.3.5: Building Block View, Section 1.5.3.7: Deployment View)*

Base Software Structure C0121 *(appears in Section 1.5.3.4: Solution Strategy)*

The software is basically structured into three parts:

- environment model generation

- calculating actions

- controlling execution of actions

Software Parts «SW» C0020 *(appears in Section 1.5: Building Block View, Section 1.5.3.4: Solution Strategy, Section 1.5.3.5: Building Block View)*

The software consists of control logic, initial data that was integrated at the factory and learned data that is aggregated during operation.

These software items are split over two logical boards and subdivided into independent execution partitions.

+-- Tactics Calculator R0236

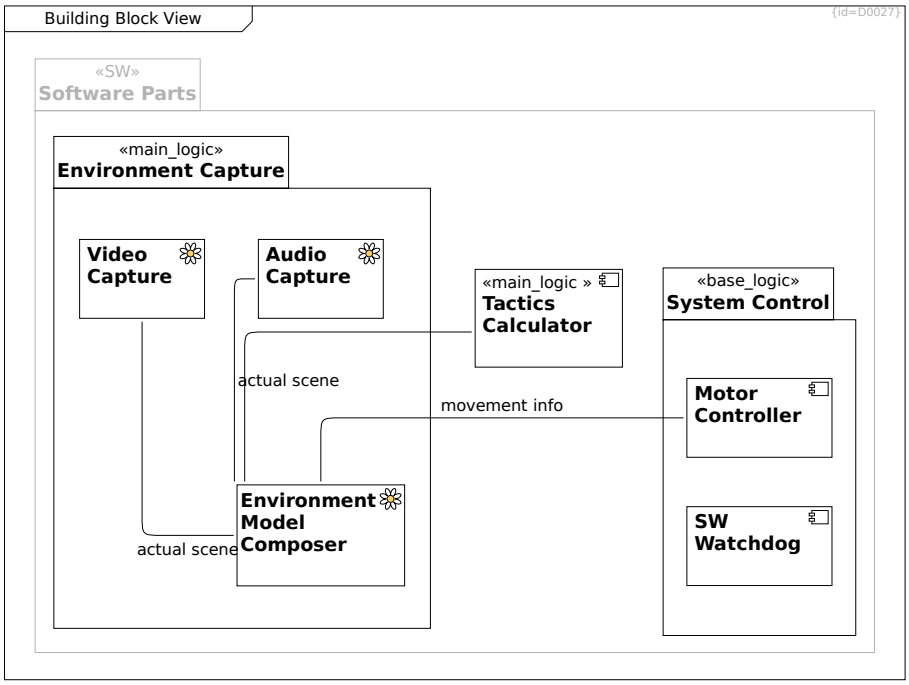
+-- Environment Capture R0241

+-- System Control R0242

1.5.3.5 Building Block View

Building Block View D0027

This section shows the parts of the MoD5G software (Solution Space, Software Level L3)



Video Capture «env-perception» C0106 *(appears in Section 1.5.3.5.1: Environment Capture, Section 1.5.3.6.1: Environment Model Composer Sequence, Section 1.5.3.5: Building Block View, Section 1.5.3.7: Deployment View)*

actual scene -- Environment Model Composer R0155

Audio Capture «env-perception» C0107 *(appears in Section 1.5.3.5.1: Environment Capture, Section 1.5.3.6.1: Environment Model Composer Sequence, Section 1.5.3.5: Building Block View, Section 1.5.3.7: Deployment View)*

actual scene -- Environment Model Composer R0154

Tactics Calculator «main_logic » C0108 *(appears in Section 1.5.3.6.1: Environment Model Composer Sequence, Section 1.5.3.4: Solution Strategy, Section 1.5.3.5: Building Block View, Section 1.5.3.7: Deployment View)*

Calculate tactics based on given strategy and current situation model

Motor Controller C0109 *(appears in Section 1.5.3.6.1: Environment Model Composer Sequence, Section 1.5.3.5.2: System Control, Section 1.5.3.5: Building Block View, Section 1.5.3.7: Deployment View)*

Move motors according to calculated tactics

movement info -- Environment Model Composer R0153

Environment Capture «main_logic» C0118 *(appears in Section 1.5.3.5.1: Environment Capture, Section 1.5.3.4: Solution Strategy, Section 1.5.3.5: Building Block View, Section 1.5.3.7: Deployment View)*

+-- Environment Model Composer R0152

+-- Video Capture R0148

+-- Audio Capture R0149

System Control «base_logic» C0119 *(appears in Section 1.5.3.5.2: System Control, Section 1.5.3.4: Solution Strategy, Section 1.5.3.5: Building Block View, Section 1.5.3.7: Deployment View)*

+-- SW Watchdog R0174

+-- Motor Controller R0151

Environment Model Composer «env-perception» C0120 *(appears in Section 1.5.3.5.1: Environment Capture, Section 1.5.3.6.1: Environment Model Composer Sequence, Section 1.5.3.5: Building Block View, Section 1.5.3.7: Deployment View)*

-- Tactics Calculator R0156

SW Watchdog C0129 *(appears in Section 1.5.3.5.2: System Control, Section 1.5.3.5: Building Block View, Section 1.5.3.8: Crosscutting Concepts)*

The SW Watchdog shall check

- validity of data as well as

- validity of sequence of checkpoints

received from software components on the Main Logic Board.

See also Section 1.5.3.8: Crosscutting Concepts.

Software Parts «SW» C0020 *(appears in Section 1.5: Building Block View, Section 1.5.3.4: Solution Strategy, Section 1.5.3.5: Building Block View)*

The software consists of control logic, initial data that was integrated at the factory and learned data that is aggregated during operation.

These software items are split over two logical boards and subdivided into independent execution partitions.

+-- Tactics Calculator R0236

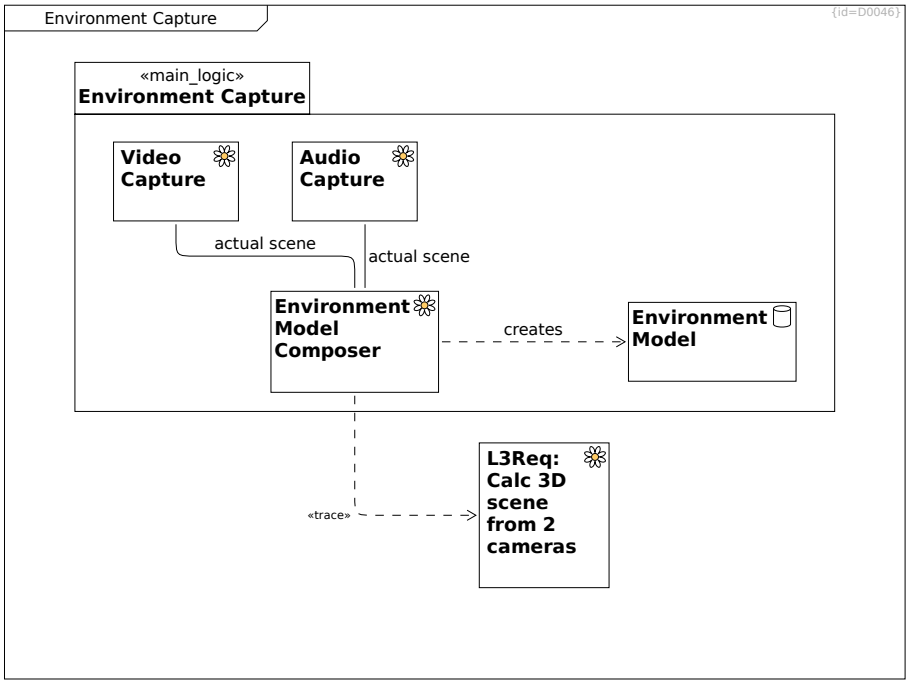
+-- Environment Capture R0241

+-- System Control R0242

1.5.3.5.1 Environment Capture

Environment Capture D0046

This diagram shows the software components that perceive the the outer environment and create a model from this data.



L3Req: Calc 3D scene from 2 cameras «env-perception» C0151 *(appears in Section 1.5.3.1.1: L3 Requirements, Section 1.5.3.5.1: Environment Capture)*

The Main Logic Board shall create a 3D model of the environment based on 2 camera images.

Audio Capture «env-perception» C0107 *(appears in Section 1.5.3.5.1: Environment Capture, Section 1.5.3.6.1: Environment Model Composer Sequence, Section 1.5.3.5: Building Block View, Section 1.5.3.7: Deployment View)*

actual scene -- Environment Model Composer R0154

Environment Capture «main_logic» C0118 *(appears in Section 1.5.3.5.1: Environment Capture, Section 1.5.3.4: Solution Strategy, Section 1.5.3.5: Building Block View, Section 1.5.3.7: Deployment View)*

+-- Environment Model Composer R0152

+-- Video Capture R0148

+-- Audio Capture R0149

+-- Environment Model R0261

Video Capture «env-perception» C0106 *(appears in Section 1.5.3.5.1: Environment Capture, Section 1.5.3.6.1: Environment Model Composer Sequence, Section 1.5.3.5: Building Block View, Section 1.5.3.7: Deployment View)*

actual scene -- Environment Model Composer R0155

Environment Model Composer «env-perception» C0120 *(appears in Section 1.5.3.5.1: Environment Capture, Section 1.5.3.6.1: Environment Model Composer Sequence, Section 1.5.3.5: Building Block View, Section 1.5.3.7: Deployment View)*

·-> L3Req: Calc 3D scene from 2 cameras R0254

creates ·-> Environment Model R0260

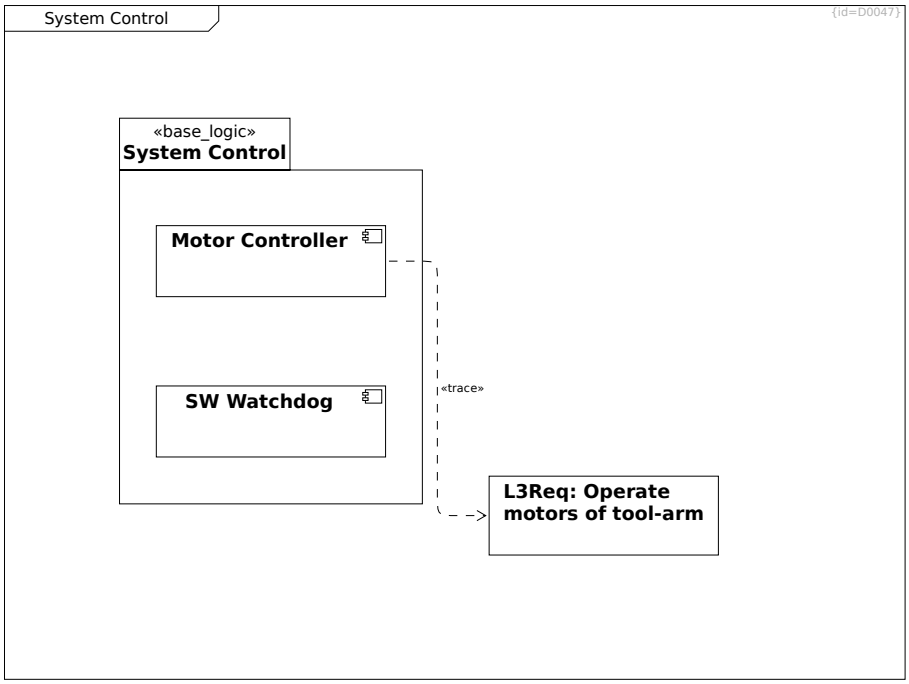
Environment Model «data» C0111 *(appears in Section 1.5.3.5.1: Environment Capture, Section 1.5.3.12: Glossary)*

The enironment model refers to the (limited) knowledge of the software on the real environment.

1.5.3.5.2 System Control

System Control D0047

This diagram shows the software components that perceive MoD5G internal sensor data and steer the actuators of the MoD5G.



L3Req: Operate motors of tool-arm C0152 *(appears in Section 1.5.3.1.1: L3 Requirements, Section 1.5.3.5.2: System Control)*

The Base Logic Board shall steer the motors of the tool-arm to a given 3D position.

System Control «base_logic» C0119 *(appears in Section 1.5.3.5.2: System Control, Section 1.5.3.4: Solution Strategy, Section 1.5.3.5: Building Block View, Section 1.5.3.7: Deployment View)*

+-- SW Watchdog R0174

+-- Motor Controller R0151

Motor Controller C0109 *(appears in Section 1.5.3.6.1: Environment Model Composer Sequence, Section 1.5.3.5.2: System Control, Section 1.5.3.5: Building Block View, Section 1.5.3.7: Deployment View)*

Move motors according to calculated tactics

·-> L3Req: Operate motors of tool-arm R0253

SW Watchdog C0129 *(appears in Section 1.5.3.5.2: System Control, Section 1.5.3.5: Building Block View, Section 1.5.3.8: Crosscutting Concepts)*

The SW Watchdog shall check

- validity of data as well as

- validity of sequence of checkpoints

received from software components on the Main Logic Board.

