Introduction to the „RAMI 4.0 Toolbox“

Author: Christoph Binder
Version: 2.1
Date: 2023-01-27

Table of Contents

1. Introduction

The document at hands describes the ideas and the usage of the *RAMI 4.0-Toolbox* for developing systems and applications based on *Industrie 4.0*. The toolbox with all its functionalities itself is derived from the ideas and knowledge gained from the *Referenzarchitekturmodell Industrie 4.0 (RAMI 4.0).* In reference to RAMI 4.0 the intention of the toolbox is to make those ideas applicable. Therefore, the main focus of this document is how to handle the RAMI 4.0-Toolbox and its' application. If you are looking for more specific information and underlying methods or technologies like Systems-Engineering, UML-based modeling or the handling of the involved modeling tool (*Enterprise Architect* from *Sparx Systems*[1]*)*, please take a look at the corresponding literature or feel free to contact us.

[1] www.sparxsystems.com

2. Referenzarchitekturmodell Industrie 4.0 (RAMI 4.0)

The Referenzarchitekturmodell Industrie 4.0 (RAMI 4.0) has been developed and introduced by several industrial associations located in Germany, BITKOM, VDMA and ZVEI. The project has been launched in order to secure a future-oriented, coordinated initiative across all industrial sectors. According to this RAMI 4.0 focuses on a structured description of a distributed Industrie 4.0 system in order to identify standardization gaps. However, the proposed architecture model appears to be very useful for architecting such systems. Hence, the RAMI 4.0-Toolbox was developed in order to ease the modeling of industrial applications in reference to RAMI 4.0.
The architecture model helps to analyze Industrie 4.0 systems and interactions by mapping them to a three dimensional cube that is depicted in Figure 1 – Figure 1.

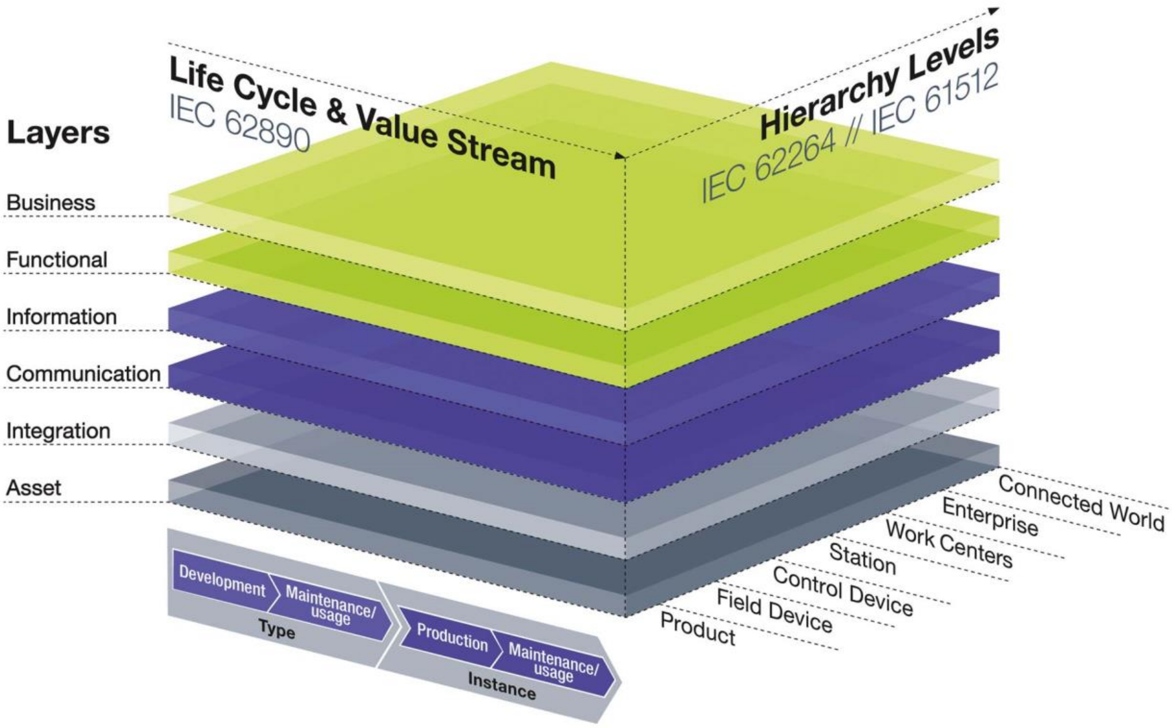


Figure 1 – Referenzarchitekturmodell Industrie 4.0 (RAMI 4.0)

2.1. The Interoperability Layers

In order to allow a clear presentation and simple handling of the architecture model, the interoperability categories are aggregated into six abstract interoperability layers.

Business Layer: The business layer represents the business view on the information exchange related to industrial processes. RAMI 4.0 can be used to map regulatory and economic (market) structures and policies, business models, business portfolios (products & services) of market parties involved. Also business capabilities and business processes can be represented in this layer. In this way it supports business executives in decision making related to (new) business models and specific business projects (business case) as well as regulators in defining new market models.

Function Layer: The function layer describes functions and services including their relationships from an architectural viewpoint. The functions are represented independent from actors and physical implementations in applications, systems and components. The functions are derived by extracting the use case functionality which is independent from actors.

Information Layer: The information layer describes the information that is being used and exchanged between functions, services and components. It contains information objects and the underlying canonical data models. These information objects and canonical data models represent the common semantics for functions and services in order to allow an interoperable information exchange via communication means.

***Communication Layer*:** The emphasis of the communication layer is to describe protocols and mechanisms for the interoperable exchange of information between components in the context of the underlying use case, function or service and related information objects or data models.

Integration Layer: The integration layers' main purpose is to provide all physical assets to the other layers in order to create events in the form of so called administration shells. Those shells represent the foundation for further processing and therefor provide information to do so. To show the context of each asset, the integration layer also provides the usage and integration of network components like routers, switches, terminals or passive ones like barcodes and QR-codes.

Asset Layer: The emphasis of the component layer is the physical distribution of all participating components in the smart grid context. This includes system actors, applications, physical components as well as documents, ideas and human beings.

2.2. The Industrie 4.0 Plane

Every layer itself is depicted by the utilization of the Industrie 4.0 Plane, which is defined as follows:

An application based on cyber physical systems distinguishes between electrical process and information management viewpoints. These viewpoints can be partitioned into the physical state the asset finds itself known as life cycle and value stream and the hierarchical zones for the management of the cyber physical system (refer to [IEC62264-1:2013, IEC 61512-1:1997]).

These two concepts together build the Industrie 4.0 plane. This allows to enable the representation on which area the interaction between single assets take place as well as the classification of those from a management point of view. According to this concept every asset has its own life cycle, depending in which state it actually is. Furthermore, to get control over information management, the asset has to be classified according to the hierarchical zones. This depends on the role it fulfills as well as the location given in the value chain.

2.3. Life Cycle & Value Stream

The Industrie 4.0 Plane covers the complete life cycle of the corresponding assets, as described in Table 1 – RAMI 4.0 Life Cycle & Value Stream.

Life Cycle	Description
Type Development	This represents the first idea of a product. At this stage the every aspect around the product is displayed, from commissioning to development, testing and the generation of the first prototypes.
Type Maintenance	Representing the result from the development stage, this shows the first model or prototype of the machine or product.
Instance Production	After specifying the requirements and generating a type, all products are developed after this template. This stage represents the development of a single part before being unique.
Instance Maintenance	The final product or machine is represented here. To meet the needs of this stage, a part has to be unique and in usage.

Table 1 – RAMI 4.0 Life Cycle & Value Stream

2.4. Hierarchy Levels

These levels reflect a hierarchical model which considers the concept of aggregation and functional separation in management of cyber physical systems. The basic idea of this hierarchical model is laid down in the Purdue Reference Model for computer-integrated manufacturing which was adopted by IEC 62264-1 standard for —enterprise-control system integration [IEC 62264-1:2013]. Furthermore, in order to cover a broader area, the model has been expanded by the standard of Batch Control [IEC 61512-1:1997].

The partitioning in levels follows the concept of functional separation. Different functions are assigned to specific zones. The reason for this assignment is typically the specific nature of functions, but also considering user philosophies. Real-time functions are typically in the field and station level and below (metering, protection, phasor-measurement, automation…). Functions which cover an area, multiple substations or plants are usually located in work centers level and above. RAMI 4.0s hierarchy levels are described in Figure 2.

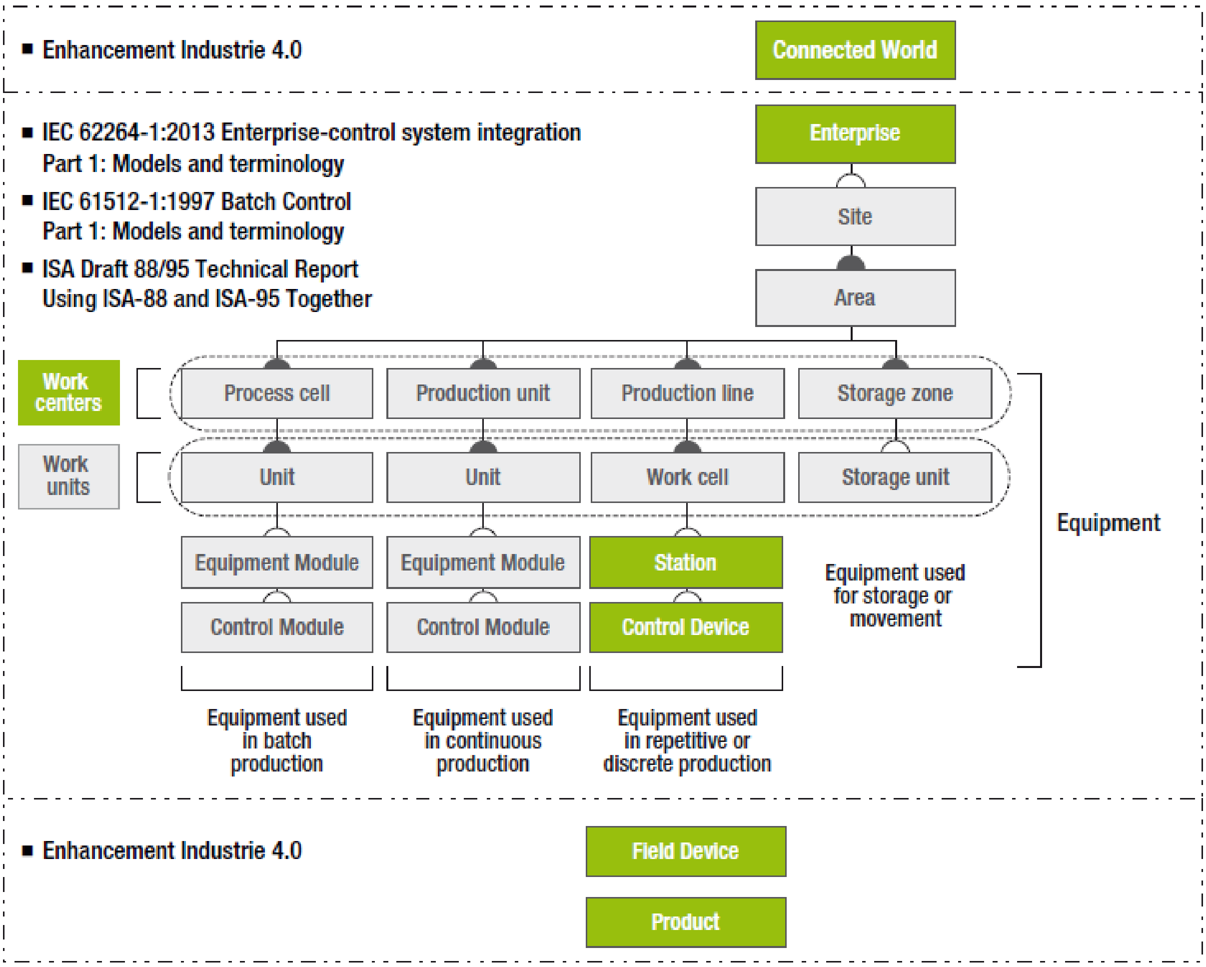


Figure 2: RAMI 4.0 Hierarchy Levels

3. RAMI 4.0 Toolbox Installation

The Toolbox itself comes with an MSI-Installer. All data used is copied to the folders where they are needed. Just go through the wizard and you are ready to go.

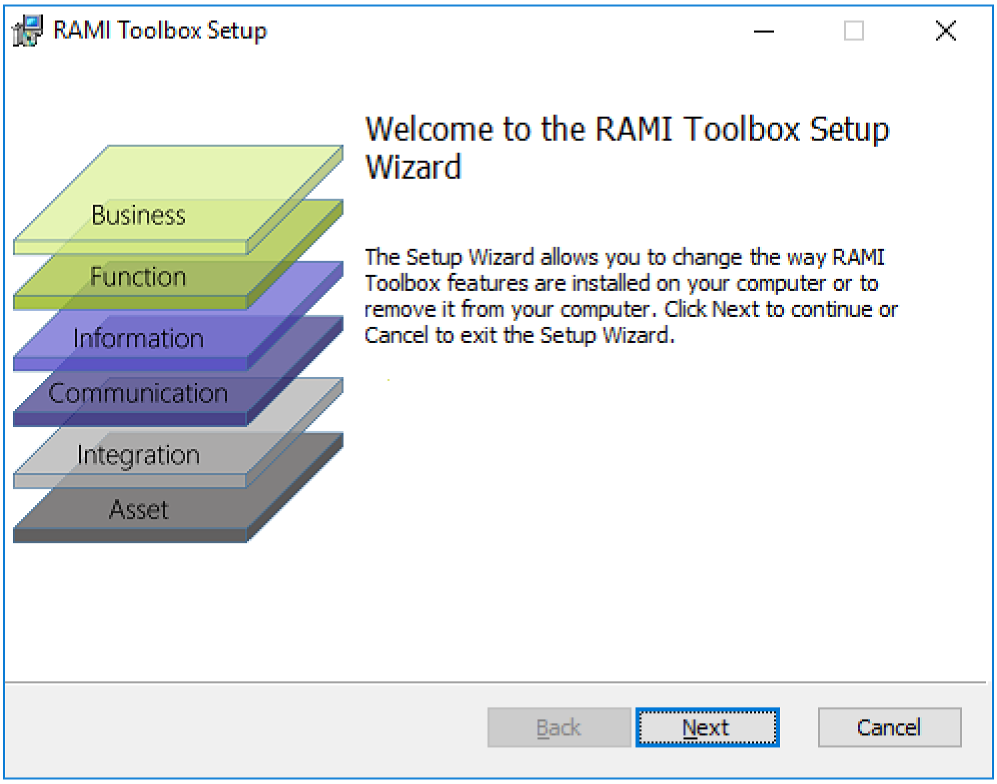


Figure 3: RAMI 4.0 Installation Wizard

4. Terms and Definitions

Business Layer	Identification of Requirements by elaborating the System Context, Stakeholders and Processes
System of Interest	The system whose life cycle is under consideration
Use Case	Description of the behavioral requirements of a system and its interaction with a user.
Business Case	Use Case focusing on economic aspects
High Level Use Case	Specific functionality to be realized in order to fulfill a certain BC
Business Actor	Legal or physical entity having individual Business Goals
Force	The cause of change in the state of motion of a particle or body (Uncontrollable Interference or Disturbance affecting the SoI)
Process	Set of interrelated or interacting activities that use inputs to deliver an intende d result
Business Process	A business process consist of a set of activities that are performed in coordination in an organizational and technical environment. These activities jointly realize a business goal. Each business process is enacted by a single organization, but it may interact with business processes performed by other organizations.
Activity (BPMN)	Activities describe the kind of work being done in a particular process instance
Task (BPMN)	The most basic level of an activity and cannot be broken down further
Kaizen Burst	Improvement Measure
Manufacturing Process	represents a single department, a process or a machine with a fixed and continuous internal material flow.
External Source	In the upper left corner of a value stream representation, the usual starting point for the material flow, this symbol stands for the supplier. It represents the customer in the upper right corner.
Databox	The data box is placed under other symbols that require data for system analysis. For example, a data box under a factory symbol could show the delivery frequency, product processing data, lot size or other information.
Transport Truck	The truck symbol stands for an external delivery to customers or from suppliers.
Transport Stapler	Used when something needs to be moved with a forklift.
Transport Symbols	These symbols are easy to understand: delivery on rails is represented by a train, air freight by an aircraft and delivery by ship by a ship symbol.