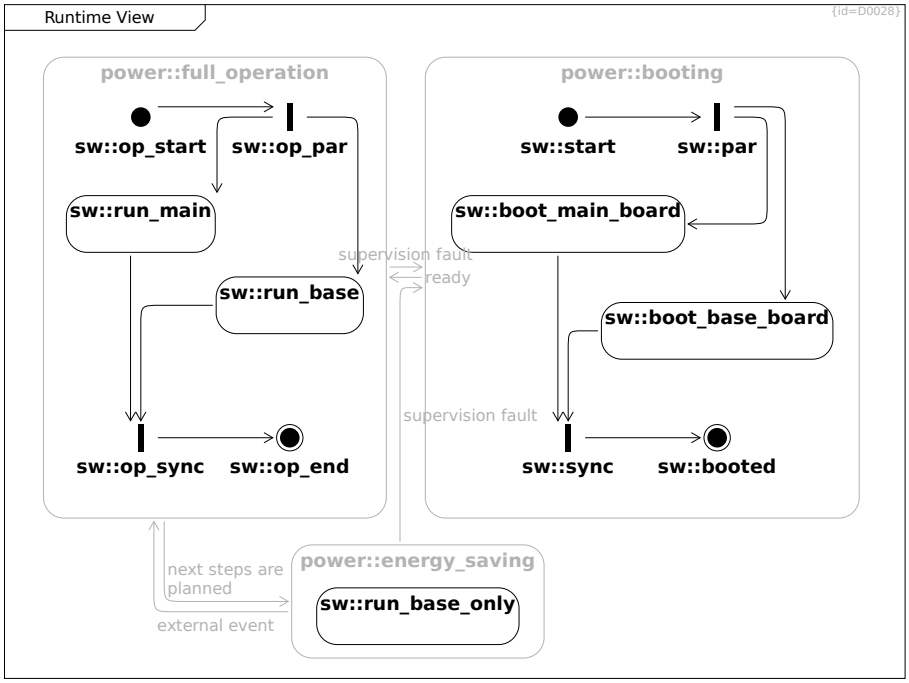
See also Section 1.5.3.8: Crosscutting Concepts.

1.5.3.6 Runtime View

Runtime View D0028

This section shows the dynamic behavior of the software (Solution Space, Software Level L3)

This diagram shows the software states embedded in the system states. See Section 1.6.1: Power Modes.



power::booting C0024 *(appears in Section 1.6.1: Power Modes, Section 1.6.1.1: Power Mode Timings, Section 1.5.3.6: Runtime View)*

+-- sw::sync R0107

+-- sw::start R0099

+-- sw::par R0100

+-- sw::boot_main_board R0098

+-- sw::booted R0110

+-- sw::boot_base_board R0097

ready --> power::full_operation R0016

power::full_operation C0025 *(appears in Section 1.6.1: Power Modes, Section 1.6.1.1: Power Mode Timings, Section 1.5.3.6: Runtime View)*

While the MoD5G is in full_operation state, all software parts are running and able to react on input data.

+-- sw::op_sync R0117

+-- sw::op_end R0115

+-- sw::op_par R0116

+-- sw::op_start R0114

+-- sw::run_base R0105

+-- sw::run_main R0104

next steps are planned --> power::energy_saving R0017

mission tactics are planned, no need to adapt

supervision fault --> power::booting R0033

power::energy_saving C0026 *(appears in Section 1.6.1: Power Modes, Section 1.6.1.1: Power Mode Timings, Section 1.5.3.6: Runtime View)*

+-- sw::run_base_only R0106

external event --> power::full_operation R0018

external event causes re-evaluating tactics

supervision fault --> power::booting R0032

sw::boot_base_board C0072 *(appears in Section 1.5.3.6: Runtime View)*

--> sw::sync R0109

sw::boot_main_board C0073 *(appears in Section 1.5.3.6: Runtime View)*

--> sw::sync R0108

sw::start C0074 *(appears in Section 1.5.3.6: Runtime View)*

--> sw::par R0101

sw::par C0075 *(appears in Section 1.5.3.6: Runtime View)*

--> sw::boot_main_board R0102

--> sw::boot_base_board R0103

sw::run_main C0076 *(appears in Section 1.5.3.6: Runtime View)*

--> sw::op_sync R0123

sw::run_base C0077 *(appears in Section 1.5.3.6: Runtime View)*

--> sw::op_sync R0122

sw::run_base_only C0078 *(appears in Section 1.5.3.6: Runtime View)*

sw::sync C0079 *(appears in Section 1.5.3.6: Runtime View)*

--> sw::booted R0111

sw::booted C0080 *(appears in Section 1.5.3.6: Runtime View)*

sw::op_start C0084 *(appears in Section 1.5.3.6: Runtime View)*

--> sw::op_par R0119

sw::op_end C0085 *(appears in Section 1.5.3.6: Runtime View)*

sw::op_par C0086 *(appears in Section 1.5.3.6: Runtime View)*

--> sw::run_main R0120

--> sw::run_base R0121

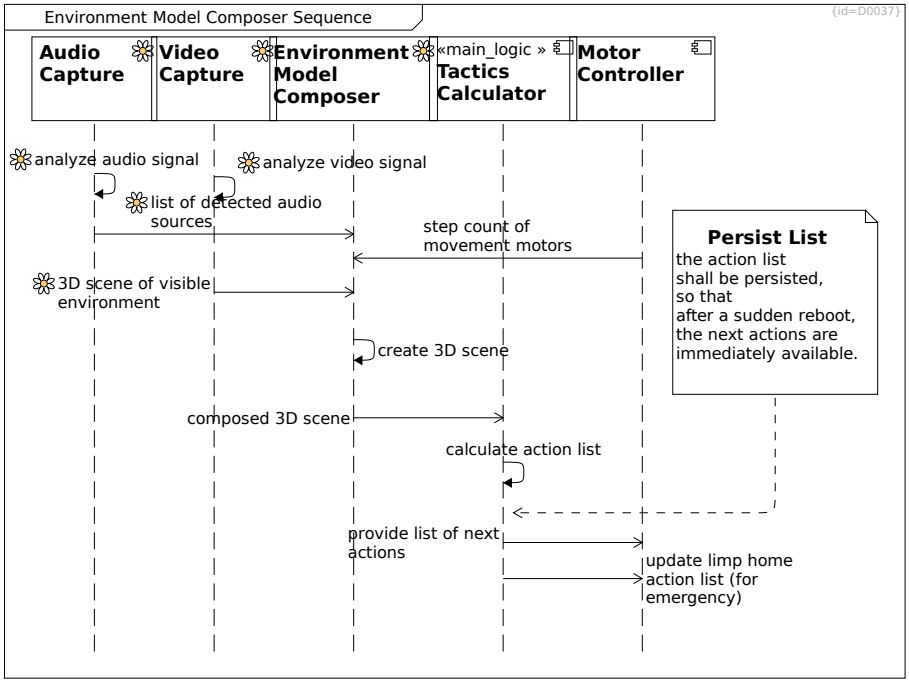
sw::op_sync C0087 *(appears in Section 1.5.3.6: Runtime View)*

--> sw::op_end R0118

1.5.3.6.1 Environment Model Composer Sequence

Environment Model Composer Sequence D0037

This diagram shows the typical communication sequence to compose the environment model.



Motor Controller C0109 *(appears in Section 1.5.3.6.1: Environment Model Composer Sequence, Section 1.5.3.5.2: System Control, Section 1.5.3.5: Building Block View, Section 1.5.3.7: Deployment View)*

Move motors according to calculated tactics

step count of movement motors --> Environment Model Composer R0166

step count of steering and movement motors

Tactics Calculator «main_logic » C0108 *(appears in Section 1.5.3.6.1: Environment Model Composer Sequence, Section 1.5.3.4: Solution Strategy, Section 1.5.3.5: Building Block View, Section 1.5.3.7: Deployment View)*

Calculate tactics based on given strategy and current situation model

calculate action list -->> Tactics Calculator R0172

calculate action list to follow the given strategy

provide list of next actions --> Motor Controller R0167

update limp home action list (for emergency) --> Motor Controller R0168

For the emergency case, update the limp home action list

Environment Model Composer «env-perception» C0120 *(appears in Section 1.5.3.5.1: Environment Capture, Section 1.5.3.6.1: Environment Model Composer Sequence, Section 1.5.3.5: Building Block View, Section 1.5.3.7: Deployment View)*

create 3D scene -->> Environment Model Composer R0171

create 3D scene based on sensors, status and history.

composed 3D scene --> Tactics Calculator R0165

Audio Capture «env-perception» C0107 *(appears in Section 1.5.3.5.1: Environment Capture, Section 1.5.3.6.1: Environment Model Composer Sequence, Section 1.5.3.5: Building Block View, Section 1.5.3.7: Deployment View)*

analyze audio signal «env-perception» -->> Audio Capture R0169

list of detected audio sources «env-perception» --> Environment Model Composer R0164

Video Capture «env-perception» C0106 *(appears in Section 1.5.3.5.1: Environment Capture, Section 1.5.3.6.1: Environment Model Composer Sequence, Section 1.5.3.5: Building Block View, Section 1.5.3.7: Deployment View)*

analyze video signal «env-perception» -->> Video Capture R0170

3D scene of visible environment «env-perception» --> Environment Model Composer R0163

Persist List C0142 *(appears in Section 1.5.3.6.1: Environment Model Composer Sequence)*

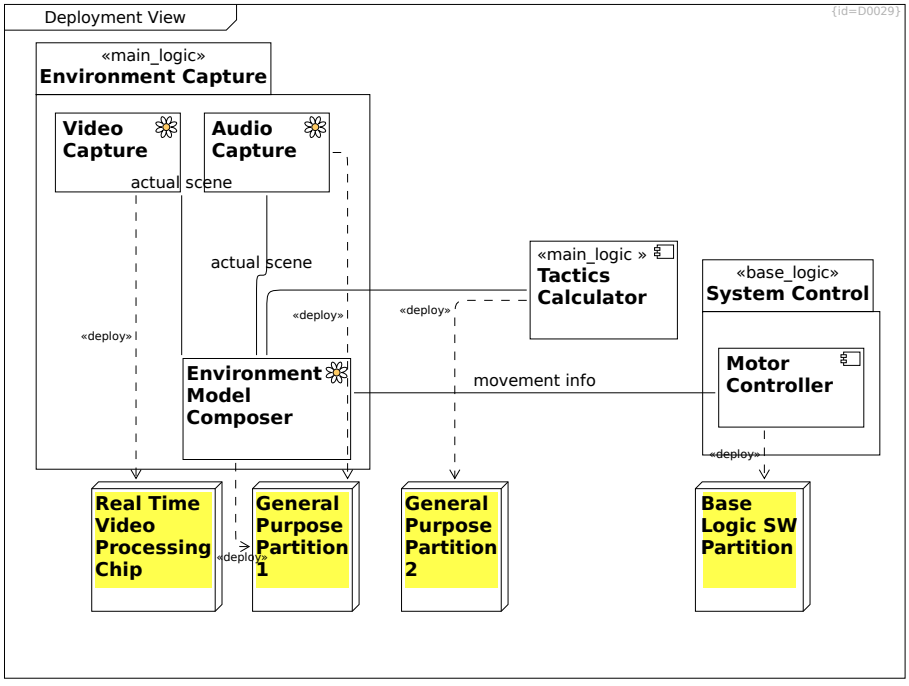
the action list shall be persisted, so that after a sudden reboot, the next actions are immediately available.