Stock	The inventory between two processes is represented by these symbols. If you need to add an inventory, you can note it under the triangle. This symbol can also show stocks.
Push	This symbol shows material that is being pushed from one process downstream to the next.
Supermarket	This symbol represents a Kanban storage location from which customers can obtain the required inventory downstream; the supplier then fills it upstream.
Goods	This symbol shows materials that come from suppliers or finished products that go from the factory to the customers.
Withdrawal	This pull symbol shows the physical withdrawal of inventory from supermarkets.
FIFO	This symbol represents a FIFO system (first-in, first-out) that limits the inventory receipt. You can write the maximum inventory capacity under the web.
Puffer	Instead of standing stocks, this symbol shows a temporary reserve stock, which should prevent problems with system failures and other things.
Manual Stream	Manual flow of information from memos, reports or conversations. If necessary, determine the type of information.
Electronic Stream	Digital information stream, e.g. B. Internet, intranet, electronic data exchange, etc.; Frequency, type of data and media used can be recorded.
Information System	Planning with an inventory control system such as material requirements planning (MRP).
Equalization	A tool that divides kanbans into lots to compensate for the differences and volume fluctuations in production.
Production Planning	Sometimes information is gathered through observation, e.g. in a production decision by a supervisor after a visual inspection of the inventory.
Employee	This symbol represents a verbal flow of information.
Rework	Percentage of rework
Discard	Percentage of discard
Function Layer	Developing Functions that fulfill the Requirements, describing their Black- and White-Box perspective and Information exchange, assign Functions to actual System Components
Activity (UML)	Description of one or more behavioral elements from the coordinated sequencing of actions that take place when a use case is instantiated
Action (UML)	The most basic level of an activity and cannot be broken down further
Function	An action, a task, or an activity performed to achieve a desired outcome through two or more different combinations of elemental behavior with defined performance encompassing multiple related disciplines. Input/output relationship of information (signals, data), materials, force or energy within the system of interest, or a model thereof.

Functional Element	Abstract system element that defines a relation between at least one input and at least one output by means of a function.
Functional Group	Set of strongly related use case activities.
Information Layer	Which technology (Protocol, Data Structure, Data Storage, ...) is used for Information exchange, e.g. JSON, XML, RDBMS, Hadoop, ...
Information Item	Specify particular information to be exchanged.
Data Model Standard	Cover the exchange of information, specified by the Information Items.
Information Object Flow	Relation between two components associating particular Information Items.
External Entity	An outside system that sends or receives data, communicating with the system being diagrammed.
Data Source	Files or repositories that hold information for later use, such as a database table or a membership form.
Data Flow	The route that data takes between the external entities, processes and data stores.
Communication Layer	Which infrastructure (Protocol, Interfaces, Communication Canal, ...) is used for Information exchange, e.g. LAN, NFC, OPC UA, ...
Port	Gate between component and environment including all hardware specifications.
Service Point	Data provider.
Request Point	Data requester.
Expose Interface	Interface either dealing as data provider or as data requester.
Interface	Providing publicly available information of an component to its environment.
Assembly	Relation between two components associating their physical interconnection.
Integration Layer	Making Components IIoT (industrial internet of things) applicable by describing HMIs, ICT infrastructures or the Administration Shell itself, e.g. API, QR Codes, Server, Switches, ...
Administration Shell	The AAS is used to describe an asset electronically in a standardized manner for exchanging asset-related data among industrial assets and between assets and production orchestration systems or engineering tools
Asset Layer	Technical Design of the Components

5. Specification of RAMI 4.0

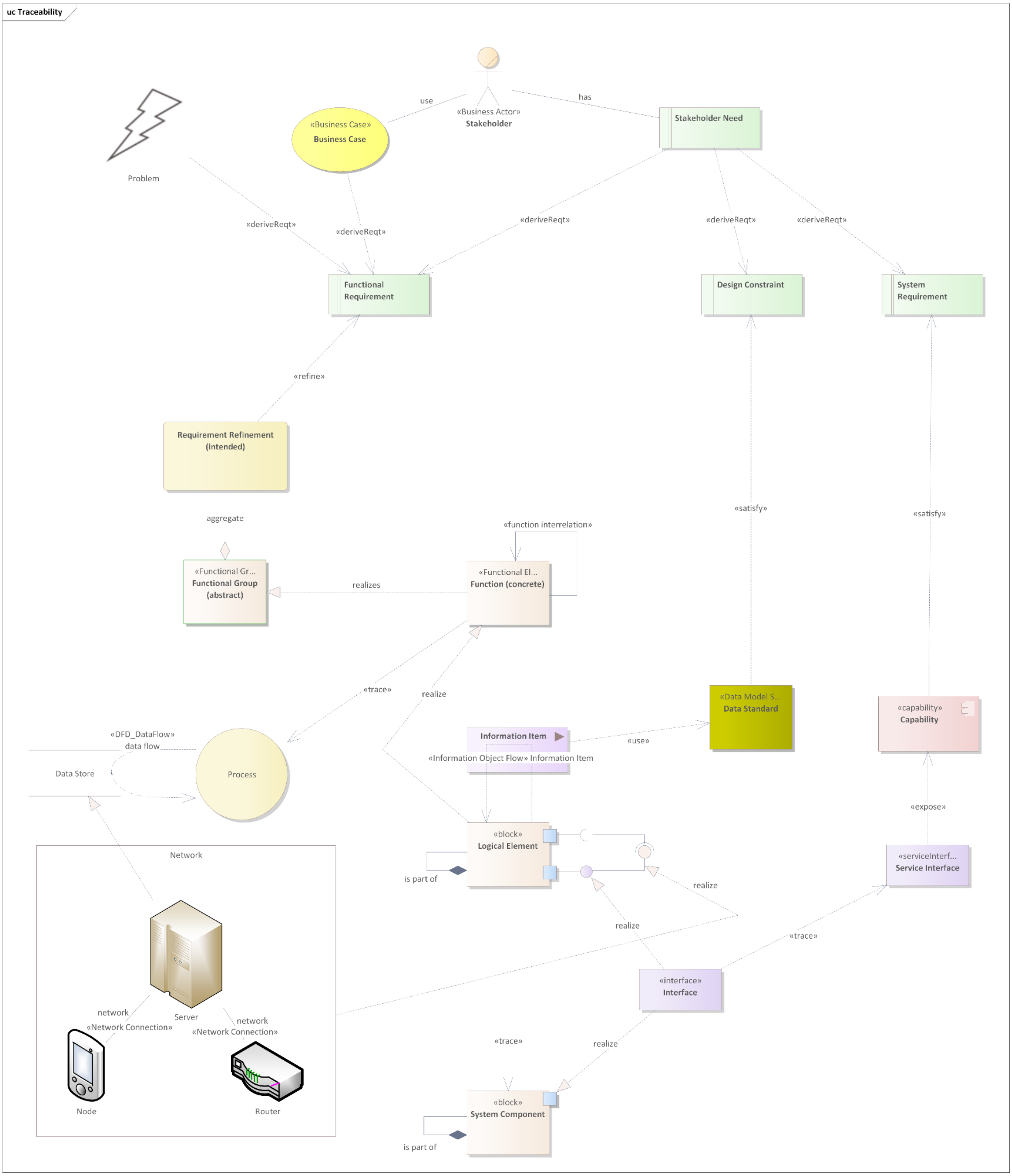
This section describes the refinement of RAMI 4.0 for specifying the theoretical concepts.

Defining Models and Model Kinds for each of the RAMI Layers.

VP	Model	Model Kind	ML	Diagram
BL	Context & Constraint Model	Context Diagram Constraint Diagram	DSL SysML	XX_System Context XX_Constraints
BL	Goals Model	Business Case Diagram Use Case Diagram	DSL UML	XX_Business Cases XX_Scenarios
BL	Requirements Model	Requirements Diagram	SysML	XX_Requirements
BL	Process Model	Business Process Diagram SIPOC Manufacturing Diagram	BPMN DSL	XX_Business Processes XX_Manufacturing Processes
FL	Function Constraint Model	Requirements Diagram Function Standard Diagram	SysML DSL	XX_Functional Requirements XX_Functional Standards
FL	Function Interconnection Model	Function Diagram	DSL	XX_Functional Architecture
FL	Functional Architecture for Systems Model	Activity Diagram Function Grouping Diagram	UML DSL	XX_Activities XX_Functional Grouping
FL	Black- & White-Box Model	Block Definition Diagram Internal Block Diagram	SysML	XX_Black-Box XX_White-Box
IL	Information Constraint Model	Requirements Diagram Information Standard Diagram	SysML DSL	XX_Information Requirements XX_Information Standards
IL	Information Exchange Model	Data Flow Diagram Information Layer Diagram	DFD DSL	XX_Data Flow XX_Information Exchange

IL	Information Object Model	Information Layer Diagram	DSL	XX_Information Objects
CL	Communication Constraint Model	Requirements Diagram Communication Standard Diagram	SysML DSL	XX_Communication Requirements XX_Communication Standards
CL	Communication Flow Model	Communication Diagram Component Diagram	DSL DSL	XX_Communication Flow XX_Interfaces
CL	Network Infrastructure Model	Network Infrastructure Diagram Network Technology Diagram	DSL	XX_Network Infrastructure XX_Network Technologies
CL	Service Model	Service-oriented Architecture Diagram	SoaML	XX_Services
GL	Capability Model	AutomationML Diagram	DSL	XX_Capabilities
GL	Logical Model	Logical Diagram	SysML	XX_Logical Architecture
GL	Digital Twin Architecture Model	Asset Administration Shell Diagram	DSL	XX_Digital Twin
AL	Cyber-physical System Model	Block Definition Diagram Internal Block Diagram	SysML	XX_Cyber Physical System Composition
AL	Phyiscal Connection Model	Technology Component Diagram	UML	XX_Technologies
				XX … Hierarchy Level XX … Component of Interest

5.1. Metamodel



5.2. Zachman Framework Integration:

RAMI 4.0 and the integration of the Zachman Framework appear to provide a suitable and comprehensive architecture framework to describe all aspects of Industry 4.0 systems. They span a Matrix that provides multiple panes, each one dealing with a specific aspect of a complex industrial system. While either IIRA or SPES are too generic and difficult to apply in such complex applications, RAMI 4.0 and the automation pyramid is specifically designed for such scenarios. With this panes, multiple stakeholders of the manufacturing system can be addressed and find their needed information in a particular area, while the interconnections in the whole system can also be described in this architecture. Additionally, by also specifying the architecture of the Smart Product, the corresponding architecture of the production system can directly be aligned and mutually developed or adapted to each other.

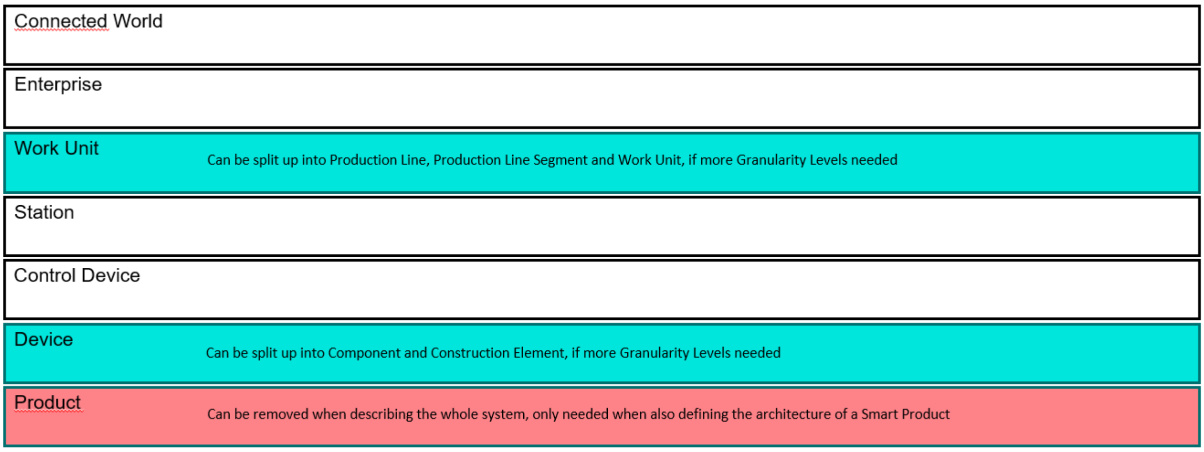
	Business Layer	Function Layer	Information Layer	Communication Layer	Integration Layer	Asset Layer
Connected World						
Enterprise						
Work Unit						
Station						
Control Device						
Device						
Product						

5.3. Row Description:

By introduction different Granularity Levels, the white box – black box perspective can be modeled. While on the upper granularity level a particular component is considered as black box, the white box perspective of the same component is modeled on the layer beyond.

Level	Core Question	Punching Station Example
Connected World	What are the Business Strategy and Business Plans?	Business Entities: Business Processes + Components
Enterprise	What are the required activities and facilities of the System to produce end products or an intermediate good?	Factories: Plastic Housing Factory, Plastic Housing Warehouse
Work Unit	Which elements (stations) comprise this work unit for manufacturing a different part of the End Product? What are their relationships and interfaces? What are the Components executing the single non-divisible Processes for manufacturing the product?	Production Lines: Plastic Housing Production Line Context of Punching Segment
Station	What is the surrounding Environment to enable the value adding non-divisible Production Processes?	Work Stations: Punching Station + Components
Control Device	What is needed to control the equipment to ensure production of product?	Machines + Controllers: Punching Machine + PLC
Device	What are the Sensors/Actors of the System? What are the Machines executing the Functions/Work Instructions?	Single Devices executing one specific Task: Measurement Sensor Placement Actor Puncher
Product	What is the Architecture of the Product, the Production System is manufacturing? Which manufacturing steps are needed to produce the product?	Products to manufacture: Plastic Housing

By using the approach of the Matrix, 36 different panes are provided to model the system. If another level of refinement is required, a new granularity level can be added to expand the model of the system. For example, Biffl et al. (2017) are using 9 different granularity levels. Each of the single panes however Introduces a particular view to the system by providing models. How to model the view is done by the systems engineer, who is able to customize the pane according to his needs. Only a model template including a DSL, describing a possible way to realize the viewpoint, is provided. There are no restriction however, which models are used to describe the system. If a specific modeling element is needed for a particular part of the system, the reference architecture and its DSL can be customized as desired.



5.4. Column Description:

By introducing the hierarchy levels of RAMI 4.0, different aspects of the system are modeled in different viewpoints. This is how the scope of the Systems Engineer can change from the overall system to the concern of one particular stakeholder.

Business Layer	Function Layer	Information Layer	Communication Layer	Integration Layer	Asset Layer
What is the Context of the System? What are operational scenarios that should be supported/ fulfilled by the system? What are actors and what are their goals?	What are the Functions of the System? What are the processes of the system?	What are the Information Structures and Information Flows inside the System and in the context of the System?	What is the Communication Infrastructure of the System? What services are provided by the system?	How are the phyiscal elements of the systems enabled to communicate with other components and provide services or functions? Which elements are additionally needed to enable this integration?	What are the real world Components of the System? What is the Technical Specification of the Components?

5.5. Distinction between Greenfield and Brownfield:

A Greenfield is a model of a completely new system.

A Brownfield is a model of an already existing system.

This distinction has to be done, because in a Greenfield the system can be developed from scratch an no exisiting conditions need to be met. In a Brownfield, the production system is developed based on an already existing system and aims to enhance it. Both types of systems need to be considered, because both types usually occur for industrial systems.

With the modeling framework, both type of systems can be modeled. Taking the example of the architecture development standard "TOGAF", this is an iterative process. This means an AS-IS architecture is an architecture describing an industrial system as it is, while a TO-BE architecture is an architecture as we want the system to be. This means, a Brownfield can be compared with the AS-IS system architecture, while a Greenfield can be compared with the TO-BE system architecture. After defining either the AS-IS or TO-BE architecture of the system, in the next iteration always the TO-BE architecture is considered with a Brownfield system.