·-> Tactics Calculator R0220

1.5.3.7 Deployment View

Deployment View D0029

This section shows the deployment of the solution into the environment. (Solution Space, Software Level L3)



Motor Controller C0109 *(appears in Section 1.5.3.6.1: Environment Model Composer Sequence, Section 1.5.3.5.2: System Control, Section 1.5.3.5: Building Block View, Section 1.5.3.7: Deployment View)*

Move motors according to calculated tactics

movement info -- Environment Model Composer R0153

·-> Base Logic SW Partition R0200

Base Logic SW Partition C0064 *(appears in Section 1.5.3.7: Deployment View, Section 1.5.3: L2 Software [subsystem])*

General Purpose Partition 2 C0063 *(appears in Section 1.5.3.7: Deployment View, Section 1.5.3: L2 Software [subsystem])*

Tactics Calculator «main_logic » C0108 *(appears in Section 1.5.3.6.1: Environment Model Composer Sequence, Section 1.5.3.4: Solution Strategy, Section 1.5.3.5: Building Block View, Section 1.5.3.7: Deployment View)*

Calculate tactics based on given strategy and current situation model

·-> General Purpose Partition 2 R0202

Audio Capture «env-perception» C0107 *(appears in Section 1.5.3.5.1: Environment Capture, Section 1.5.3.6.1: Environment Model Composer Sequence, Section 1.5.3.5: Building Block View, Section 1.5.3.7: Deployment View)*

actual scene -- Environment Model Composer R0154

·-> General Purpose Partition 1 R0204

General Purpose Partition 1 C0062 *(appears in Section 1.5.3.7: Deployment View, Section 1.5.3: L2 Software [subsystem])*

Real Time Video Processing Chip C0060 *(appears in Section 1.5.3.7: Deployment View, Section 1.5.3: L2 Software [subsystem])*

Video Capture «env-perception» C0106 *(appears in Section 1.5.3.5.1: Environment Capture, Section 1.5.3.6.1: Environment Model Composer Sequence, Section 1.5.3.5: Building Block View, Section 1.5.3.7: Deployment View)*

actual scene -- Environment Model Composer R0155

·-> Real Time Video Processing Chip R0201

Environment Model Composer «env-perception» C0120 *(appears in Section 1.5.3.5.1: Environment Capture, Section 1.5.3.6.1: Environment Model Composer Sequence, Section 1.5.3.5: Building Block View, Section 1.5.3.7: Deployment View)*

-- Tactics Calculator R0156

·-> General Purpose Partition 1 R0203

Environment Capture «main_logic» C0118 *(appears in Section 1.5.3.5.1: Environment Capture, Section 1.5.3.4: Solution Strategy, Section 1.5.3.5: Building Block View, Section 1.5.3.7: Deployment View)*

+-- Environment Model Composer R0152

+-- Video Capture R0148

+-- Audio Capture R0149

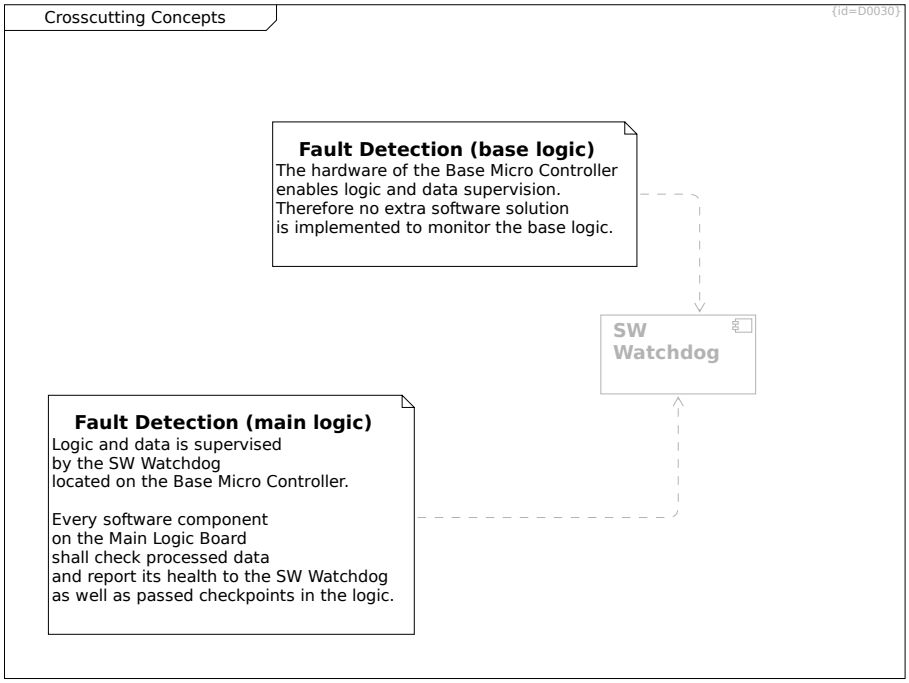
System Control «base_logic» C0119 *(appears in Section 1.5.3.5.2: System Control, Section 1.5.3.4: Solution Strategy, Section 1.5.3.5: Building Block View, Section 1.5.3.7: Deployment View)*

+-- Motor Controller R0151

1.5.3.8 Crosscutting Concepts

Crosscutting Concepts D0030

This section shows the recurring concepts within the the designed solution. (Solution Space, Software Level L3)



Fault Detection (main logic) C0127 *(appears in Section 1.5.3.8: Crosscutting Concepts, Section 1.5.3.11: Risks and Technical Debts)*

Logic and data is supervised by the SW Watchdog located on the Base Micro Controller.

Every software component on the Main Logic Board shall check processed data and report its health to the SW Watchdog as well as passed checkpoints in the logic.

·-> SW Watchdog R0175

Fault Detection (base logic) C0128 *(appears in Section 1.5.3.8: Crosscutting Concepts)*

The hardware of the Base Micro Controller enables logic and data supervision. Therefore no extra software solution is implemented to monitor the base logic.

·-> SW Watchdog R0198

SW Watchdog C0129 *(appears in Section 1.5.3.5.2: System Control, Section 1.5.3.5: Building Block View, Section 1.5.3.8: Crosscutting Concepts)*

The SW Watchdog shall check

- validity of data as well as

- validity of sequence of checkpoints

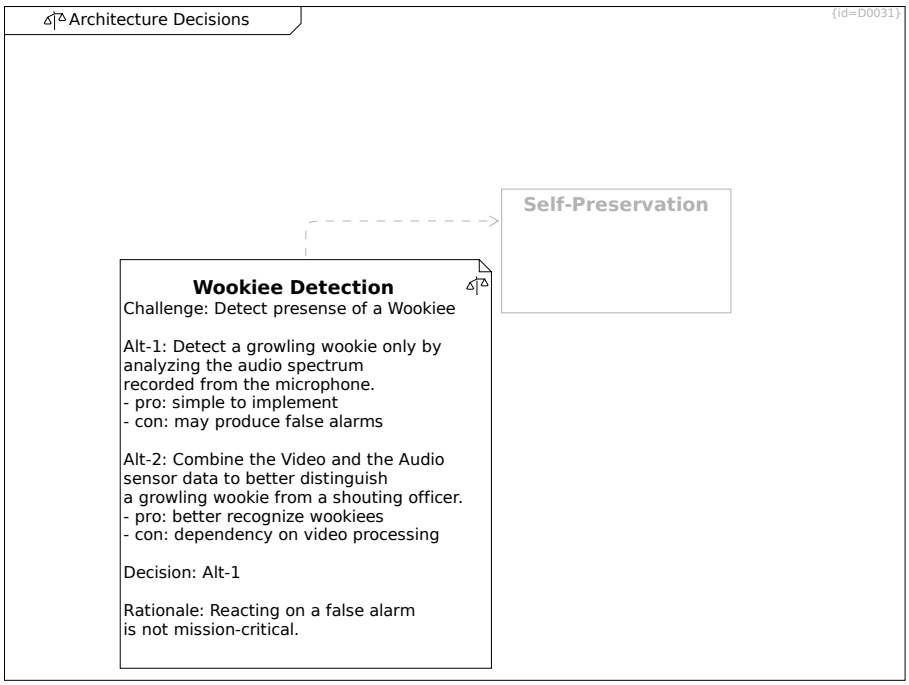
received from software components on the Main Logic Board.

See also Section 1.5.3.8: Crosscutting Concepts.

1.5.3.9 Architecture Decisions

Architecture Decisions «decision» D0031

This section documents the major design decisions. (Solution Space, Software Level L3)



Wookiee Detection «decision» C0131 *(appears in Section 1.5.3.9: Architecture Decisions)*

Challenge: Detect presense of a Wookiee

Alt-1: Detect a growling wookie only by analyzing the audio spectrum recorded from the microphone.

- pro: simple to implement

- con: may produce false alarms

Alt-2: Combine the Video and the Audio sensor data to better distinguish a growling wookie from a shouting officer.

- pro: better recognize wookiees

- con: dependency on video processing

Decision: Alt-1

Rationale: Reacting on a false alarm is not mission-critical.

·-> Self-Preservation R0177

Self-Preservation C0081 *(appears in Section 1.5.3.2: Constraints, Section 1.5.3.9: Architecture Decisions)*

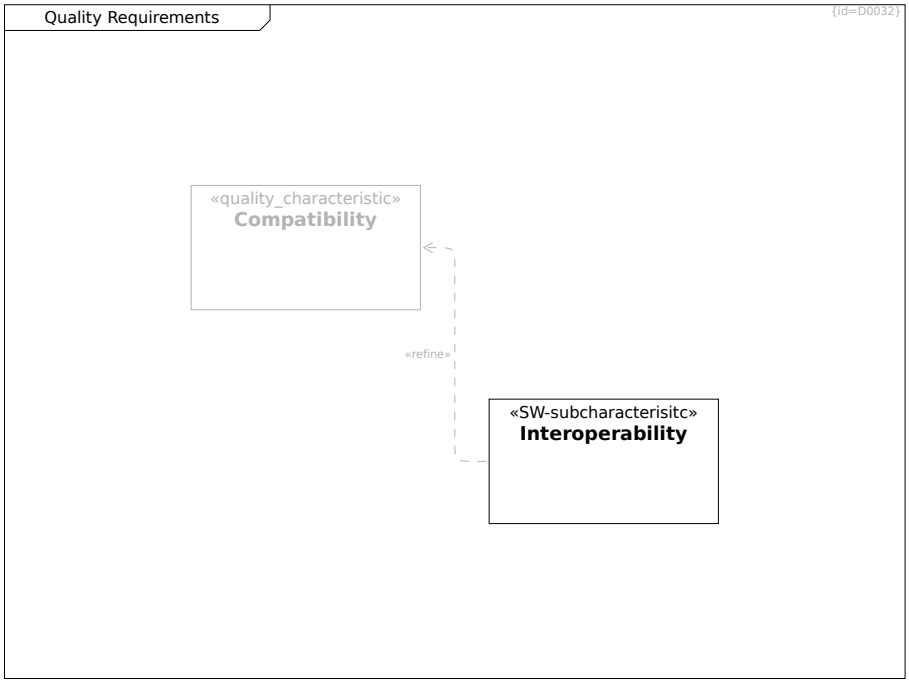
In case a wookiee growls at the MoD5G, it shall flee for self-preservation

1.5.3.10 Quality Requirements

Quality Requirements D0032

This section shows the major quality scenarios. (Problem Space, Software Level L3)

Similar to Section 1.10: Quality Requirements for system level L1, this section shows quality expectations: The WHAT shall be implemented, not the HOW.



Compatibility «quality_characteristic» C0122 *(appears in Section 1.5.3.10.1: Quality Tree, Section 1.10: Quality Requirements, Section 1.5.3.10: Quality Requirements)*

Compatibility defines a set of attributes that measures how well data and messages can be exchanged with other programs and/or versions.

Interoperability «SW-subcharacterisitc» C0123 *(appears in Section 1.5.3.10.1: Quality Tree, Section 1.5.3.2: Constraints, Section 1.5.3.10.2: Quality Scenarios, Section 1.5.3.10: Quality Requirements)*

The programming and charging interfaces of the MoD5G shall be compatible to

- old republic terminals

- imperial terminals

·-> Compatibility R0158

1.5.3.10.1 Quality Tree

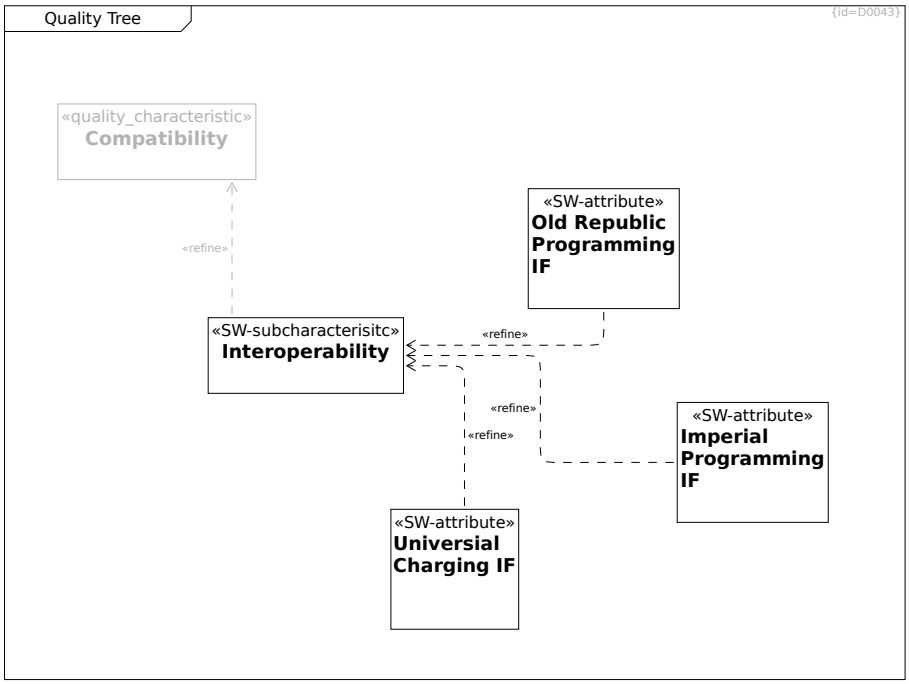
Quality Tree D0043

When specifying quality requirements, these are categorized by two dimensions:

1) Which characteristic do they affect

2) Which use case do they serve

Thich chapter lists the quality requirements ordered by their main characteristic.



Old Republic Programming IF «SW-attribute» C0124 *(appears in Section 1.5.3.10.1: Quality Tree)*

The old republic protocol for programming a droid shall be supported.

··> Interoperability R0159

Imperial Programming IF «SW-attribute» C0125 *(appears in Section 1.5.3.10.1: Quality Tree)*

The imperial protocol for programming a droid shall be supported.

··> Interoperability R0160

Universial Charging IF «SW-attribute» C0126 *(appears in Section 1.5.3.10.1: Quality Tree)*

The intergalactic standard protocol for power charging shall be supported.

··> Interoperability R0161

Interoperability «SW-subcharacterisitc» C0123 *(appears in Section 1.5.3.10.1: Quality Tree, Section 1.5.3.2: Constraints, Section 1.5.3.10.2: Quality Scenarios, Section 1.5.3.10: Quality Requirements)*

The programming and charging interfaces of the MoD5G shall be compatible to

- old republic terminals
- imperial terminals

·-> Compatibility R0158

Compatibility «quality_characteristic» C0122 *(appears in Section 1.5.3.10.1: Quality Tree, Section 1.10: Quality Requirements, Section 1.5.3.10: Quality Requirements)*

Compatibility defines a set of attributes that measures how well data and messages can be exchanged with other programs and/or versions.

1.5.3.10.2 Quality Scenarios

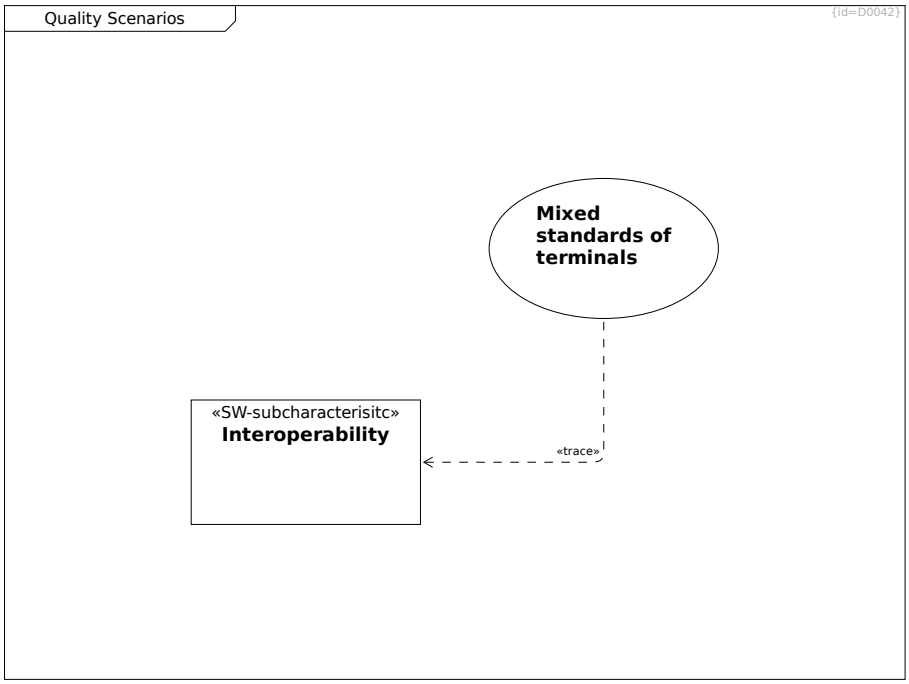
Quality Scenarios D0042

When specifying quality requirements, these are categorized by two dimensions:

1) Which characteristic do they affect

2) Which use case do they serve

Thich chapter lists the use cases which have special importance for quality requirements.



Mixed standards of terminals C0145 *(appears in Section 1.5.3.10.2: Quality Scenarios)*

precondition:

- The MoD5G operates in an environment providing mixed terminal standards

trigger:

- The MoD5G drives to a charging or programming terminal which complied to either old republic or imperial standard.

scenario:

- The MoD5G determines the applicable standard
- The MoD5G uses the terminal for programming or charging

·-> Interoperability R0227

Interoperability «SW-subcharacterisitc» C0123 *(appears in Section 1.5.3.10.1: Quality Tree, Section 1.5.3.2: Constraints, Section 1.5.3.10.2: Quality Scenarios, Section 1.5.3.10: Quality Requirements)*

The programming and charging interfaces of the MoD5G shall be compatible to

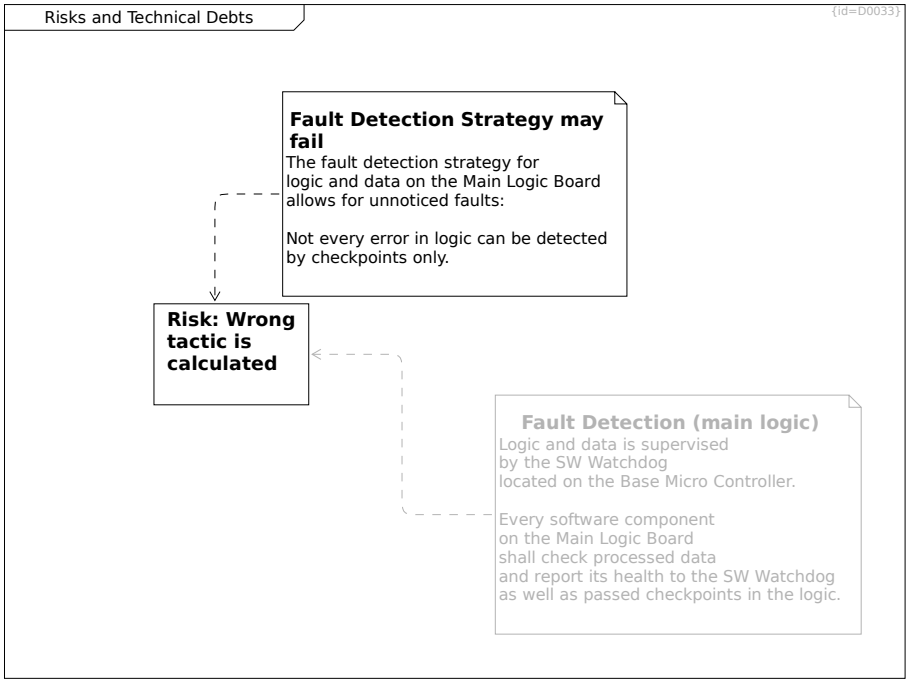
- old republic terminals

- imperial terminals

1.5.3.11 Risks and Technical Debts

Risks and Technical Debts D0033

This section lists the risks and not-yet-addressed requirements. (Solution Space, Software Level L3)



Fault Detection Strategy may fail C0130 *(appears in Section 1.5.3.11: Risks and Technical Debts)*

The fault detection strategy for logic and data on the Main Logic Board allows for unnoticed faults:

Not every error in logic can be detected by checkpoints only.

··> Risk: Wrong tactic is calculated R0181

Fault Detection (main logic) C0127 *(appears in Section 1.5.3.8: Crosscutting Concepts, Section 1.5.3.11: Risks and Technical Debts)*

Logic and data is supervised by the SW Watchdog located on the Base Micro Controller.

Every software component on the Main Logic Board shall check processed data and report its health to the SW Watchdog as well as passed checkpoints in the logic.

··> Risk: Wrong tactic is calculated R0199

Risk: Wrong tactic is calculated C0134 *(appears in Section 1.5.3.11: Risks and Technical Debts)*

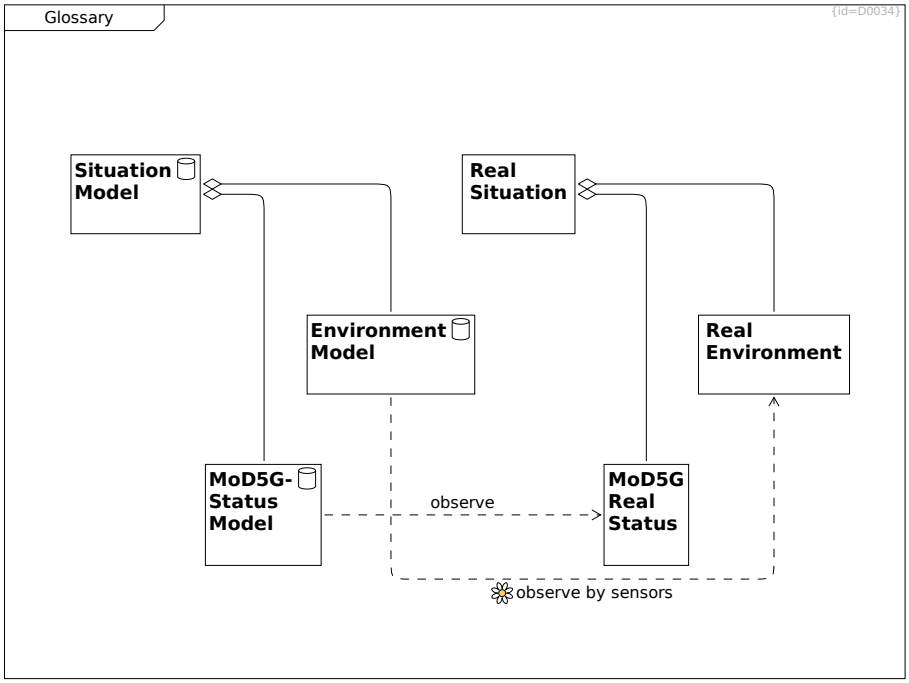
- cause/fault: Due to cosmic rays, the main logic board performs a miscalculation that goes unnoticed by control flow supervision

- risk/failure: the MoD5G calculates a tactic that results in falling off a cliff

1.5.3.12 Glossary

Glossary D0034

This section explains the used terms. (Domain and Solution Space, Software Level L3)



Situation Model «data» C0110 *(appears in Section 1.5.3.12: Glossary)*

The situation model refers to the (limited/erroneous) knowledge of the software on environment and status.

o-- MoD5G-Status Model R0142

o-- Environment Model R0143

Environment Model «data» C0111 *(appears in Section 1.5.3.5.1: Environment Capture, Section 1.5.3.12: Glossary)*

The enironment model refers to the (limited) knowledge of the software on the real environment.

observe by sensors «env-perception» ·-> Real Environment R0146

Sensor data is the basis for assuming an environment model.

MoD5G-Status Model «data» C0112 *(appears in Section 1.5.3.12: Glossary)*

The status model refers to the (limited) knowledge of the software on the real status.

observe ·-> MoD5G Real Status R0147

Sensor data is the basis for assuming a status model. The algorithm for deriving a status model shall take into account that a sensor may be defect and/or a measured value may indicate a defect (which again may have several causes).

Real Situation C0113 *(appears in Section 1.5.3.12: Glossary)*

The real situation refers to the reality of system status and environment.

o-- MoD5G Real Status R0144

o-- Real Environment R0145

Real Environment C0114 *(appears in Section 1.5.3.12: Glossary)*

The real environment refers to the physical environment of the system.

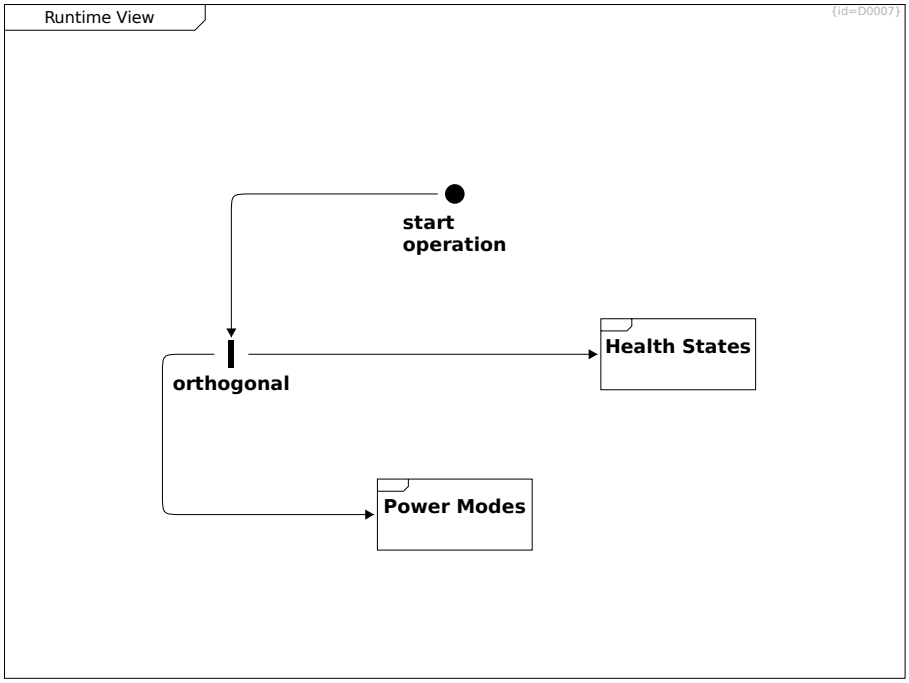
MoD5G Real Status C0115 *(appears in Section 1.5.3.12: Glossary)*

The real status refers to the real system status of the MoD5G. This may differ from what the sensors report.