Thus, there are three different contexts when developing flexible production systems, where product in batch size 1, production system and production process is integratively developed.

A mapping between RAMI 4.0 and TOGAF can be found here: https://www.en-trust.at/papers/Binder21aa.pdf

In a Greenfield, a system can be modeled as desired by the Stakholders. Thus, the following development strategy is proposed.

	Greenfield
Step 1	Model the System Context with In- and Outputs of the SoI
Step 2	Model the User Interaction with the SoI, define Actors and Use Cases
Step 3	Derive Requirements from the SoI or the user interaction
Step 4	If there are any Functional Requirements, model them in this diagram and the Use Cases to Realize them
Step 5	For each Use Case of the system, model the Activity Diagram and the intended process to fulfill the Use Case
Step 6	Summarize similar Activity Diagram Elements to Functions, which have to be executed by the system
Step 7	For each Function, model expected Input, Output, Disturbance and Interference
Step 8	Model the Functional Architecture and Information Exchange with Information Objects between the Functions
Step 9	For each Function, choose a physical component to fulfill it
Step 10	Model the physical composition of the SoI with all physical components and elements that represent an physical aspect of the System
Step 11	Model the physical representation of the System Components
Step 12	Model all needed Systems to integrate the physical SoI as Digital Twin in order to enable data exchanges or Industry 4.0-compliant communication
Step 13	Model the Information Exchange with the previously defined Information Objects between the physical components
Step 14	Model the ICT infrastructure with used technologies or protocols between the systems' elements

In a Brownfield, the system architecture of the current system needs to be developed first. Thus, the following development strategy is proposed.

	Brownfield
Step 1	Model the System Context with In- and Outputs of the SoI
Step 2	Model the Business and Manufacturing Processes of the System
Step 3	Model the Functional Architecture with all Functions of the current System
Step 4	For each Function, model Input, Output, Disturbance and Interference
Step 5	Model the Information Exchange with the previously defined Information Objects between the Functions
Step 6	For each Function, connect the physical component that fulfills it
Step 7	Model the ICT infrastructure with used technologies or protocols between the systems' elements
Step 8	Model all used Systems to integrate the physical SoI as Digital Twin in order to display data exchanges or Industry 4.0-compliant communication
Step 9	Model the physical composition of the SoI with all physical components and elements that represent an physical aspect of the System
Step 10	Model the physical representation of the System Components

After either specifying a Greenfield or Brownfield architecture, the next iteration of architecture development is based on an already existing system. This is also done for all future iterations. Thus, the following development strategy is proposed.

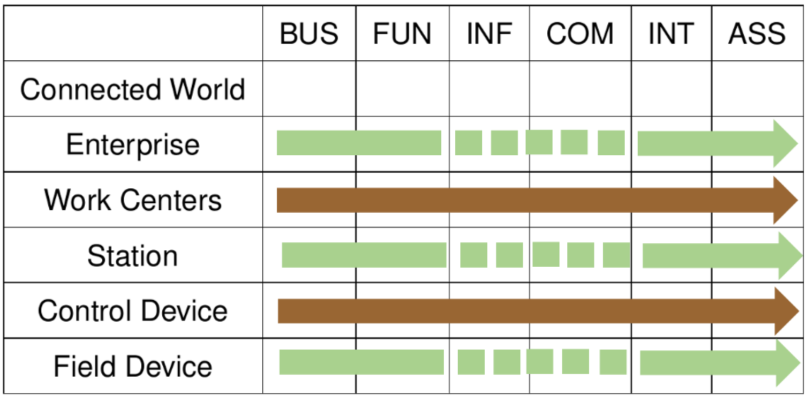
	Greenfield/Brownfield + 1
Step 1	Adapt the System Context with In- and Outputs of the SoI
Step 2	Model the User Interaction with the SoI, define Actors and Use Cases
Step 3	Derive Requirements from the SoI or the user interaction
Step 4	If there are any Functional Requirements, model them in this diagram and the Use Cases to Realize them
Step 5	For each Use Case of the system, look if any Business or Manufacturing Process already fulfills it
Step 6	If not, model the Activity Diagram and the intended process to fulfill the Use Case
Step 7	Summarize similar Activity Diagram Elements to Functions, which have to be executed by the system
Step 8	Look if any existing System Functions fulfill them, otherwise specify new ones
Step 9	For each new Function, model Input, Output, Disturbance and Interference
Step 10	Model the Functional Architecture and Information Exchange with Information Objects between the Functions
Step 11	For each Function, look if a physical component fullfils it
Step 12	If not, create new physical Components and connect all Functions to them
Step 13	Adapt the physical composition of the SoI with all physical components and elements that represent an physical aspect of the System
Step 14	Adapt the physical representation of the System Components
Step 15	Adapt all needed Systems to integrate the physical SoI for the new Components
Step 16	Adapt l the Information Exchange with the previously defined Information Objects for the new physical components
Step 17	Adapt the ICT infrastructure with used technologies or protocols between the systems' new elements

5.6. Development Strategies

This page addresses the collaboration across different teams based on the modeled system architecture. While each team is responsible for specific aspects of a component, for example requirements, functions or technical aspects, they need to exchange models with each other and build up on their respective results. Another case would be a change to the existing system model, which should be integrated into all viewpoints and levels of the model. Therefore, based on the matrix of the architecture modeling framework, different ways to exchange those models or distribute changes from a diagram to other diagrams within the model need to be available. In conclusion, when modeling a flexible production system according to the architecture, the following four directions can be addressed:

<u>Model Driven Engineering:</u>

From left to right, the RAMI 4.0 layers are iterated top-down, which describes the systems from requirements to technical components. Thereby, the method popular with systems engineers is applied and solutions can be created from previously outlined problem spaces.



<u>Greenfield:</u>

Business:

Modeling Steps: System Context → Business Cases → Processes → Requirements

Business → Function: Pass the Requirements Diagram to the Function Layer.

Function:

Modeling Steps: FAS → Functional Architecture → Black- & White-Box

Function → Information: Model Transformation from Functional Architecture to Process Steps

Information:

Modeling Steps: Data Flow

Information → Communication: Pass Data Flow Diagram

Communication:

Modeling Steps: Communication Interconnection

Communication → Integration: Use Process Steps, Data Flows and Interconnections to make an Architecture Decision for Logical Components

Integration:

Modeling Steps: Logical Components

Integration → Asset: Model Transformation to Physical Components

150% Model

Asset:

Modeling Steps: Physical Elements, Physical Connection

<u>Brownfield:</u>

Business:

Modeling Steps: System Context → Business Cases → Processes → Requirements

Business → Function: Model Transformation from Business Cases / Processes to Functions

Function:

Modeling Steps: Functional Architecture → Black- & White-Box

Function → Integration: Consider Logical Elements that fulfill your Functions

Integration:

Modeling Steps: Logical Components → Logical Architecture

Integration → Asset: Model Transformation to Physical Components

150% Model

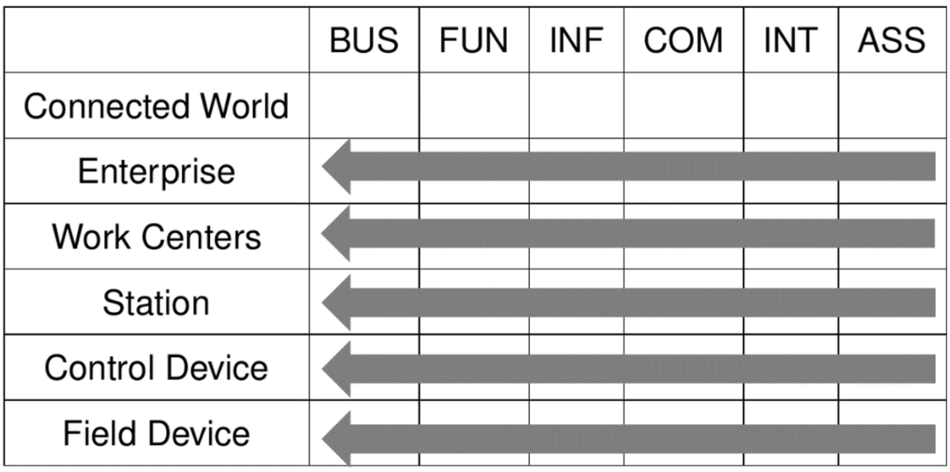
Asset:

Modeling Steps: Physical Elements, Physical Connection

Continue with Digital Twin Development →

Digital Twin Development:

From right to left, the RAMI 4.0 layers are iterated bottom-up, which enables the Digital Twin description of already existing physical elements. This can be applied if the system needs to be defined from already existing production lines.



Greenfield:

Asset:

Modeling Steps: Physical Elements → Physical Connection

Asset → Integration: Model Transformation to Virtual Components

Integration:

Modeling Steps: Asset Administration Shell → Logical Architecture → Logical Components

Integration → Communication: Pass Logical Components to Communication Layer

Communication:

Modeling Steps: Provided Services → Communication Infrastructure

Communication → Information: Pass Services & Infrastructure to Information Layer

Communication:

Modeling Steps: Information Exchange → Data Model Standards

Information → Function: Model Transformation Logical Components to Function Layer

Function:

Modeling Steps: FAS → Functional Architecture → Black- & White-Box Functions

Function → Business: Pass Functions to Business Layer

Business:

Modeling Steps: Requirements

Brownfield:

Asset:

Modeling Steps: Physical Elements → Physical Connection

Asset → Integration: Model Transformation to Virtual Components

Integration:

Modeling Steps: Logical Components

Integration → Communication: Pass Logical Components to Communication Layer

Communication:

Modeling Steps: Provided Services → Communication Infrastructure

Communication → Information: Pass Services & Infrastructure to Information Layer

Communication:

Modeling Steps: Information Exchange → Data Model Standards

Information → Function: Model Transformation Logical Components to Function Layer

Function:

Modeling Steps: Functional Architecture → Black- & White-Box Functions

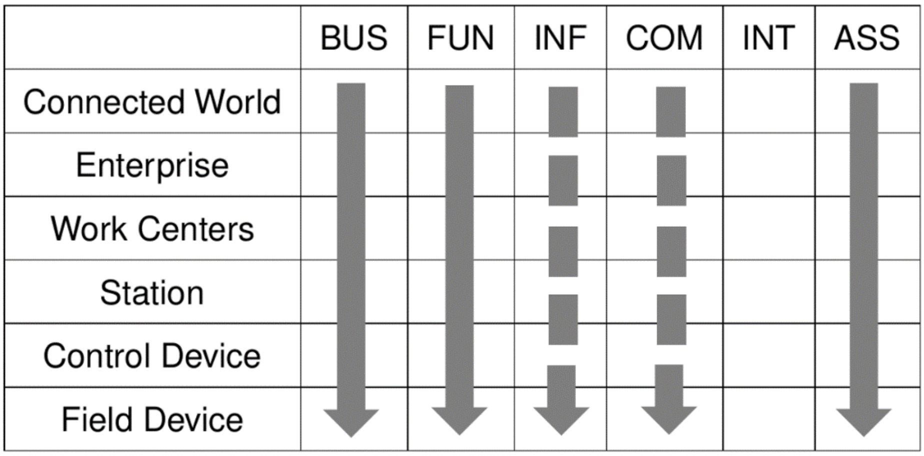
Function → Business: Pass Functions to Business Layer

Business:

Modeling Steps: Requirements

Component Refinement:

From top to down, the context is further refined from factory to workstation up to a single machine. With this method, dependencies between requirements, functions or components might be decomposed into lower granularity levels.



Product/Connected World → Enterprise:

Derive Value Creation Process from Product Development

Pass Process Steps to Enterprise

Enterprise → Work Unit / Work Unit → Station / Station → Control Device / Control Device → Device:

Business: Refinement of Requirements

Function: Refinement of Functions

Asset: Refinement of Assets

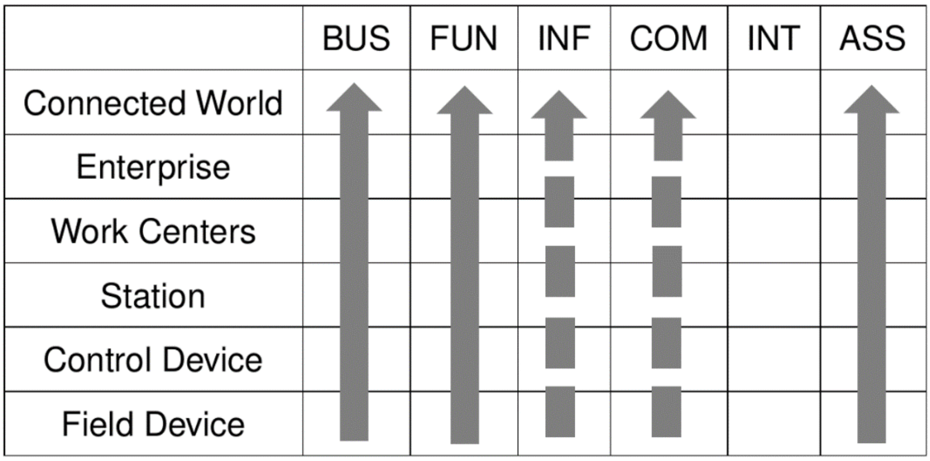
(Information: Refinement of Data Management Components)

(Communication: Refinement of Network Elements)

Pass the higher granularity element as black-box to the lower level.

Factory Integration:

From bottom to up, the existing machines can be integrated within a factory. This allows to compose similar or interconnected system elements to a higher hierarchy level.



Device → Control Device / Control Device → Station / Station → Work Unit / Work Unit → Enterprise:

Business: Summarize similar Requirements an create superior one.

Function: Summarize similar Functions an create superior one.

Asset: Summarize similar Assets an create superior one.

(Information: Summarization of Data Management Components)

(Communication: Summarization of Network Elements)

Pass the lower granularity element as white-box to the higher level.

Enterprise → Connected World:

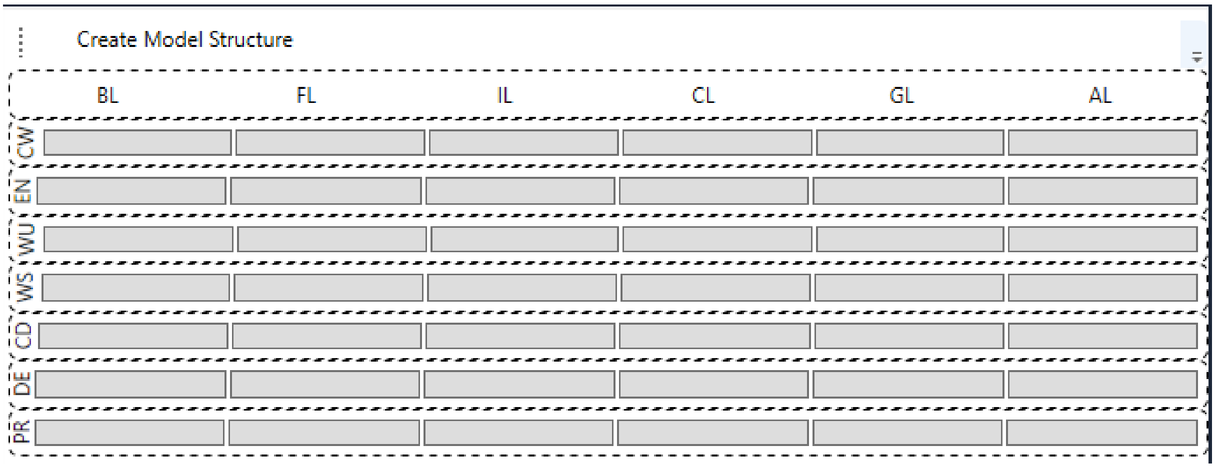
Pass Functions to Connected World

Derive Value Creation Process from Functions

6. The RAMI Toolbox

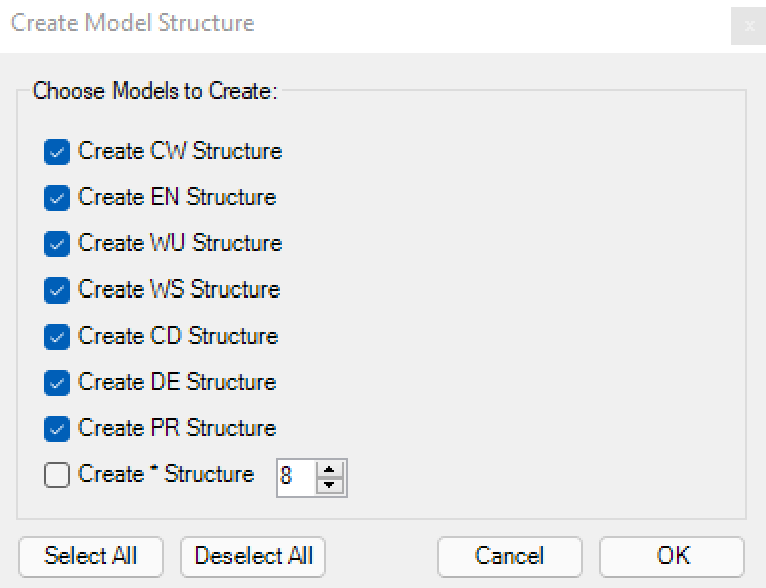
This section gives an overview how the RAMI Toolbox works to navigate within instantiated models created by the architecture modeling framework. Thereby, a GUI and a corresponding event handler listens to the chosen elements and diagrams and navigates to the interconnected pane.

Main GUI Window:

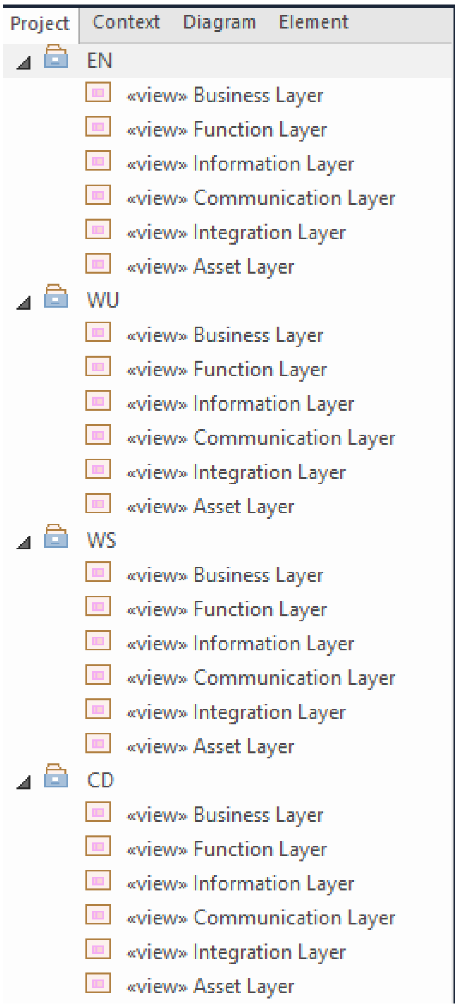


The main windows shows all of the panes within the Matrix.

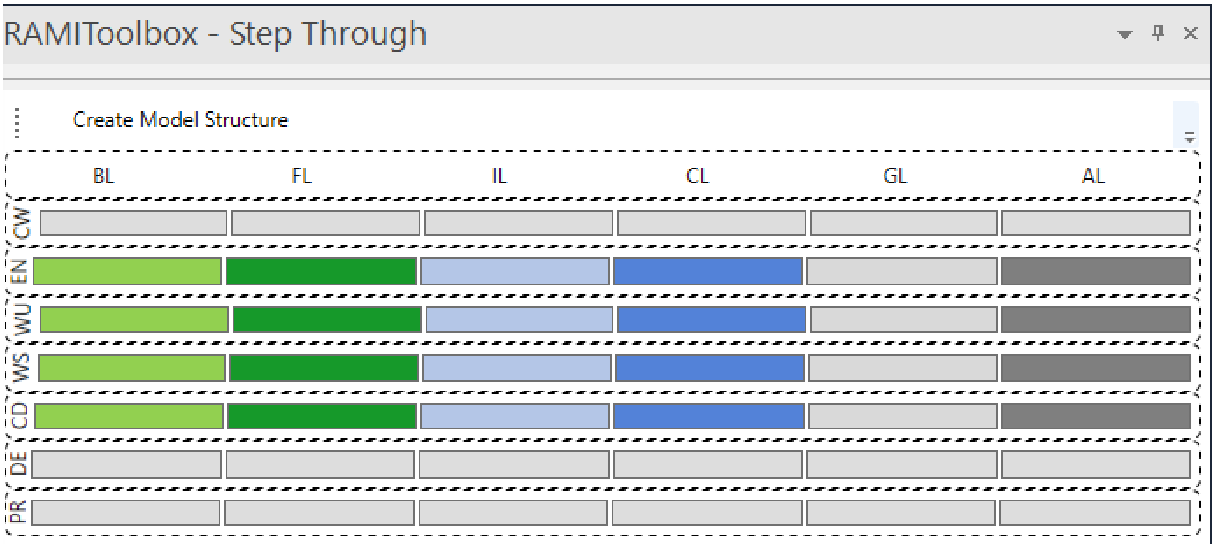
Create Model Structure:



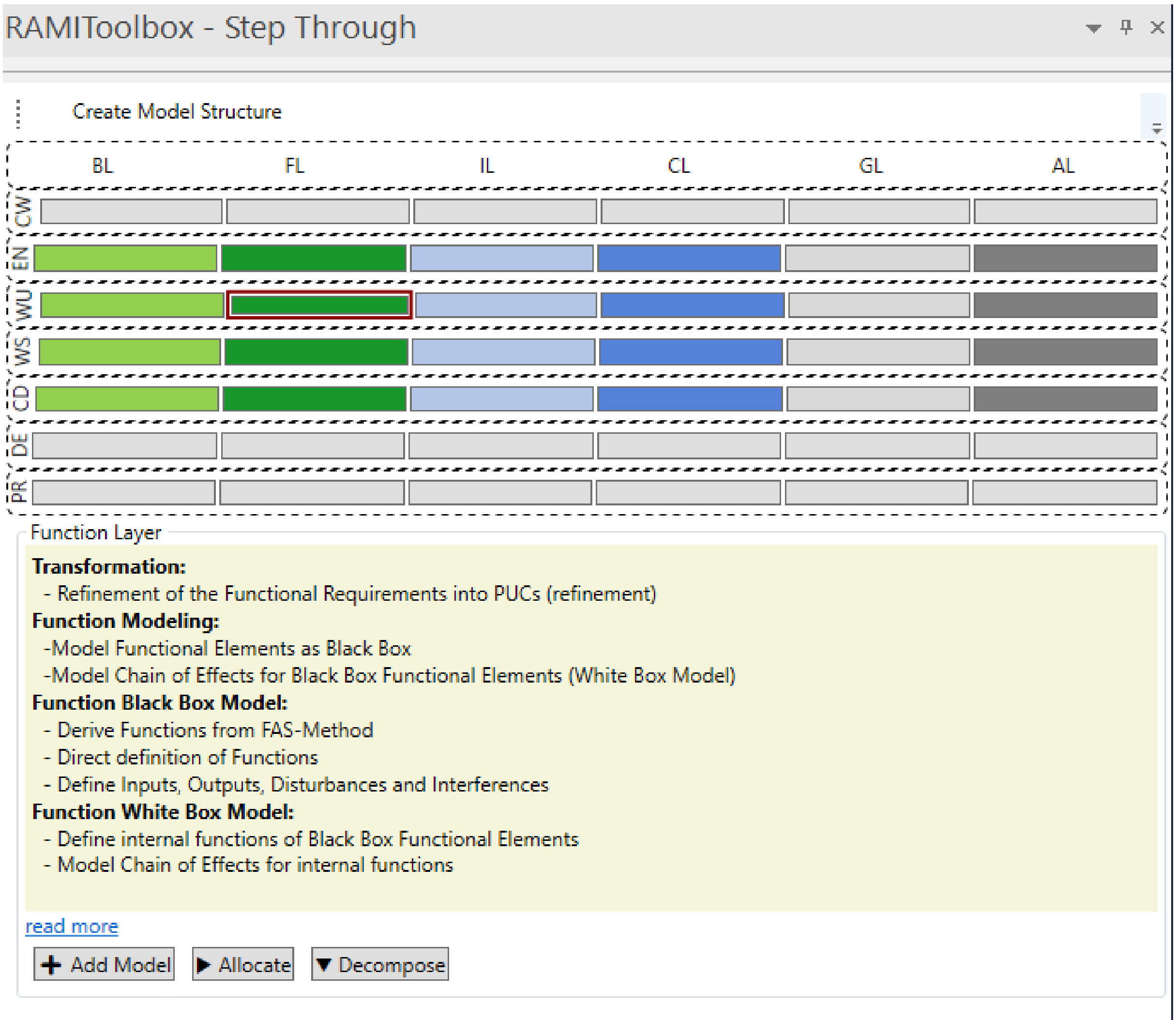
The create Model Structure allows to create the Folder Structure within an EA-Project:



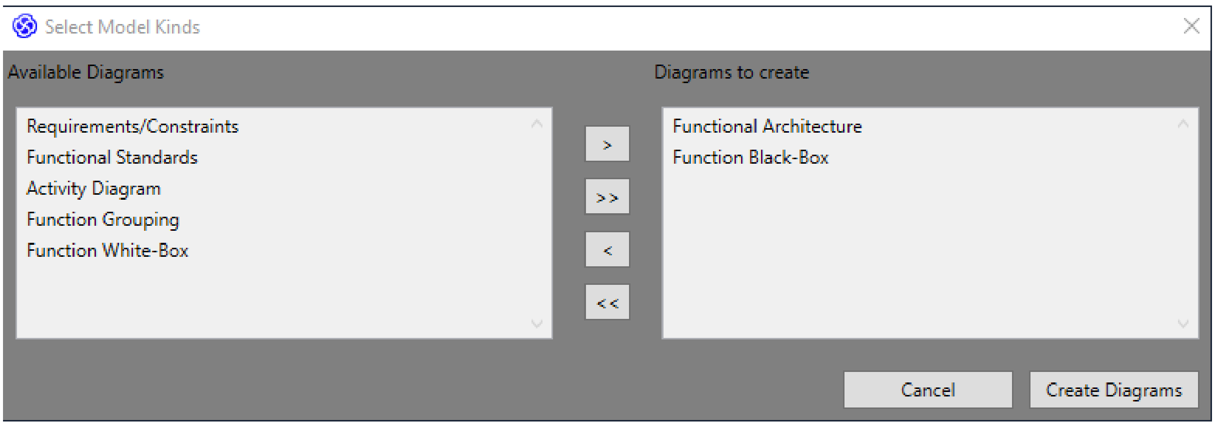
Now, in the GUI, only the selected panes are shown in color:



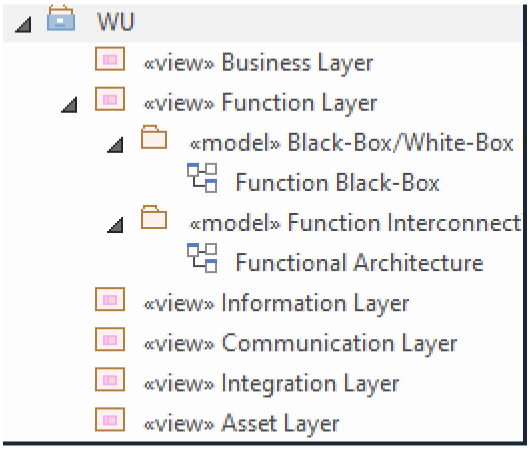
If you click on each pane, additional Functions are offered:



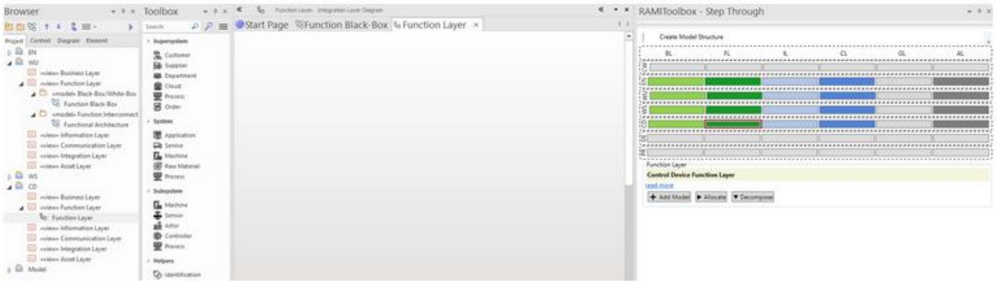
Clicking on Add Model, a set of available diagrams are shown to describe the View.

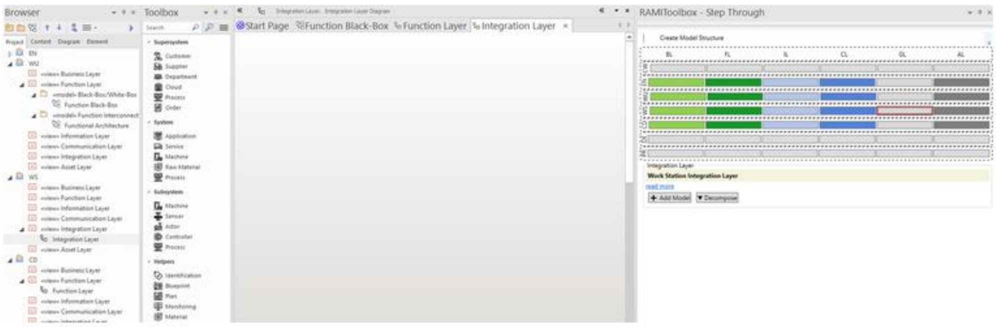


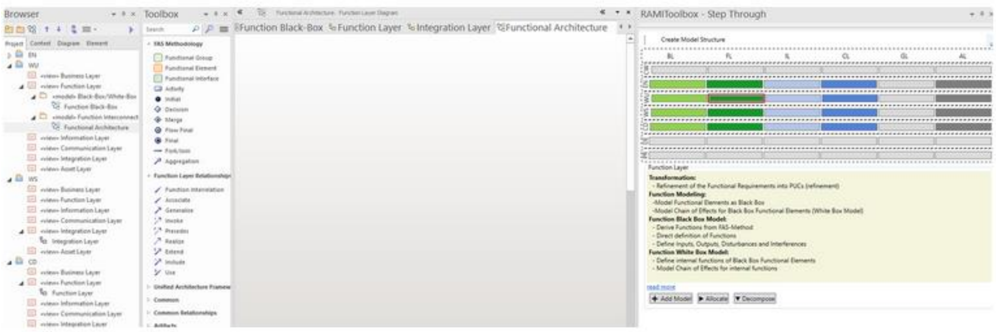
They are automatically created within the Folder structure:



When opening a diagram or switching to it in the tabs, the corresponding pane is automatically selected:







7. Model-driven development with the RAMI Toolbox

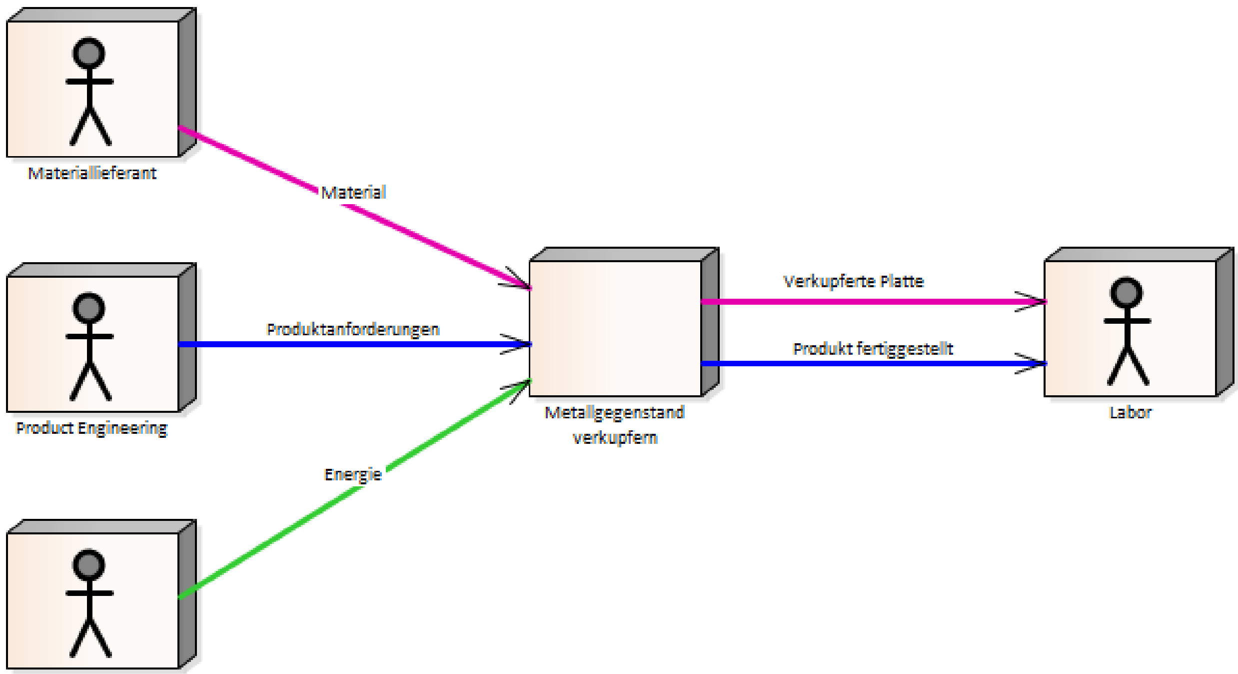
This section explains the application of the RAMI Toolbox based on a practical example. This example deals with model-driven engineering of a Greenfield. However, there are multiple other scenarios available with the matrix of the toolbox. For further information, contact the authors.