1.6 Runtime View

Runtime View D0007

This section shows the dynamic behavior of the system (Solution Space, System Level L1)



Power Modes C0021 *(appears in Section 1.6: Runtime View)*

refers to Section 1.6.1: Power Modes

Health States C0022 *(appears in Section 1.6: Runtime View)*

refers to Section 1.6.2: Health States

start operation C0153 *(appears in Section 1.6: Runtime View)*

-->> orthogonal R0262

orthogonal C0154 *(appears in Section 1.6: Runtime View)*

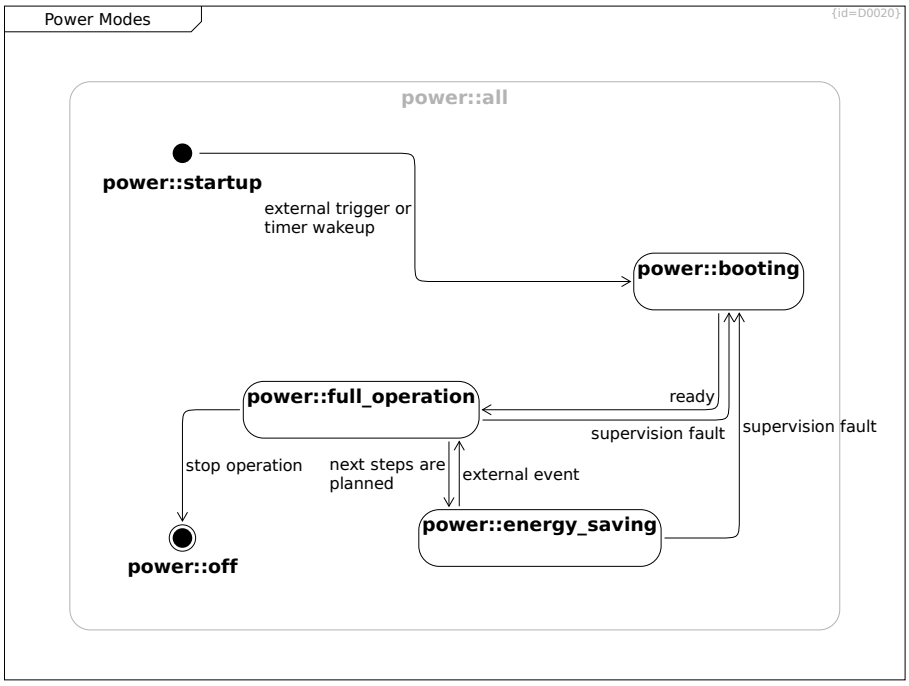
-->> Health States R0263

-->> Power Modes R0264

1.6.1 Power Modes

Power Modes D0020

This diagram shows the power states that are globally valid to all parts of the system.



power::startup C0023 *(appears in Section 1.6.1: Power Modes, Section 1.6.1.1: Power Mode Timings)*

external trigger or timer wakeup --> power::booting R0015

power::booting C0024 *(appears in Section 1.6.1: Power Modes, Section 1.6.1.1: Power Mode Timings, Section 1.5.3.6: Runtime View)*

ready --> power::full_operation R0016

power::full_operation C0025 *(appears in Section 1.6.1: Power Modes, Section 1.6.1.1: Power Mode Timings, Section 1.5.3.6: Runtime View)*

While the MoD5G is in full_operation state, all software parts are running and able to react on input data.

next steps are planned --> power::energy_saving R0017

mission tactics are planned, no need to adapt

stop operation --> power::off R0019

stop operation, set wakeup time

supervision fault --> power::booting R0033

power::energy_saving C0026 *(appears in Section 1.6.1: Power Modes, Section 1.6.1.1: Power Mode Timings, Section 1.5.3.6: Runtime View)*

external event --> power::full_operation R0018

external event causes re-evaluating tactics

supervision fault --> power::booting R0032

power::off C0027 *(appears in Section 1.6.1: Power Modes, Section 1.6.1.1: Power Mode Timings)*

power::all C0138 *(appears in Section 1.6.1: Power Modes)*

The statemachine of all power states

+-- power::booting R0192

+-- power::energy_saving R0193

+-- power::full_operation R0194

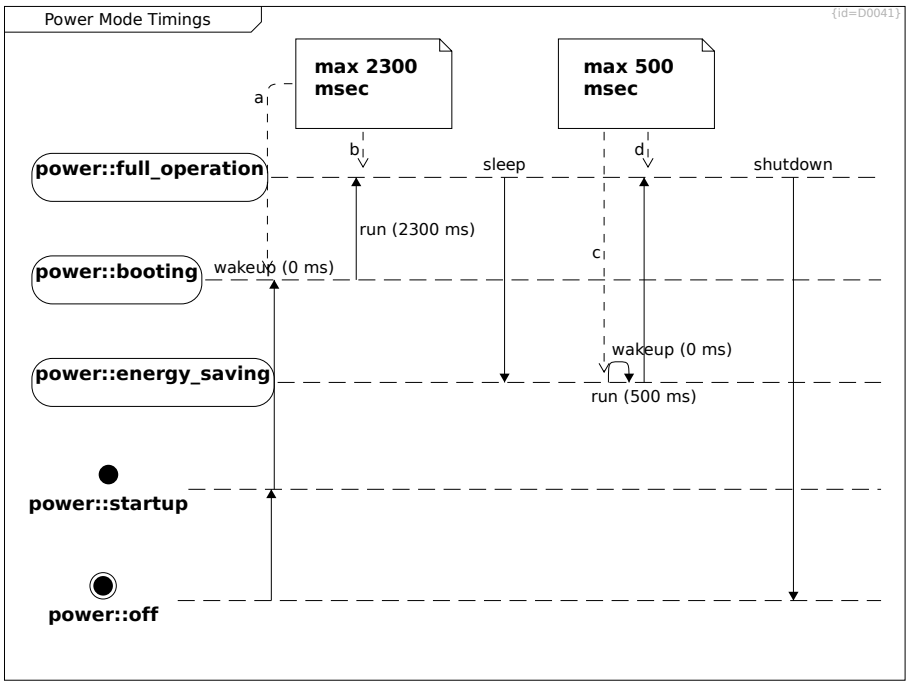
+-- power::startup R0195

+-- power::off R0196

1.6.1.1 Power Mode Timings

Power Mode Timings D0041

This diagram shows the expected startup and shutdown timings.



power::off C0027 *(appears in Section 1.6.1: Power Modes, Section 1.6.1.1: Power Mode Timings)*

-->> power::startup R0209

power::full_operation C0025 *(appears in Section 1.6.1: Power Modes, Section 1.6.1.1: Power Mode Timings, Section 1.5.3.6: Runtime View)*

While the MoD5G is in full_operation state, all software parts are running and able to react on input data.

sleep -->> power::energy_saving R0212

shutdown -->> power::off R0215

power::startup C0023 *(appears in Section 1.6.1: Power Modes, Section 1.6.1.1: Power Mode Timings)*

wakeup (0 ms) -->> power::booting R0210

power::booting C0024 *(appears in Section 1.6.1: Power Modes, Section 1.6.1.1: Power Mode Timings, Section 1.5.3.6: Runtime View)*

run (2300 ms) -->> power::full_operation R0211

power::energy_saving C0026 *(appears in Section 1.6.1: Power Modes, Section 1.6.1.1: Power Mode Timings, Section 1.5.3.6: Runtime View)*

wakeup (0 ms) -->> power::energy_saving R0213

run (500 ms) -->> power::full_operation R0214

max 2300 msec C0139 *(appears in Section 1.6.1.1: Power Mode Timings)*

a ·-> power::booting R0218

b ·-> power::full_operation R0216

max 500 msec C0140 *(appears in Section 1.6.1.1: Power Mode Timings)*

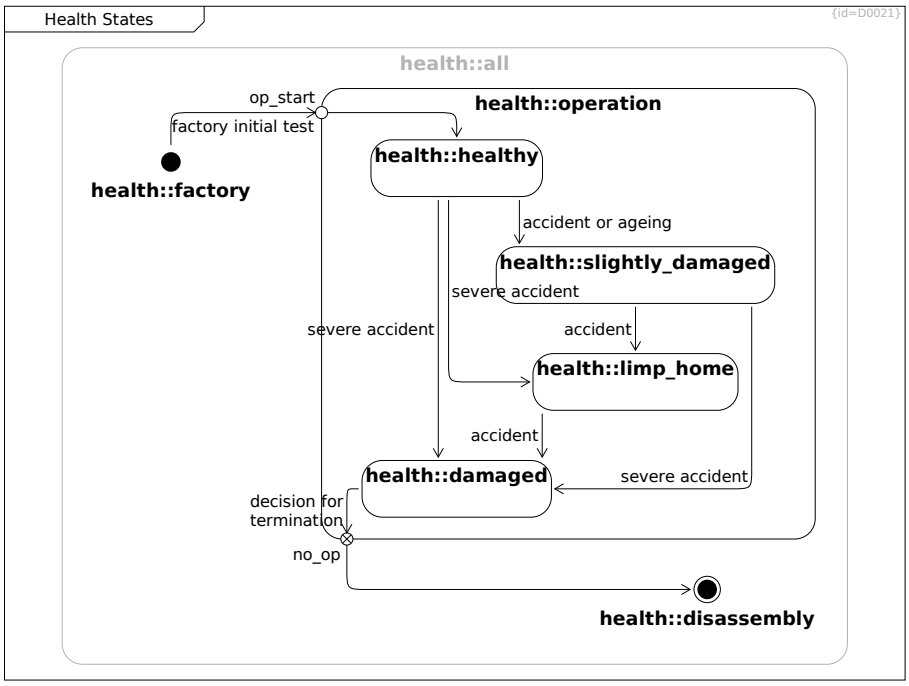
c ·-> power::energy_saving R0219

d ·-> power::full_operation R0217

1.6.2 Health States

Health States D0021

This diagram shows the health states of the MoD5G system.



health::factory C0028 *(appears in Section 1.6.2: Health States)*

factory initial test --> health::operation R0207

health::disassembly C0029 *(appears in Section 1.6.2: Health States)*

health::healthy C0030 *(appears in Section 1.6.2: Health States)*

accident or ageing --> health::slightly_damaged R0021

severe accident --> health::damaged R0029

severe accident --> health::limp_home R0030

health::slightly_damaged C0031 *(appears in Section 1.6.2: Health States)*

accident --> health::limp_home R0022

severe accident --> health::damaged R0031

health::limp_home C0032 *(appears in Section 1.6.2: Health States, Section 1.11: Risks and Technical Debts)*

In case the full operation is not possible anymore, the MoD5G shall drive back to the home charging station.