7.1. Develop Business Layer

The focus of the RAMI Business Layer is to show the system as it is and find potential for optimizing current processes. Thus, first the Business Cases need to be found and the requirements have to be derived. It is suggested to use the involved High-Level Use Cases as model elements to show realizations for a Business Case. The steps below describe how to model the RAMI Business Layer for a single Business Case.

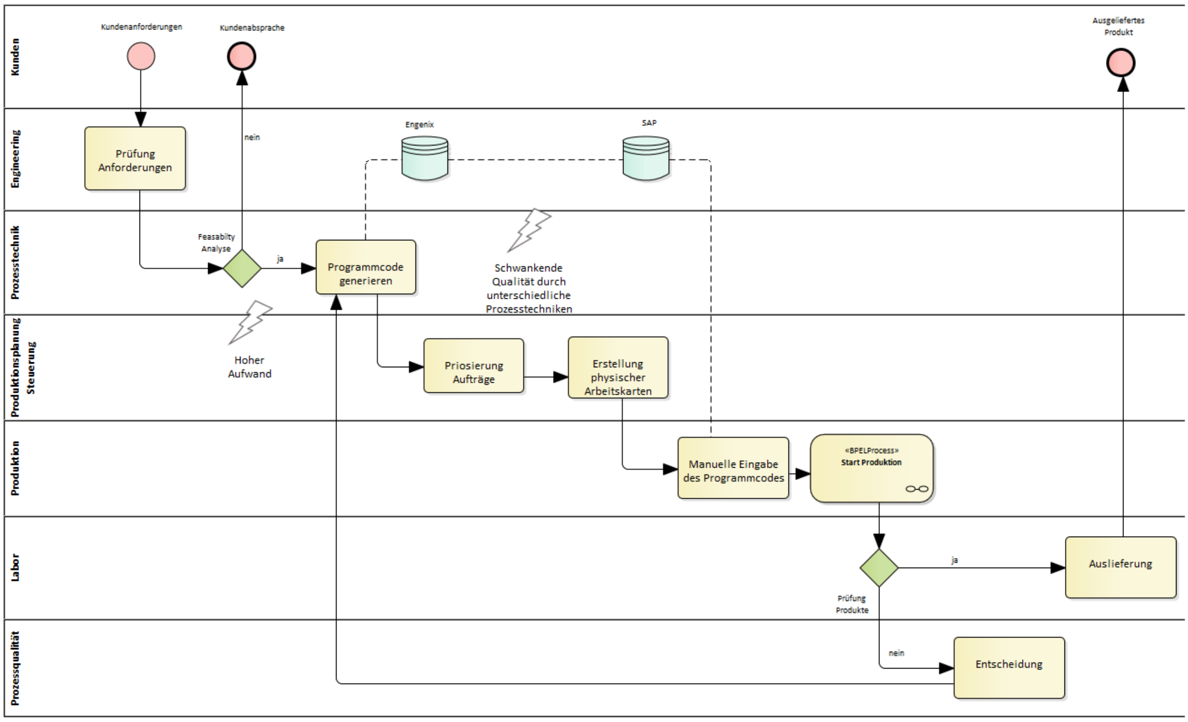
1. *Indicate what is send into your system of interest and what is given to the customers with a SIPOC diagram.*

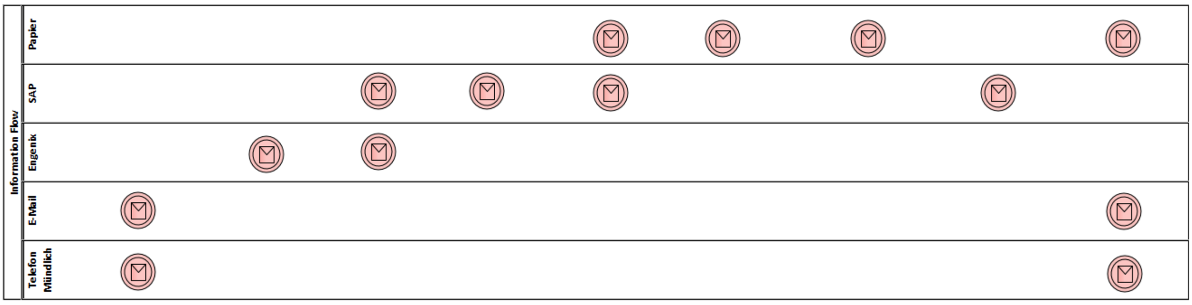
- Create a new Package called Business Layer
- Create a Sub-Package called Context Analysis
- Add a RAMI Context Diagram
- Model the In- and Output from your System of Interest



2. *Use a BPMN Diagram for describing the Process how the received products are converted to goods and send to the customer within the SoI.*

- Right click on your System of Interest and add a new Child Diagram -> Composite Structure Diagram
- Add a Sub-Package called Process Model and emerging BPMN Diagram there
- Model the Process of your SoI with this BPMN Diagram

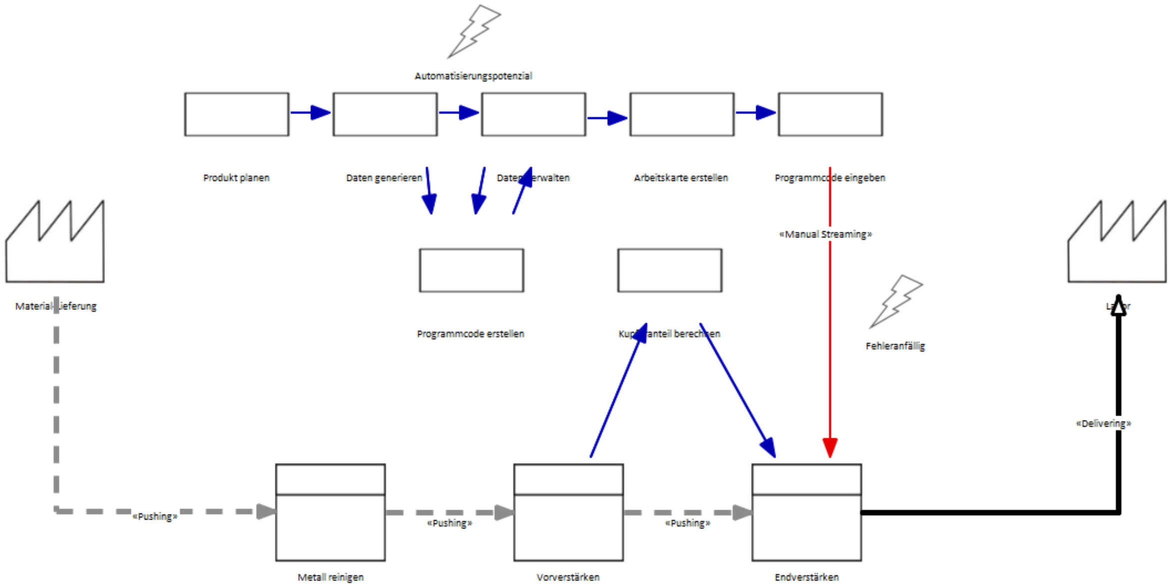




- Use different abstraction levels to describe Business Processes on different abstraction levels
- Then, use Kaizen Bursts for indicating possible problems or potential for optimization

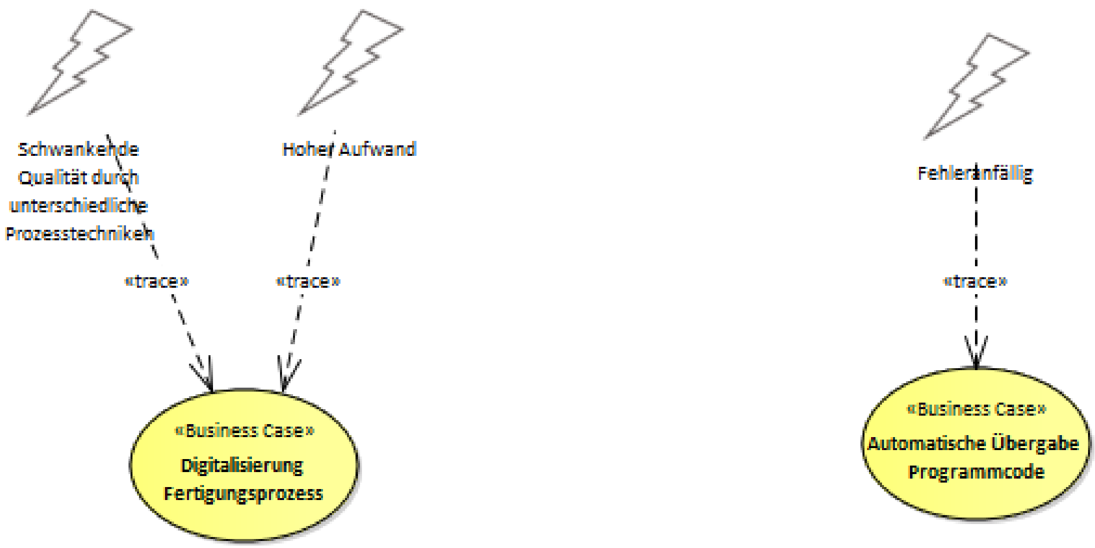
3. *In addition, single manufacturing processes can be modeled with the help of a Wertstromanalyse-Diagram, which is used for better illustrating the engineering view.*

- If you want to describe a single manufacturing process within the Business Process in detail, you can add a new Wertstromanalyse-Diagram and select it as Child Diagram



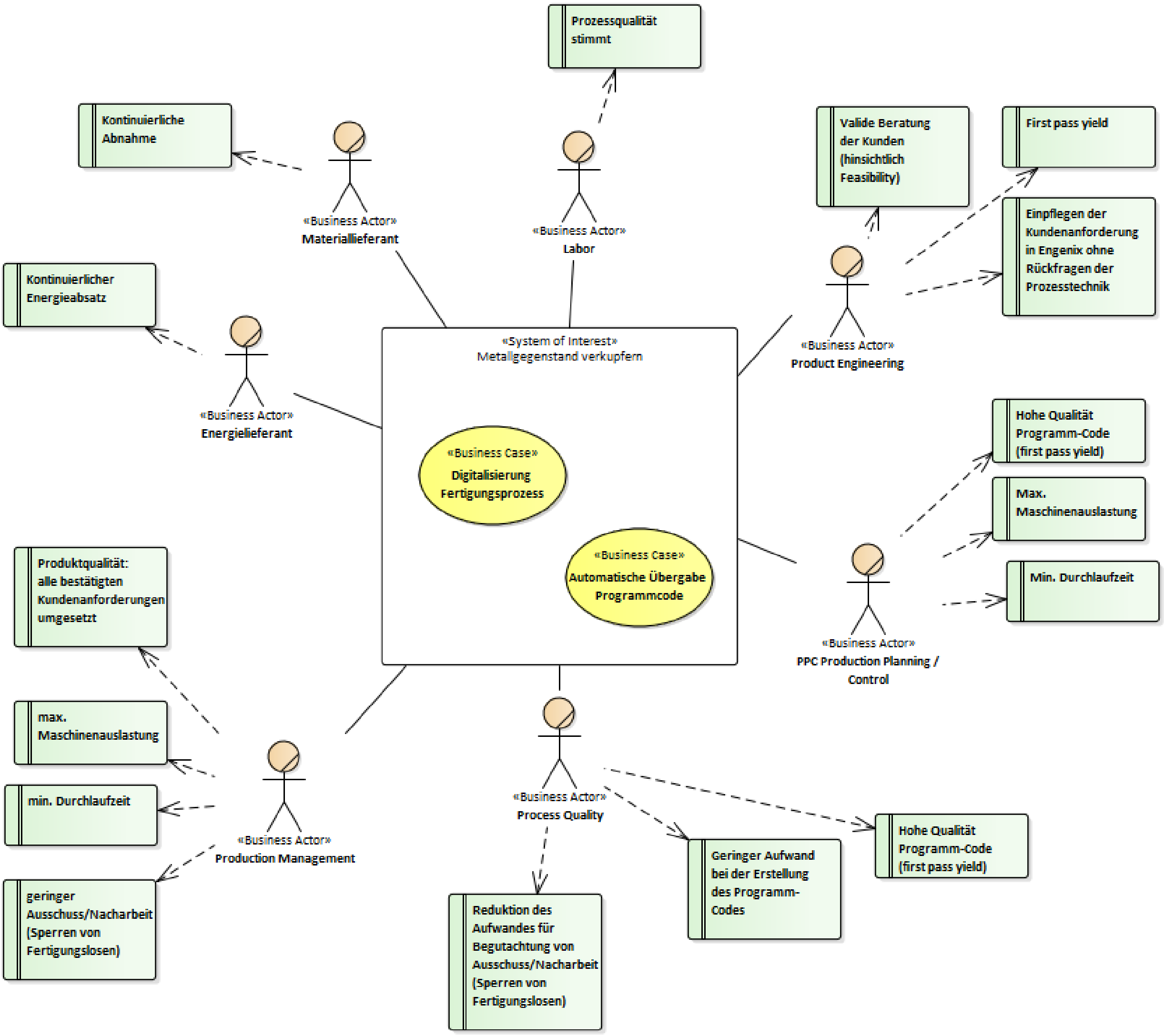
4. *Gather all Kaizen Bursts and summarize related ones into Business Cases. A Business Case thereby explicitly needs economic assessment.*

- Create a new Sub-Package called Business Analysis
- Add a RAMI Business Layer Diagram and move all Kaizen Bursts there
- Consider Business Cases regarding to the identified problems



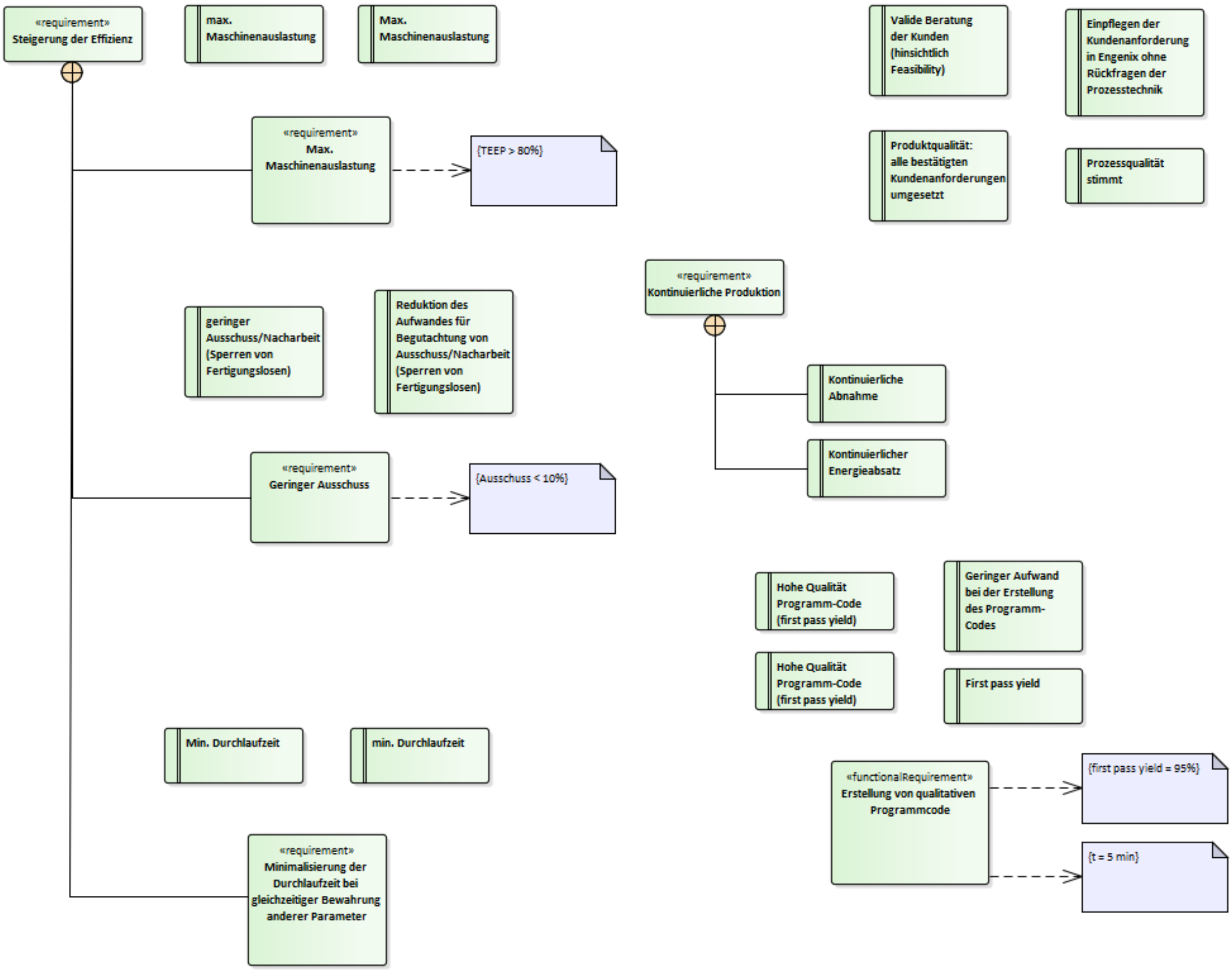
5. *Model the Stakeholders and their interest into the whole manufacturing system as well as for the chosen Business Cases*

- Add a new RAMI Business Layer Diagram into Business Analysis called Goal Model
- Create a Boundary for your SoI and move the Business Case in there
- Add all Stakeholders having interest into the System or the Business Cases

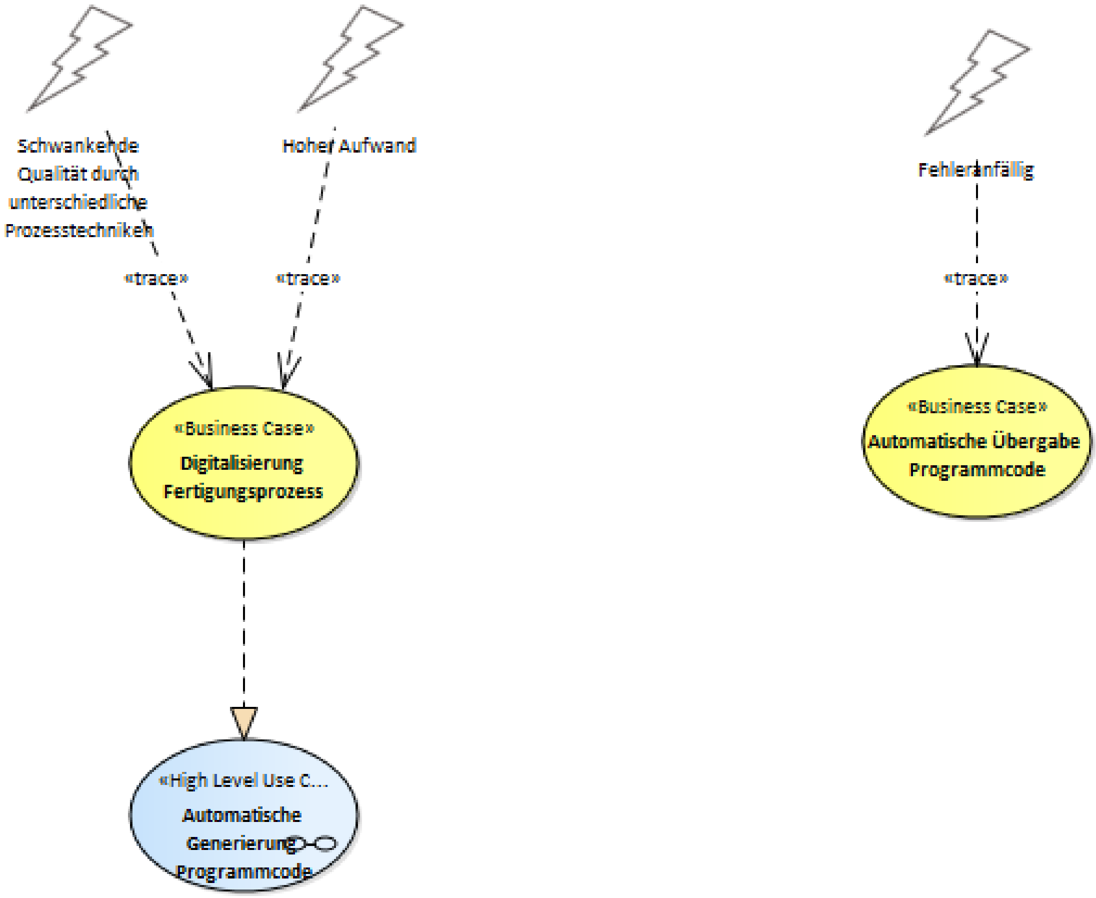


6. Stakeholder needs are summarized to quantitively Requirements

- Create a new Sub-Package called Requirements Analysis
- Add a SysML Requirements Diagram
- Derive Requirements from the Stakeholder Needs or any other elaborated requirements and indicate them in this diagram
- Also use quantitative information where possible



7. *On basis of the identified requirements, consider possible ways to approach the solving of the Business Case. Determine a specific solution and model it as High Level Use Case in the Business Case Diagram.*



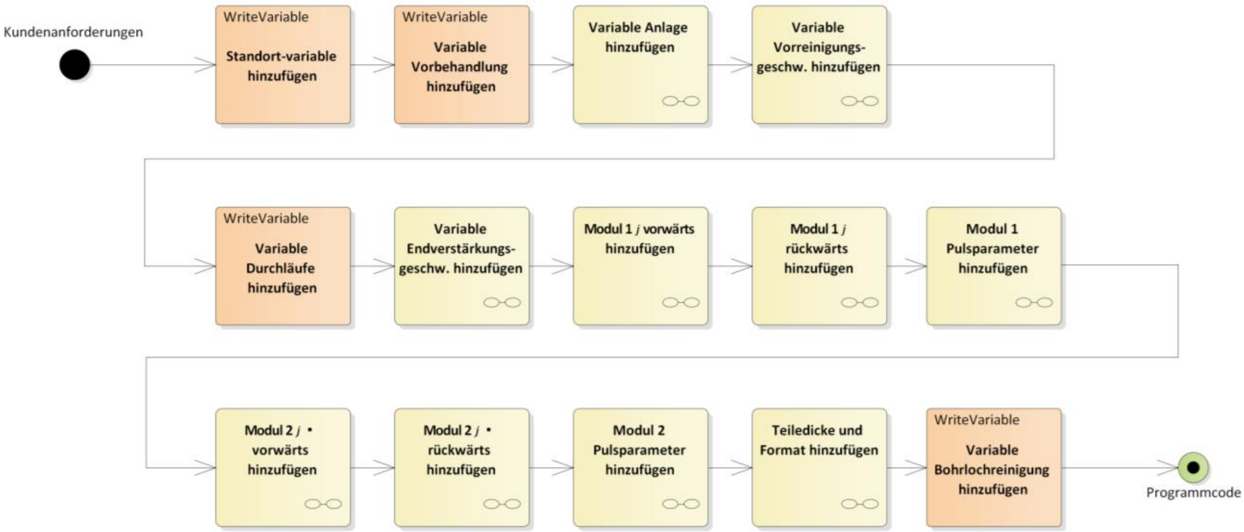
A High Level Use Case builds the base for further system development. All future aspects are based on the outcomings of this Business Analysis.

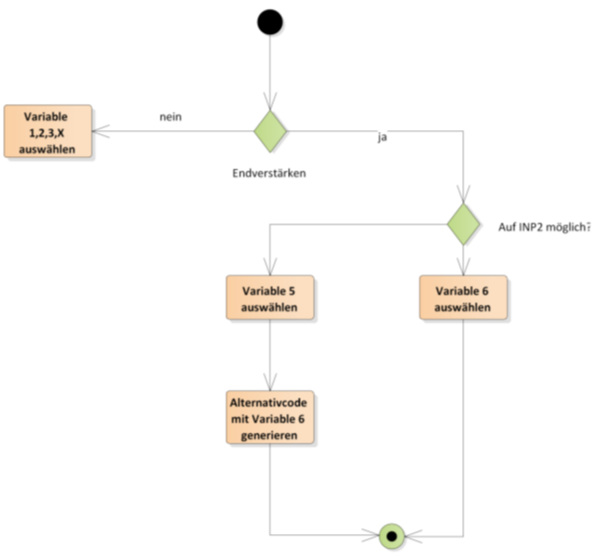
7.2. Develop Function Layer

The RAMI Function Layer deals with developing functions in order to fulfil the requirements and specify realization elements that execute the identified functions.

1. Model the HLUC in detail with an Activity Diagram on multiple granularity levels

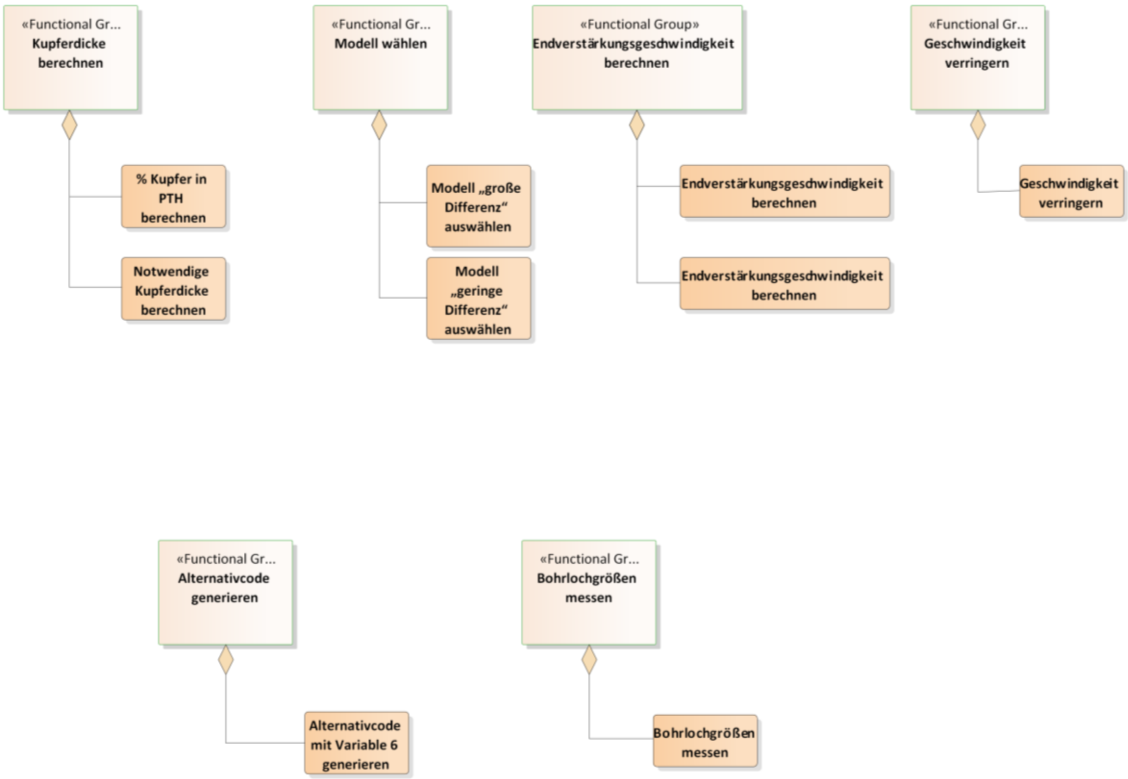
- Create a new Package called Function Layer
- Create a Sub-Package called Use Case Refinement
- Right click on your HLUC and add a new Activity Child Diagram
- Model the intended Process how the HLUC should be executed
- Use different abstraction levels and be as detailed as possible





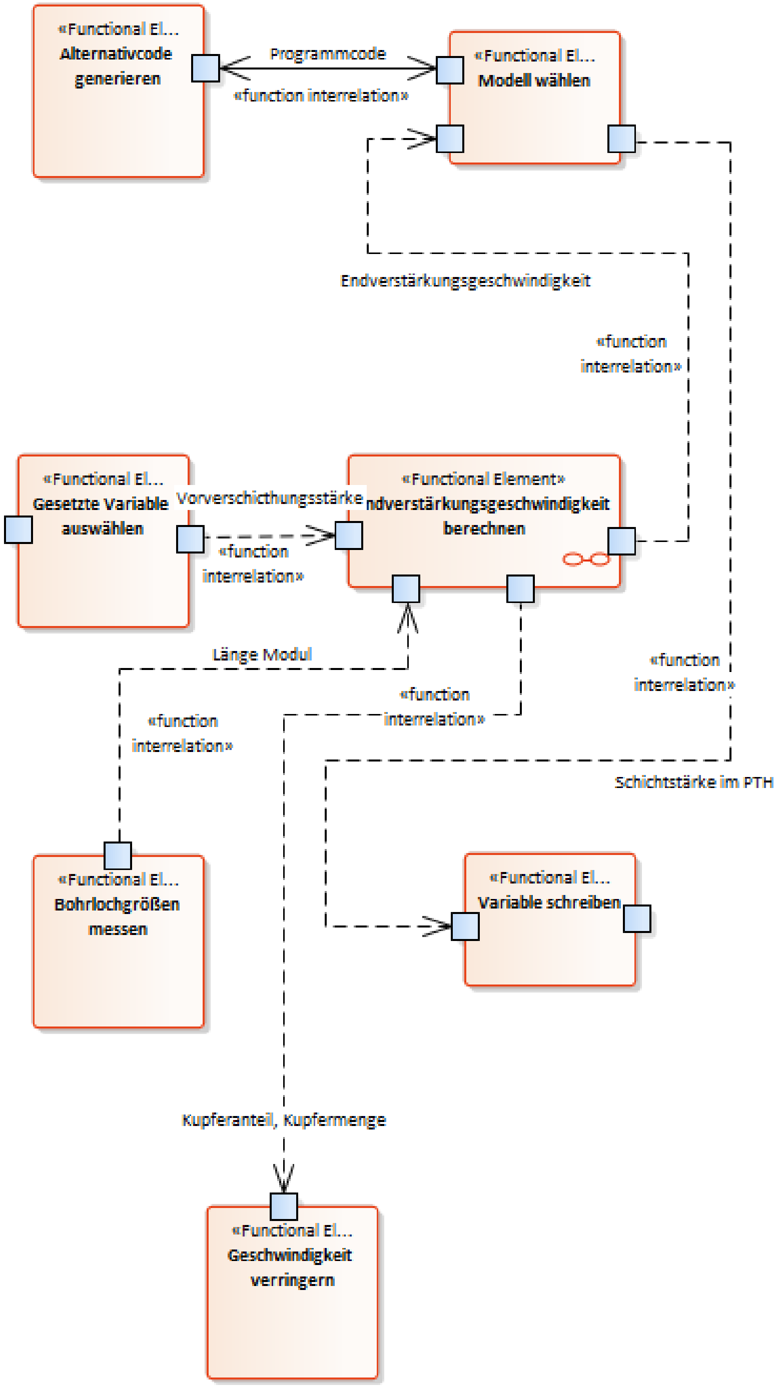
2. *Summarize Identical Actions and map them to Functional Groups*