accident --> health::damaged R0023

health::damaged C0033 *(appears in Section 1.6.2: Health States)*

decision for termination --> health::operation R0206

health::operation C0034 *(appears in Section 1.6.2: Health States)*

op_start F0019

no_op F0018

+-- health::limp_home R0024

+-- health::slightly_damaged R0025

+-- health::healthy R0026

+-- health::damaged R0027

--> health::disassembly R0205

--> health::healthy R0208

health::all C0137 *(appears in Section 1.6.2: Health States)*

The statemachine of all health states

+-- health::factory R0189

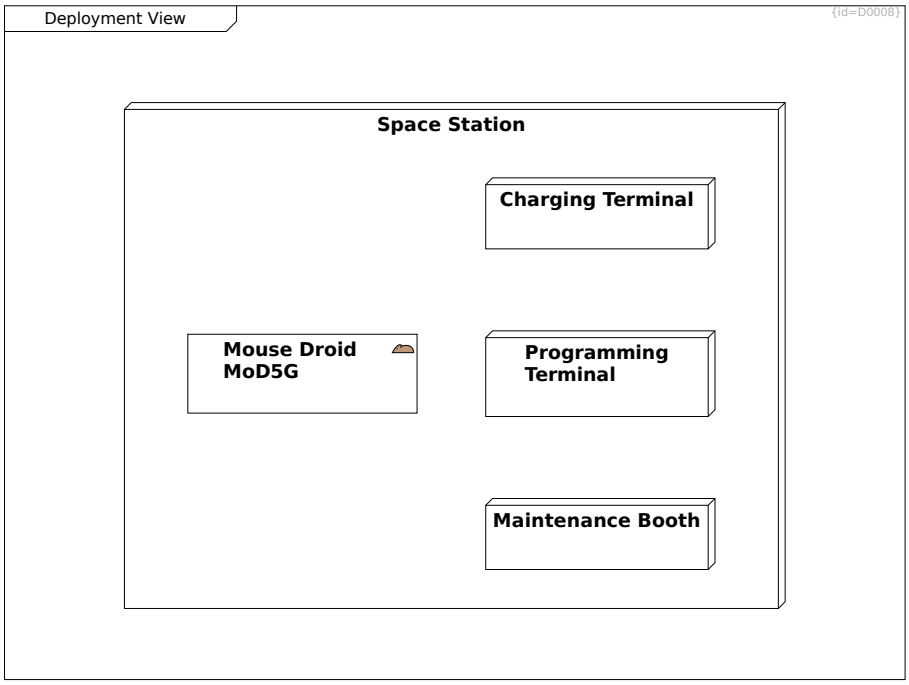
+-- health::operation R0190

+-- health::disassembly R0191

1.7 Deployment View

Deployment View D0008

This section shows the deployment of the solution into the environment. (Solution Space, System Level L1)



Space Station C0095 *(appears in Section 1.7: Deployment View)*

+-- Programming Terminal R0129

+-- Charging Terminal R0130

+-- Maintenance Booth R0131

+-- Mouse Droid MoD5G R0128

Mouse Droid MoD5G «mouse-droid» C0004 *(appears in Section 1.5.3.1: Requirements and Goals, Section 1.1: Requirements and Goals, Section 1.5.3.3: Scope and Context, Section 1.7: Deployment View)*

The Mouse Droid (MoD5G) is a repair droid that can be instructed to perform a mission and which then autonomously selects tactics to achieve the mission goals.

Programming Terminal C0096 *(appears in Section 1.7: Deployment View)*

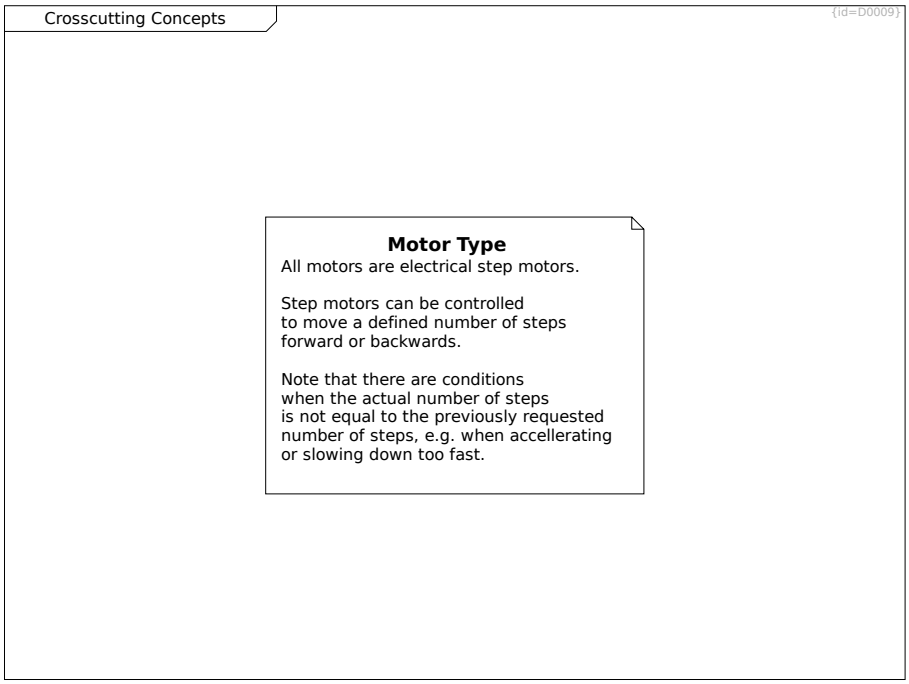
Charging Terminal C0097 *(appears in Section 1.7: Deployment View)*

Maintenance Booth C0098 *(appears in Section 1.7: Deployment View)*

1.8 Crosscutting Concepts

Crosscutting Concepts D0009

This section shows the recurring concepts within the the designed solution. (Solution Space, System Level L1)



Motor Type C0099 *(appears in Section 1.8: Crosscutting Concepts)*

All motors are electrical step motors.

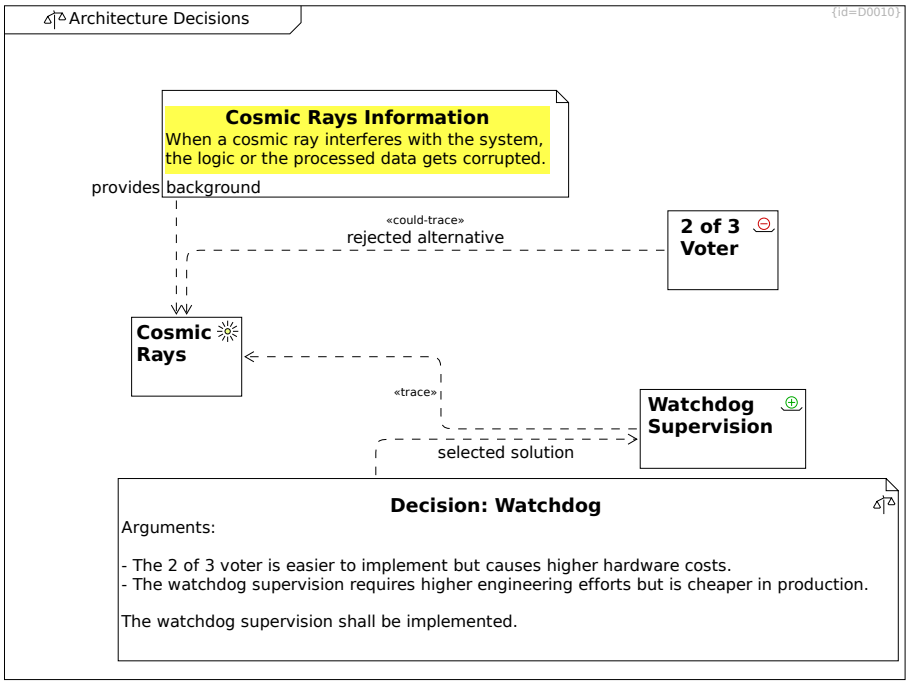
Step motors can be controlled to move a defined number of steps forward or backwards.

Note that there are conditions when the actual number of steps is not equal to the previously requested number of steps, e.g. when accellerating or slowing down too fast.

1.9 Architecture Decisions

Architecture Decisions «decision» D0010

This section documents the major design decisions. (Solution Space, System Level L1)



Cosmic Rays «environment» C0002 *(appears in Section 1.2: Constraints, Section 1.9: Architecture Decisions)*

The droid shall ensure data and program integrity.

It shall continue operation after cosmic rays have interfered with data storage or program execution.

Corrupted data must not be stored permanently.

2 of 3 Voter «rejected_alternative» C0100 *(appears in Section 1.9: Architecture Decisions)*

In order to support integrity of the system, the logic boards and the data storages are deployed three times as three identical parts.

All three parts shall produce the same outcomes given the same input.

If one deviates, it's result is ignored and the part is rebooted.

rejected alternative «could-trace» ·-> Cosmic Rays R0135

Cosmic Rays Information C0101 *(appears in Section 1.9: Architecture Decisions)*

When a cosmic ray interferes with the system, the logic or the processed data gets corrupted.

provides background ·-> Cosmic Rays R0132

Watchdog Supervision «chosen_alternative» C0102 *(appears in Section 1.9: Architecture Decisions)*

In order to support integrity of the logic and data, a multi-stage hierarchy supervision shall be implemented.

Software watchdogs shall supervise the running software parts in a way that logic errors and corrupted data can be detected.

A hardware watchdog shall supervise the software watchdogs.

In case of a failure in the supervised logic/data, the system shall reboot. In case of a failure in the monitors, the system may reboot or it shall fall back to a valid supervision mode.

·-> Cosmic Rays R0134

Decision: Watchdog «decision» C0103 *(appears in Section 1.9: Architecture Decisions)*

Arguments:

- The 2 of 3 voter is easier to implement but causes higher hardware costs.

- The watchdog supervision requires higher engineering efforts but is cheaper in production.

The watchdog supervision shall be implemented.

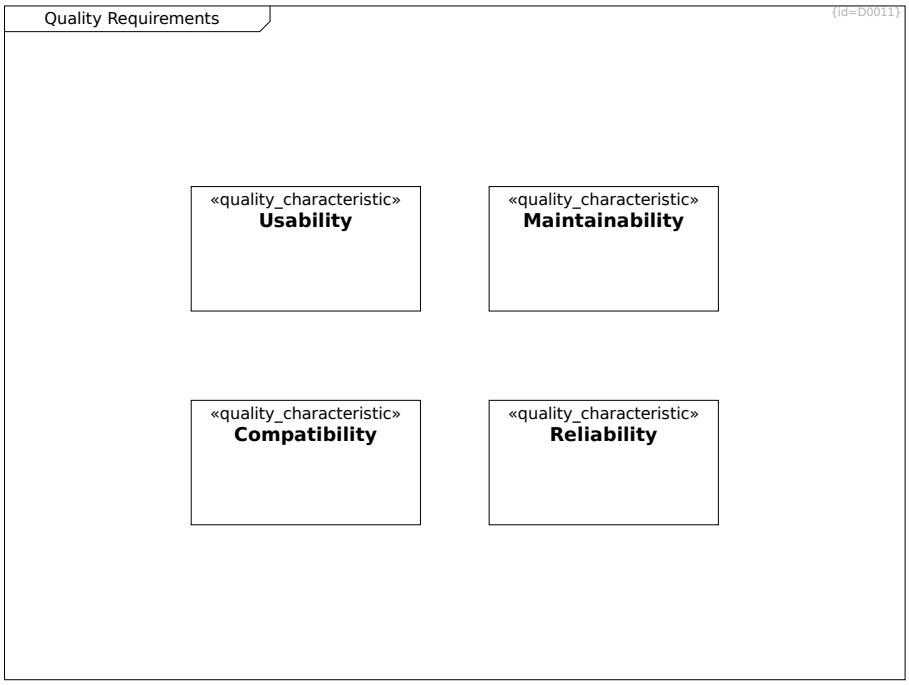
selected solution ·-> Watchdog Supervision R0133

1.10 Quality Requirements

Quality Requirements D0011

This section shows the major quality requrements and scenarios. (Problem Space, System Level L1)

In the following, requirements and scenarios are selected that show the quality expectations: The WHAT shall be implemented, not the HOW.



Usability «quality_characteristic» C0009 *(appears in Section 1.10: Quality Requirements)*

Usability defines a set of attributes that measures how easy to learn and use the program is.

Maintainability «quality_characteristic» C0010 *(appears in Section 1.10.1.1: Maintainability, Section 1.10.1: Quality Tree, Section 1.10: Quality Requirements)*

Maintainability defines a set of attributes that influence how to analyze and mitigate defects that occur during operation.

Reliability «quality_characteristic» C0011 *(appears in Section 1.10: Quality Requirements)*

Reliability defines a set of attributes that measures how mature and fault-tolerant the software is.

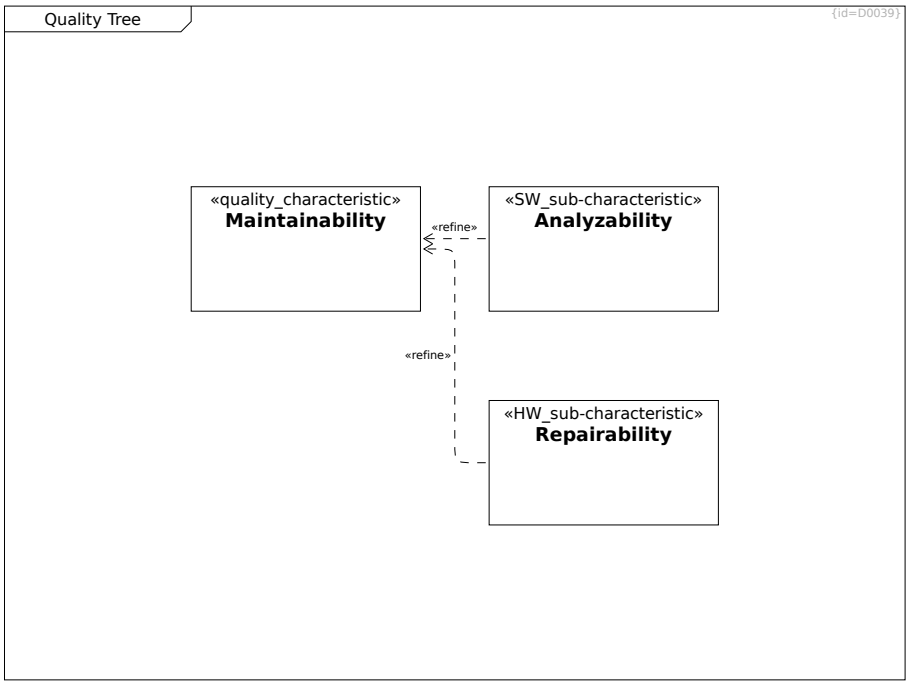
Compatibility «quality_characteristic» C0122 *(appears in Section 1.5.3.10.1: Quality Tree, Section 1.10: Quality Requirements, Section 1.5.3.10: Quality Requirements)*

Compatibility defines a set of attributes that measures how well data and messages can be exchanged with other programs and/or versions.