- Create a Sub-Package called Functional Grouping
- Add a new RAMI Function Layer Diagram
- Move all granular Actions or Tasks into this Diagram
- Identify similar Actions and create Functional Groups describing them in more detail



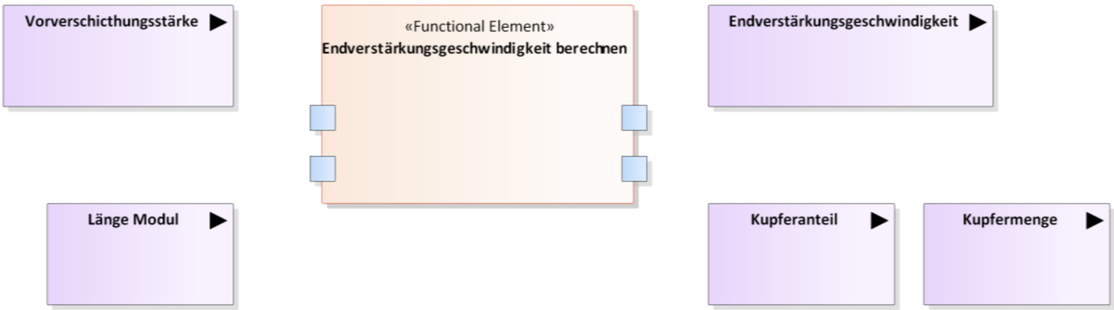
3. *Develop Functions from Groups and model their interconnection as well as transmitted elements*

- Create a Sub-Package called Functional Layer
- Add a new RAMI Function Layer Diagram
- Create a Functional Element for each Functional Group
- Add them to the Diagram and specify Ports for In- and Outputs
- Model the Function Interrelations between the Functional Elements



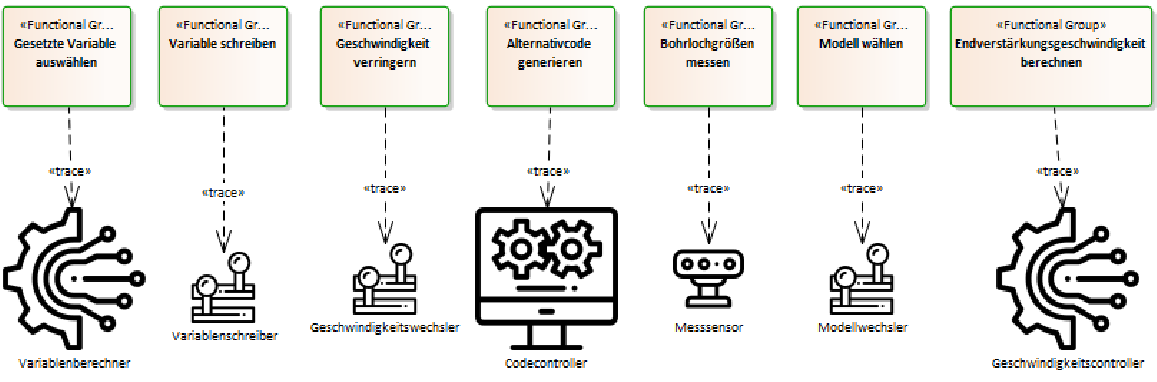
4. *Model each Function in detail according to Black- and White-Box perspective*

- Create a new SysML Block Definition Diagram for each Functional Element and move it beyond the element in the package explorer
- Model the Function itself, interfaces and ports in more Detail by showing In- and Outputs as well as disturbances and interferences



5. *Trace Functions to Logical Elements*

- Create a Sub-Package called Actor Mapping
- Add a new RAMI Integration Layer Diagram
- Move all Functional Groups into this Diagram
- Add physical components fulfilling the single Functions to this Diagram (n:n relationships)

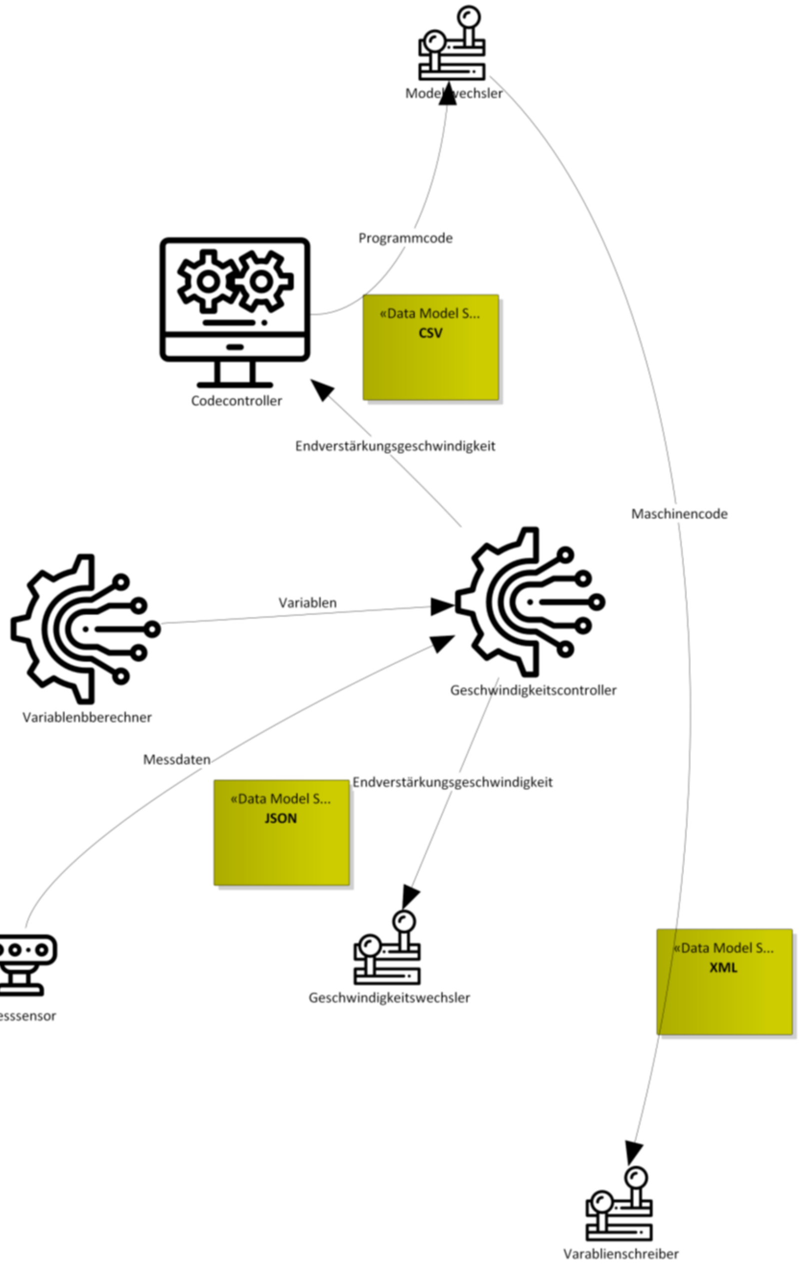


7.3. Develop Information Layer

Focus of this task is to model the information object flows between the single components and to identify proper data model standards that are suitable to reflect these information objects.

1. Model Data exchange between Logical Components and specify Data Model Standards

- Create a new Package called Information Layer
- Add a new RAMI Information Layer Diagram
- Move all logical elements into this Diagram
- Model the information exchange and all related data in this Diagram

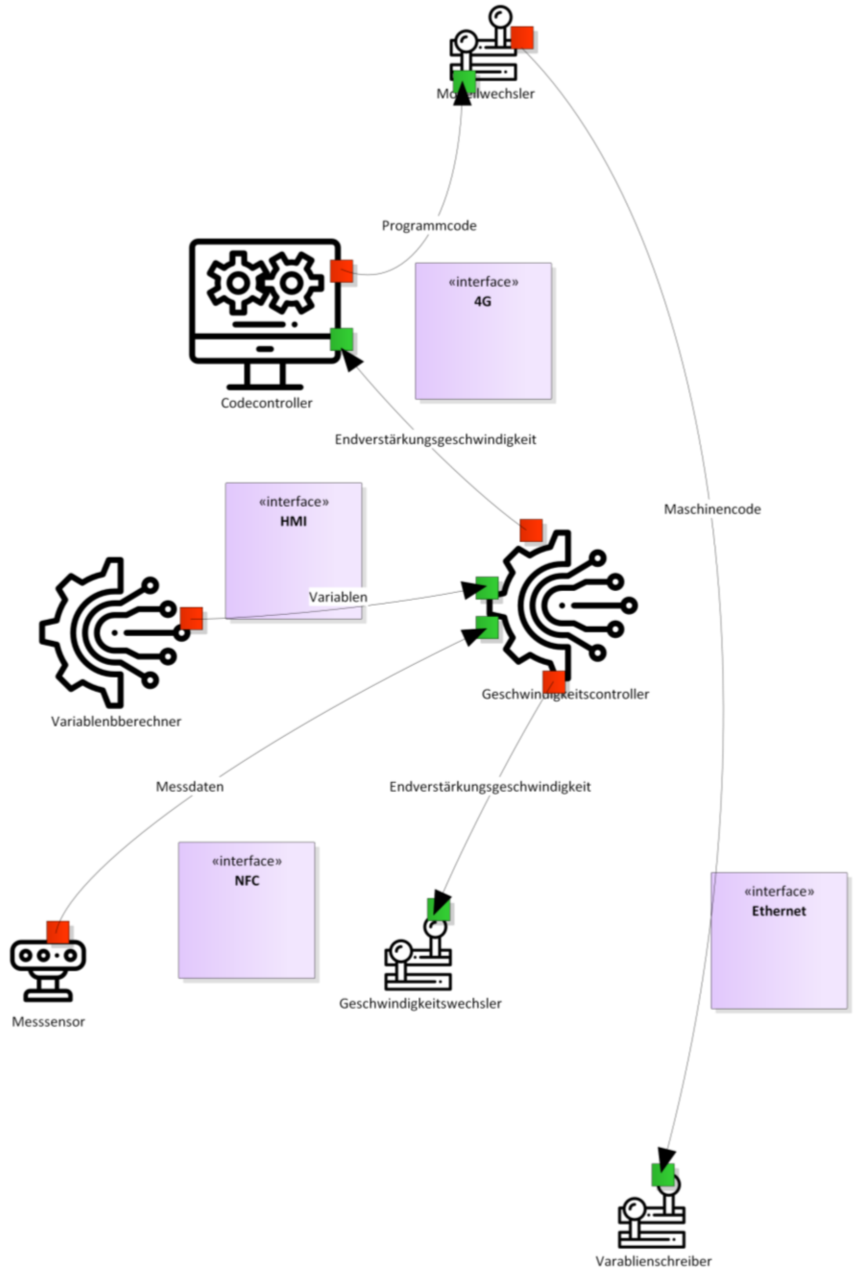


7.4. Develop Communication Layer

The focus of the Communication Layer is to depict the used technology and protocols for the communication between single components on basis of a Service-oriented Architecture.

1. Model Interfaces on which the data is exchanged.

- Create a new Package called Communication Layer
- Add a new RAMI Communication Layer Diagram
- Move all logical elements into this Diagram
- Model the communication structure and all interfaces in this Diagram

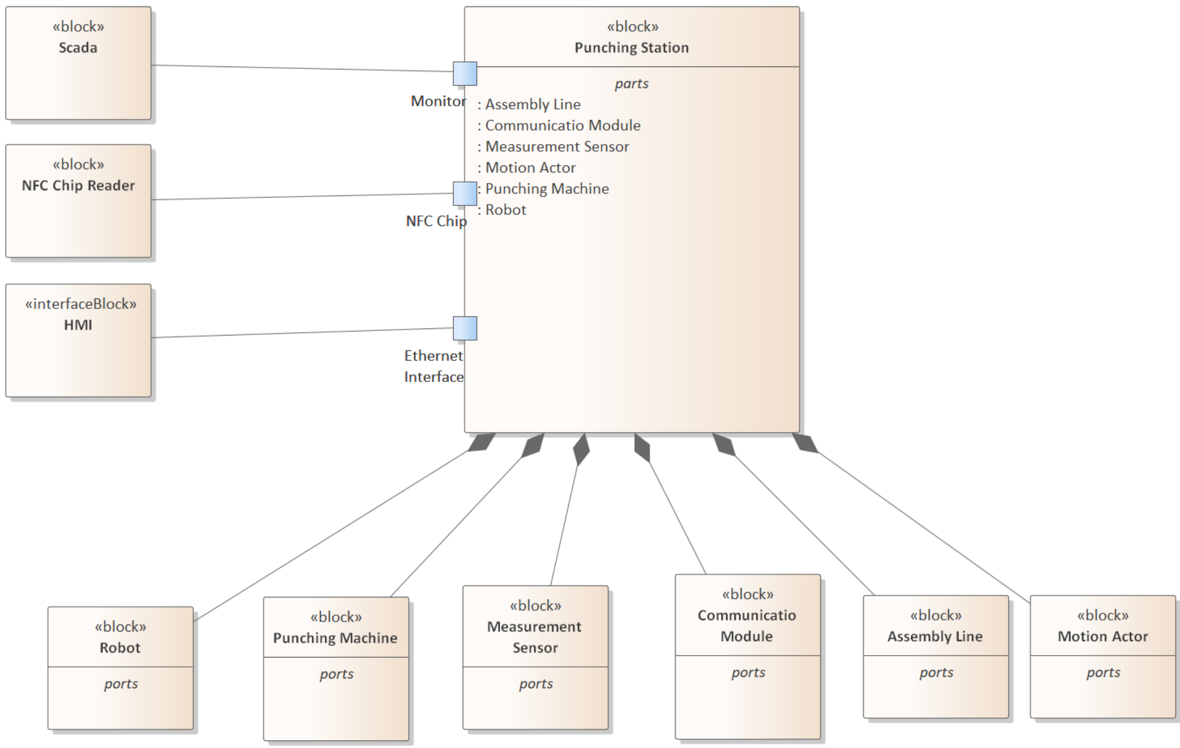


7.5. Develop Integration Layer

By developing the RAMI Integration Layer, the ICT Infrastructure and the HMI's of the Logical Elements are added to the Model.

1. Model ICT Network Topology and HMIs to create a digital twin of an asset.

- Create a new Package called Integration Layer
- Add a new RAMI Integration Layer Diagram
- Move all logical elements into this Diagram
- Model the dependencies and all HMI`s in this Diagram

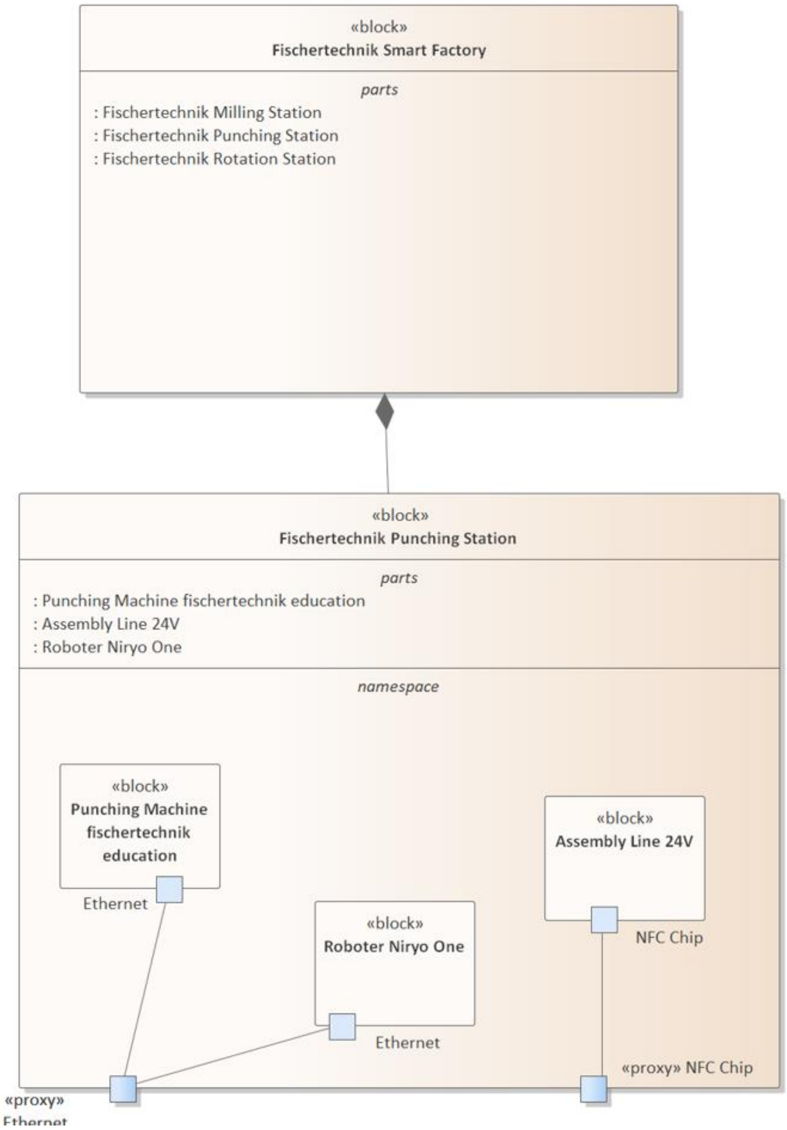


7.6. Develop Asset Layer

By developing the RAMI Asset Layer, real-world representation is added to the model.