1.10.1 Quality Tree

Quality Tree D0039

This section shows the quality requirements ordered by quality characteristics.



Maintainability «quality_characteristic» C0010 *(appears in Section 1.10.1.1: Maintainability, Section 1.10.1: Quality Tree, Section 1.10: Quality Requirements)*

Maintainability defines a set of attributes that influence how to analyze and mitigate defects that occur during operation.

Analyzability «SW_sub-characteristic» C0014 *(appears in Section 1.10.1.1: Maintainability, Section 1.10.1: Quality Tree, Section 1.10.2: Quality Scenarios)*

The MoD5G shall allow to analyze faults that occurred during operation.

·-> Maintainability R0009

Repairability «HW_sub-characteristic» C0012 *(appears in Section 1.10.1.1: Maintainability, Section 1.10.1: Quality Tree, Section 1.10.2: Quality Scenarios)*

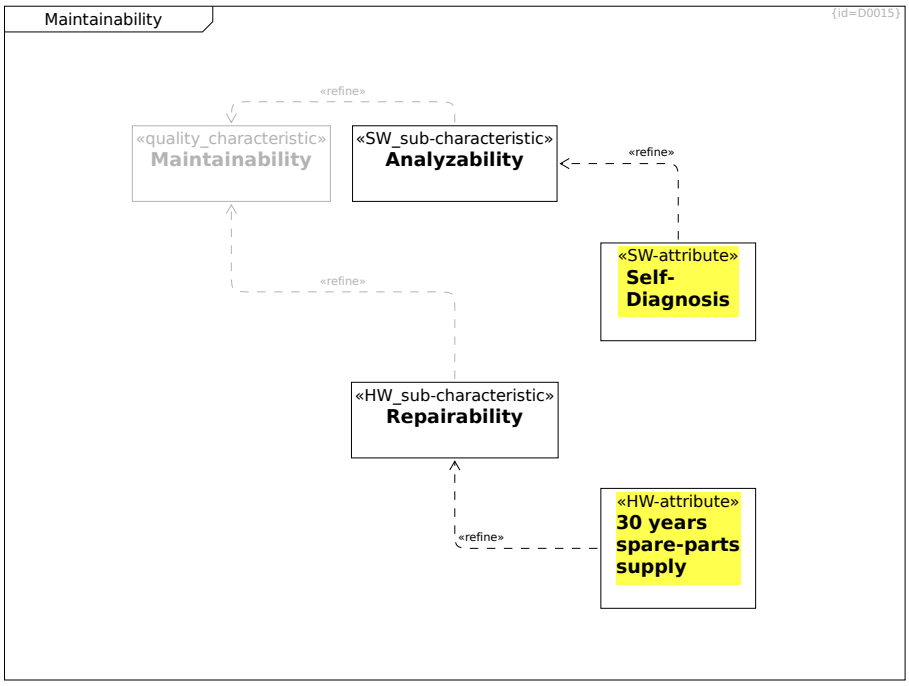
The MoD5 hardware parts shall be exchangeable in case they are damaged.

·-> Maintainability R0007

1.10.1.1 Maintainability

Maintainability D0015

This diagram shows the quality requirements related to the characteristic "Maintainability".



Maintainability «quality_characteristic» C0010 *(appears in Section 1.10.1.1: Maintainability, Section 1.10.1: Quality Tree, Section 1.10: Quality Requirements)*

Maintainability defines a set of attributes that influence how to analyze and mitigate defects that occur during operation.

Repairability «HW_sub-characteristic» C0012 *(appears in Section 1.10.1.1: Maintainability, Section 1.10.1: Quality Tree, Section 1.10.2: Quality Scenarios)*

The MoD5 hardware parts shall be exchangeable in case they are damaged.

·-> Maintainability R0007

30 years spare-parts supply «HW-attribute» C0013 *(appears in Section 1.10.1.1: Maintainability)*

The mechanical and electrical/electronics parts of the MoD5 shall be produceable in identical or similar form and quality for 30 years after production of the unit.

·-> Repairability R0008

Analyzability «SW_sub-characteristic» C0014 *(appears in Section 1.10.1.1: Maintainability, Section 1.10.1: Quality Tree, Section 1.10.2: Quality Scenarios)*

The MoD5G shall allow to analyze faults that occurred during operation.

·-> Maintainability R0009

Self-Diagnosis «SW-attribute» C0015 *(appears in Section 1.10.1.1: Maintainability)*

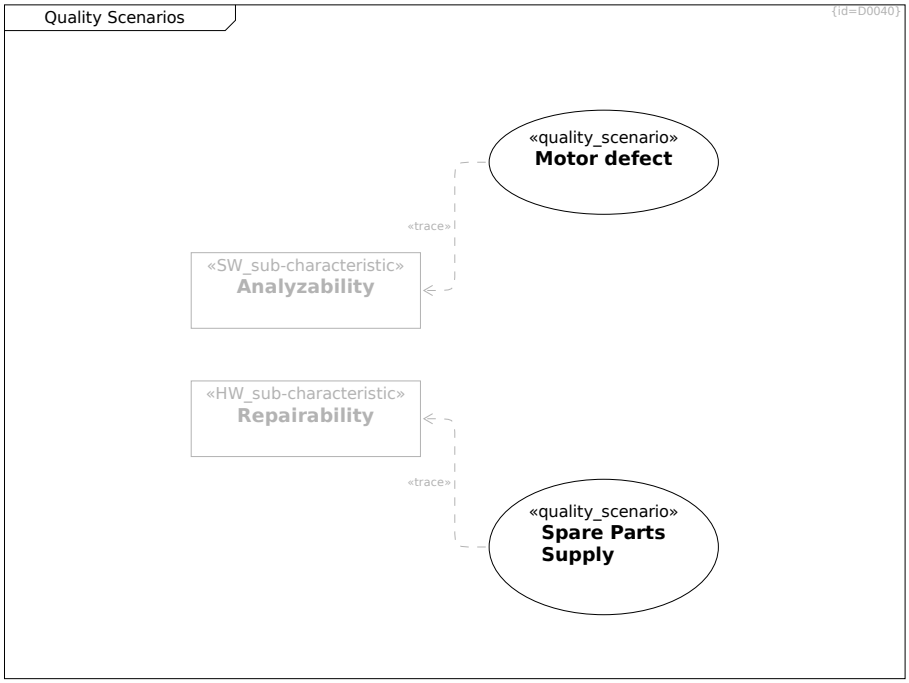
At the maintenance booth, the MoD5G shall provide an error log. This error log contains detected errors from operation and related environment conditions. It also lists possible causes(faults).

·-> Analyzability R0010

1.10.2 Quality Scenarios

Quality Scenarios D0040

This section shows the quality-related scenarios in which the quality requirements shown in Section 1.10.1: Quality Tree are of special importance.



Spare Parts Supply «quality_scenario» C0017 *(appears in Section 1.10.2: Quality Scenarios)*

pre-condition:
- the stock of MoD5G spare parts is empty

trigger:
- 20 years after production, a MoD5G needs a spare part that is not available anymore

scenario:
- a service mechanic orders a batch of parts
- a factory creates the parts that fit in form, function and quality to the MoD5G
- spare parts are delivered

·-> Repairability R0012

Motor defect «quality_scenario» C0016 *(appears in Section 1.10.2: Quality Scenarios)*

pre-condition:
- the MoD5G is performing a 1-day mission autonomously

trigger:
- a motor fails to operate
- the goals of the 1-day mission cannot be accomplished anymore

scenario:
- the MoD5G cancels the mission and returns to the service point
- a service mechanic reads out the error log
- the MoD5G proposes to replace the suspicious motor
- the service mechanic replaces the motor

·-> Analyzability R0197

Repairability «HW_sub-characteristic» C0012 *(appears in Section 1.10.1.1: Maintainability, Section 1.10.1: Quality Tree, Section 1.10.2: Quality Scenarios)*

The MoD5 hardware parts shall be exchangeable in case they are damaged.

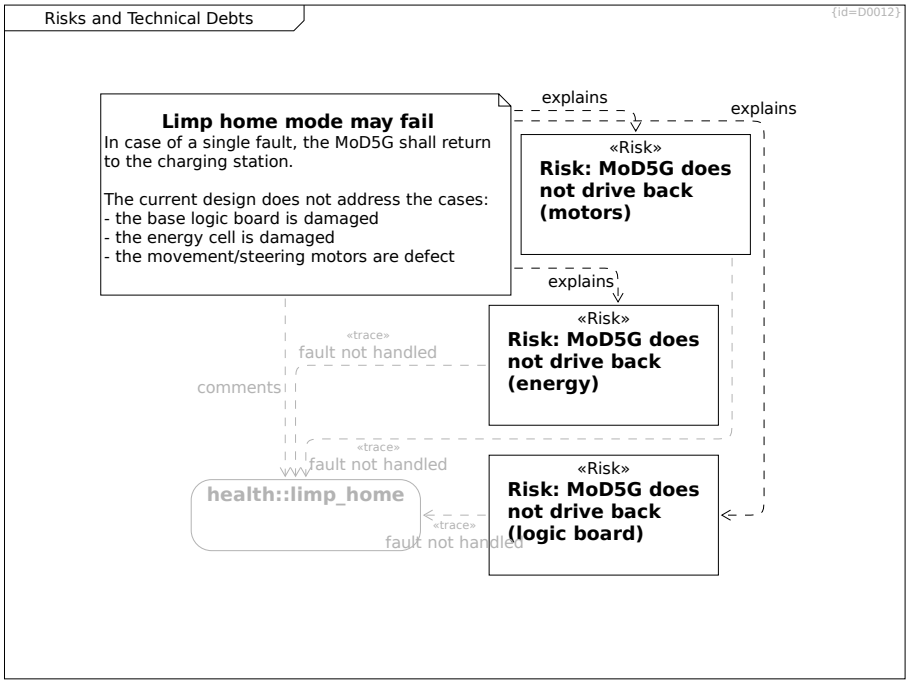
Analyzability «SW_sub-characteristic» C0014 *(appears in Section 1.10.1.1: Maintainability, Section 1.10.1: Quality Tree, Section 1.10.2: Quality Scenarios)*

The MoD5G shall allow to analyze faults that occurred during operation.

1.11 Risks and Technical Debts

Risks and Technical Debts D0012

This section lists the risks and not-yet-addressed requirements. (Solution Space, System Level L1)



Limp home mode may fail C0105 *(appears in Section 1.11: Risks and Technical Debts)*

In case of a single fault, the MoD5G shall return to the charging station.

The current design does not address the cases:

- the base logic board is damaged

- the energy cell is damaged

- the movement/steering motors are defect

comments ·-> health::limp_home R0136

explains ·-> Risk: MoD5G does not drive back (logic board) R0179

explains ··> Risk: MoD5G does not drive back (motors) R0229

explains ··> Risk: MoD5G does not drive back (energy) R0230

health::limp_home C0032 *(appears in Section 1.6.2: Health States, Section 1.11: Risks and Technical Debts)*
In case the full operation is not possible anymore, the MoD5G shall drive back to the home charging station.

Risk: MoD5G does not drive back (logic board) «Risk» C0133 *(appears in Section 1.11: Risks and Technical Debts)*
- cause/fault: the base logic board is damaged
- risk/failure: the MoD5G cannot drive anymore

fault not handled ··> health::limp_home R0180

Risk: MoD5G does not drive back (energy) «Risk» C0147 *(appears in Section 1.11: Risks and Technical Debts)*
- cause/fault: the energy cell is damaged
- risk/failure: the MoD5G cannot drive anymore

fault not handled ··> health::limp_home R0232

Risk: MoD5G does not drive back (motors) «Risk» C0148 *(appears in Section 1.11: Risks and Technical Debts)*
- cause/fault: the ,ovement/steering motors are damaged
- risk/failure: the MoD5G cannot drive anymore

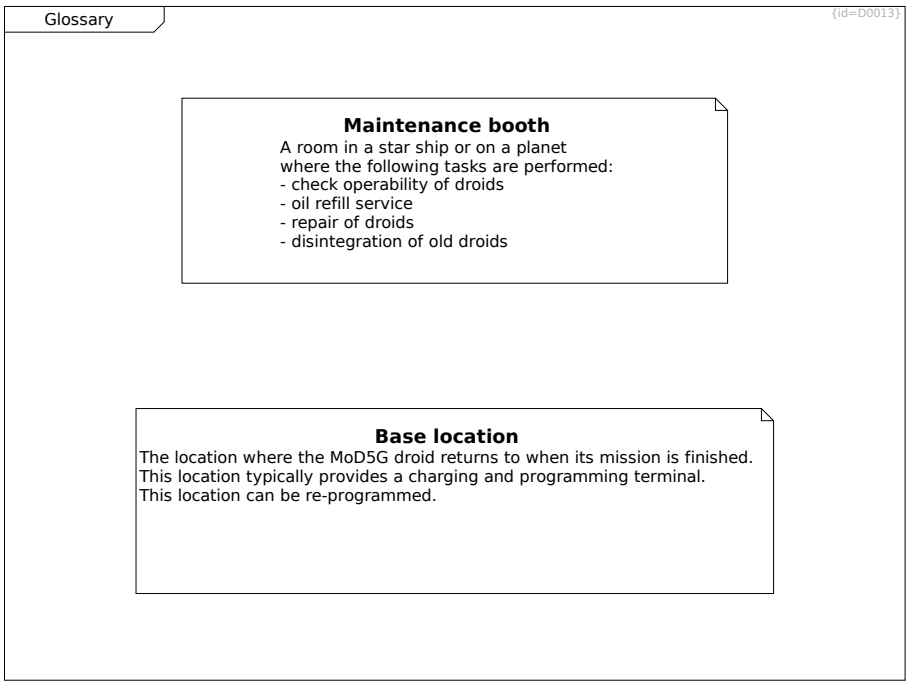
fault not handled ··> health::limp_home R0231

1.12 Glossary

Glossary D0013

This section explains the used terms. (Domain and Solution Space, System Level L1)

See also Section 1.5.3.12: Glossary for software terms.



Maintenance booth C0104 *(appears in Section 1.12: Glossary)*

A room in a star ship or on a planet where the following tasks are performed:

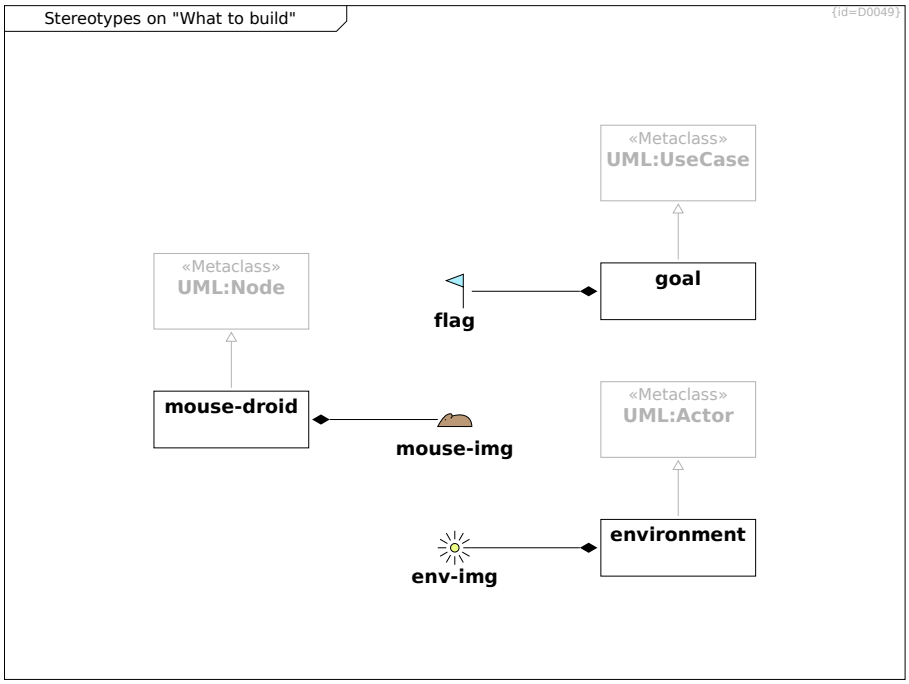
- check operability of droids
- oil refill service
- repair of droids
- disintegration of old droids

Base location C0143 *(appears in Section 1.12: Glossary)*

The location where the MoD5G droid returns to when its mission is finished. This location typically provides a charging and programming terminal. This location can be re-programmed.

1.12.1 Stereotypes on "What to build"

Stereotypes on "What to build" D0049



UML:Node «Metaclass» C0165 *(appears in Section 1.12.1: Stereotypes on "What to build")*

mouse-droid C0163 *(appears in Section 1.12.1: Stereotypes on "What to build")*

<path d=" c 3, -0.8 4.5, 0.5 4.5, 2.5 l -8, 0 c 1, -2, 2, -3, 3, -3 c 0.5, 0 0.5, 1 0, 1 m -1.1, 0.3 l -0.2, 0.2 " fill="#bb9977" />

*-- mouse-img R0271

--|> UML:Node R0272

mouse-img «mouse-droid» C0164 *(appears in Section 1.12.1: Stereotypes on "What to build")*

UML:UseCase «Metaclass» C0170 *(appears in Section 1.12.1: Stereotypes on "What to build")*

goal C0171 *(appears in Section 1.12.1: Stereotypes on "What to build")*

<path fill="#aaeeff" d=" M 5,9 L 5,1 1,2.5 5,4 "/>

--|> UML:UseCase R0277

*-- flag R0278

flag «goal» C0172 *(appears in Section 1.12.1: Stereotypes on "What to build")*

UML:Actor «Metaclass» C0174 *(appears in Section 1.12.1: Stereotypes on "What to build")*

environment C0175 *(appears in Section 1.12.1: Stereotypes on "What to build")*

<path fill="#eeff88" d=" M 1,0 C 1,0.55 0.55,1 0,1 C -0.55,1 -1,0.55, -1,0 C -1,-0.55 -0.55,-1 0,-1 C 0.55,-1, 1,-0.55, 1,0 "/> <path d=" M 2,0 L 4,0 M 1.73,1 L 3.46,2 M 1,1.72 L 2,3.46 M 0,2 L 0,4 M -1,1.72 L -2,3.46 M -1.73,1 L -3.46,2 M -2,0 L -4,0 M -1.73,-1 L -3.46,-2 M -1,-1.72 L -2,-3.46 M 0,-2 L 0,-4 M 1,-1.72 L 2,-3.46 M 1.73,-1 L 3.46,-2 "/>

--|> UML:Actor R0281

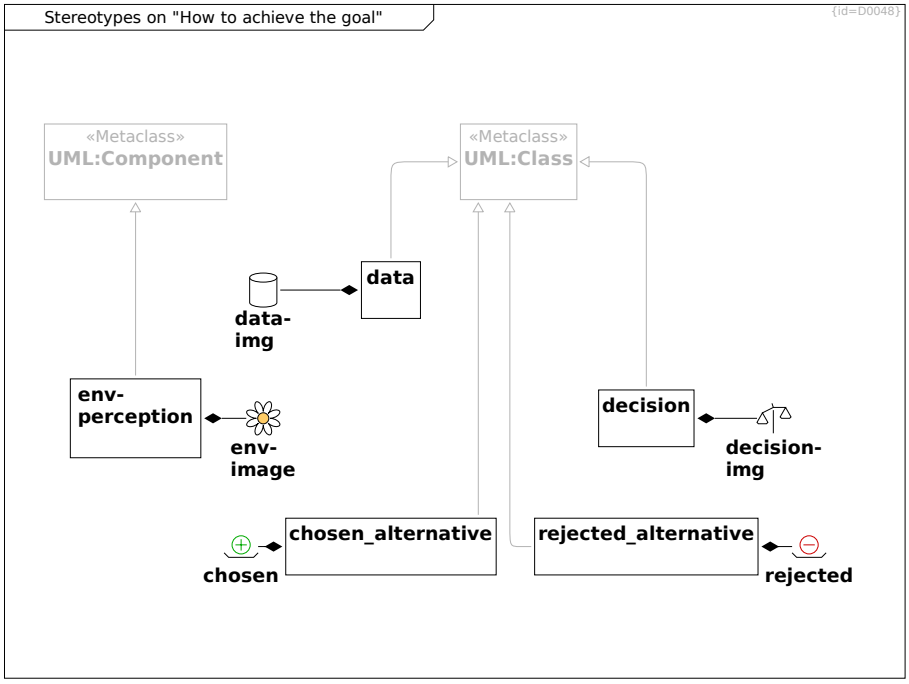
*-- env-img R0282

env-img «environment» C0176 *(appears in Section 1.12.1: Stereotypes on "What to build")*

1.12.2 Stereotypes on "How to achieve the goal"

Stereotypes on "How to achieve the goal" D0048

Stereotypes allow a classification of elements into project-specific categories.



UML:Component «Metaclass» C0155 *(appears in Section 1.12.2: Stereotypes on "How to achieve the goal")*

env-perception C0156 *(appears in Section 1.12.2: Stereotypes on "How to achieve the goal")*

<path fill="#ffcc66" d=" M 0.8,0 a 0.8,0.8 0 1 0 -1.6,0 a 0.8,0.8 0 1 0 1.6,0 " /> <path d=" M 1,0 A 1,0.5 22 1 1 0.707,0.707 A 1,0.5 67 1 1 0,1 A 1,0.5 112 1 1 -0.707,0.707 A 1,0.5 157 1 1 -1,0 A 1,0.5 202 1 1 -0.707,-0.707 A 1,0.5 247 1 1 0,-1 A 1,0.5 -68 1 1 0.707,-0.707 A 1,0.5 -23 1 1 1,0 " />

--|> UML:Component R0265

*-- env-image R0266

env-image «env-perception» C0157 *(appears in Section 1.12.2: Stereotypes on "How to achieve the goal")*

UML:Class «Metaclass» C0158 *(appears in Section 1.12.2: Stereotypes on "How to achieve the goal")*

decision C0159 *(appears in Section 1.12.2: Stereotypes on "How to achieve the goal")*

<path d="m 8,12 l -4,7 1,1 6,0 1,-1 -4,-7 l 8,-4 9,0 l -4,7 1,1 6,0 1,-1 -4,-7 " /><path d="m 15,5 l 1,3 m 0,2 l 0,17 " />

--|> UML:Class R0267

*-- decision-img R0268

decision-img «decision» C0160 *(appears in Section 1.12.2: Stereotypes on "How to achieve the goal")*

data C0161 *(appears in Section 1.12.2: Stereotypes on "How to achieve the goal")*

<path d="m 4,5 l 0,22 c 0,2.25 5.25,4 12,4 s 12,-1.75 12,-4 l 0,-22 " /><path d="m 4,5 c 0,-2.25 5.25,-4 12,-4 s 12,1.75 12,4 s -5.25,4 -12,4 s -12,-1.75 -12,-4 " />

--|> UML:Class R0269

*-- data-img R0270

data-img «data» C0162 *(appears in Section 1.12.2: Stereotypes on "How to achieve the goal")*

rejected_alternative C0166 *(appears in Section 1.12.2: Stereotypes on "How to achieve the goal")*

<path d="m 1,24 4,4 22,0 4,-4 " /><path stroke="#cc0000" d="m 8,17 c 0,-4.4 3.6,-8 8,-8 s 8,3.6 8,8 s -3.6,8 -8,8 s -8,-3.6 -8,-8 " /><path stroke="#cc0000" d="m 11,17 l 10,0 " />

*-- rejected R0273

--|> UML:Class R0274

rejected «rejected_alternative» C0167 *(appears in Section 1.12.2: Stereotypes on "How to achieve the goal")*

chosen_alternative C0168 *(appears in Section 1.12.2: Stereotypes on "How to achieve the goal")*

<path d="m 1,24 4,4 22,0 4,-4 " /><path stroke="#00aa00" d="m 8,17 c 0,-4.4 3.6,-8 8,-8 s 8,3.6 8,8 s -3.6,8 -8,8 s -8,-3.6 -8,-8 " /><path stroke="#00aa00" d="m 11,17 l 10,0 m -5,-5 l 0,10 " />

*-- chosen R0275

--|> UML:Class R0276

chosen «chosen_alternative» C0169 *(appears in Section 1.12.2: Stereotypes on "How to achieve the goal")*