1. *Model physical components and their dependencies of the system.*

- Create a new Package called Asset Layer
- Add a new SysML Block Definition Diagram
- Create all physical elements based on logical ones into this Diagram
- Model the dependencies in this Diagram



8. Case Studies & Applications

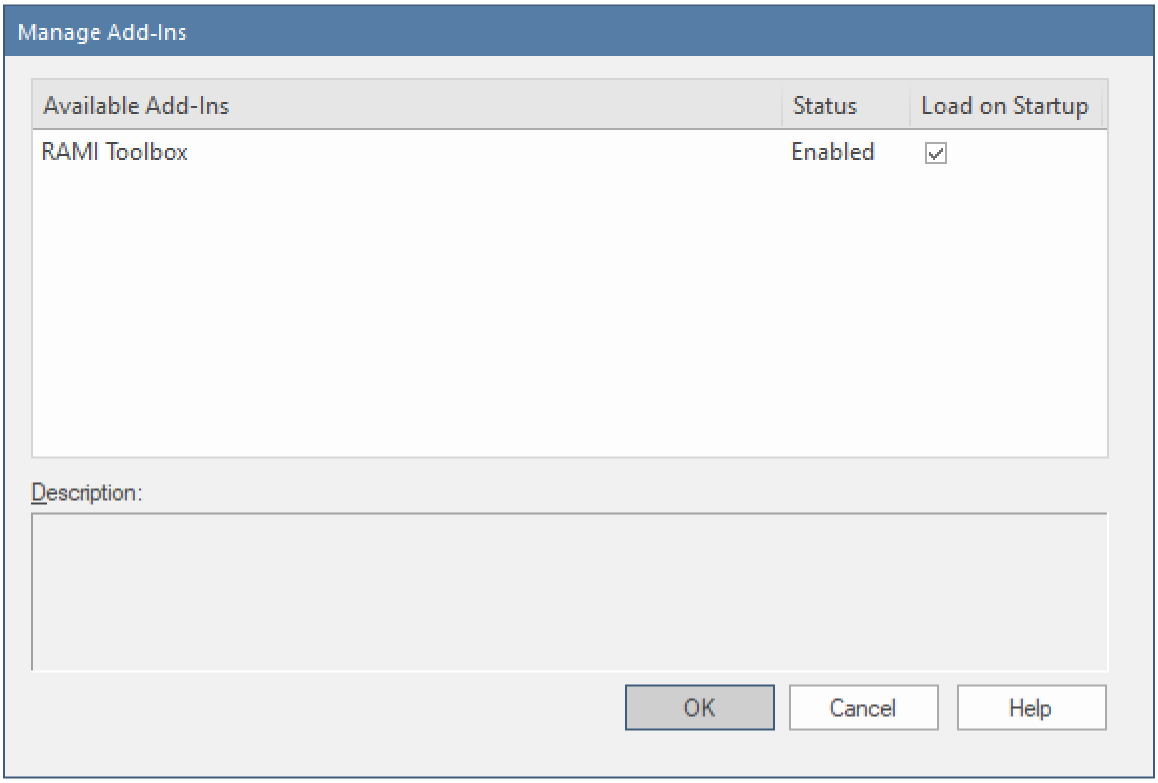
8.1. PIM -> PSM fully automated model transformation

1. Install the corresponding Version of the RAMI Toolbox

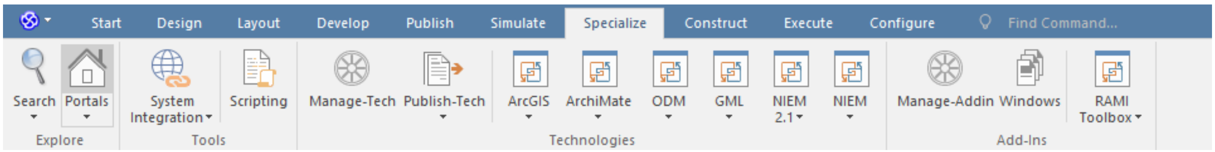
Make sure, the RAMI Toolbox is installed correctly and available in Enterprise Architect. The source files are located in:

C:\Program Files (x86)\RAMI Toolbox

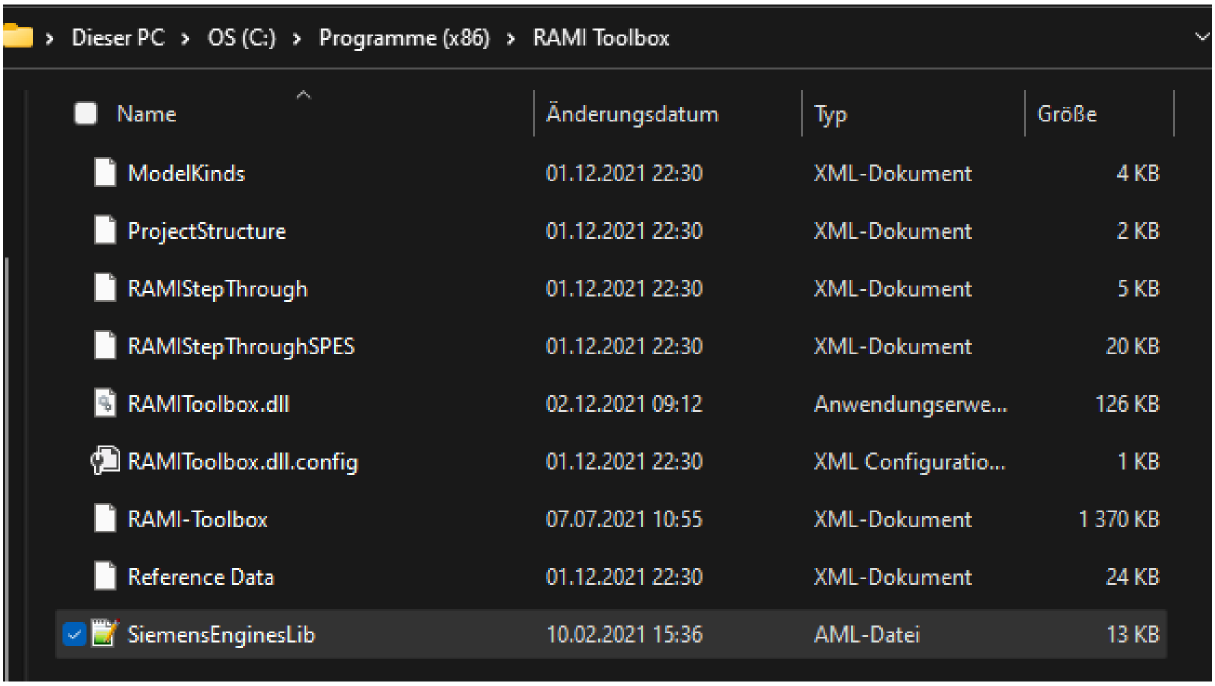
The availability of the RAMI Toolbox Add-In can be checked in EA in the “Specialize”-Tab with the function “Manage-Addin”:



On the right hand side, the RAMI Toolbox can be accessed and functions can be called:



2. *Place the AutomationML Library file “SiemensEnginesLib.aml” into the source folder of the RAMI Toolbox*



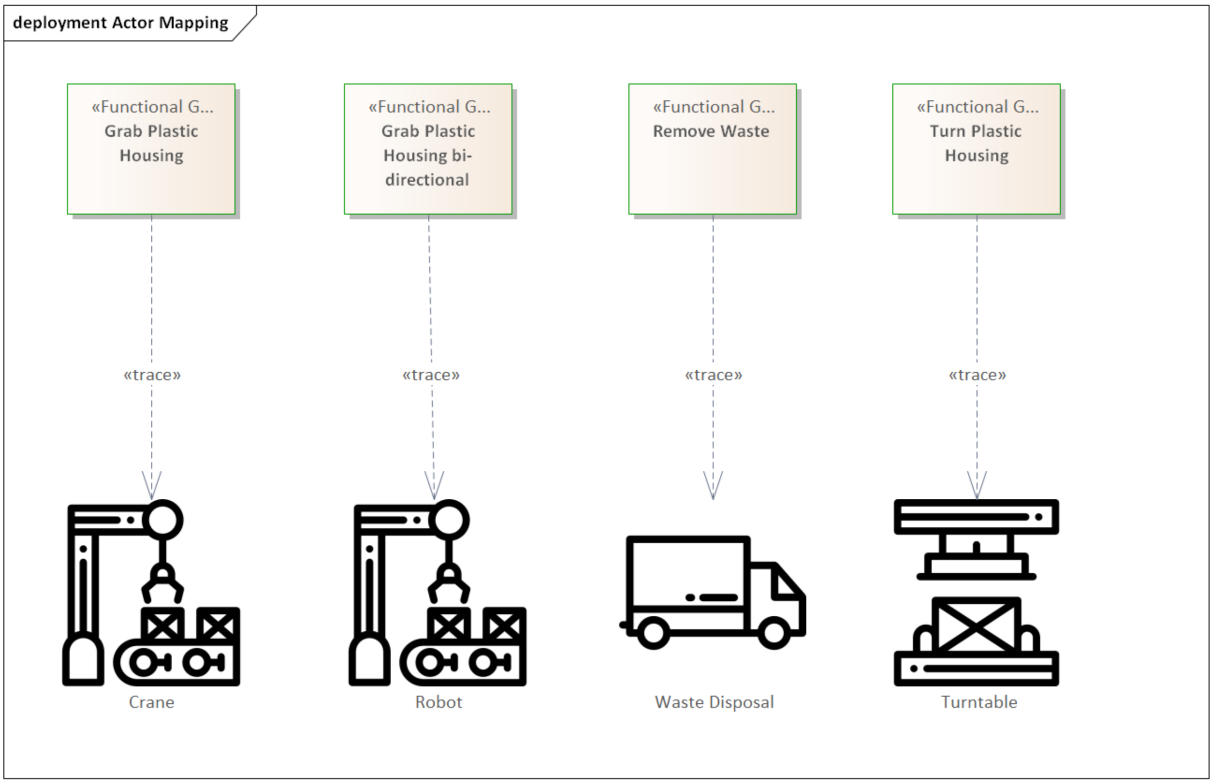
3. *Use the already provided “Fischertechnik” Case Study to examine an example for the development of industrial systems with the RAMI Toolbox or model your own system according to the proposed development strategy*

➔ Further information about this step is mentioned in Chapter 7

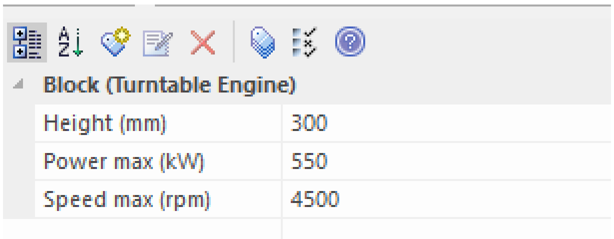
4. *To use the transformation from PIM to PSM, model the Function Layer of RAMI 4.0*

- Refine Use Cases and interactions with the system by applying Activity Diagrams
- Use the FAS methodology to group similar actions
- Create Functional Elements to further refine the grouped actions with a Functional Architecture, Black- and White-Box models
- Model an Actor Mapping Diagram, where the mapping of Functions to physical components is depicted

Example for Actor Mapping Model:



5. *For each Functional Group representing the usage of an engine, add the Tagged Values "Height (mm)", "Power max (kW)" and "Speed max (rpm)"*

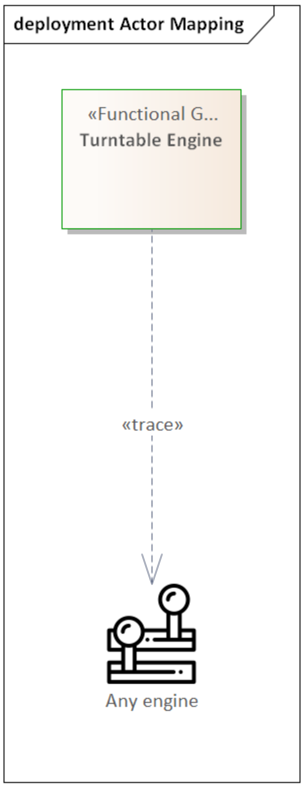


Block (Turntable Engine)	
Height (mm)	300
Power max (kW)	550
Speed max (rpm)	4500

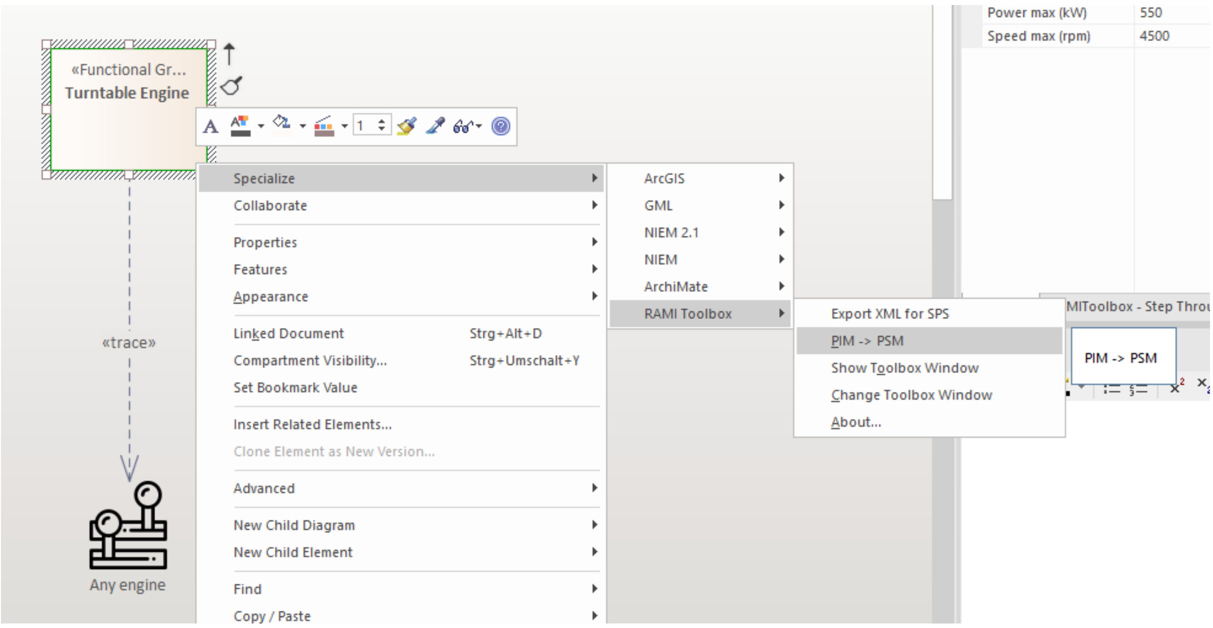
6. *According to the desired Specifications of the Engine, set the values to be strived for*

Either use real values derived from technological system requirements or just put any value for testing purposes. In this scenario, the turntable engine must have a max height of 30 cm, a max power supply of 550 kW and a max speed of 4500 rpm.

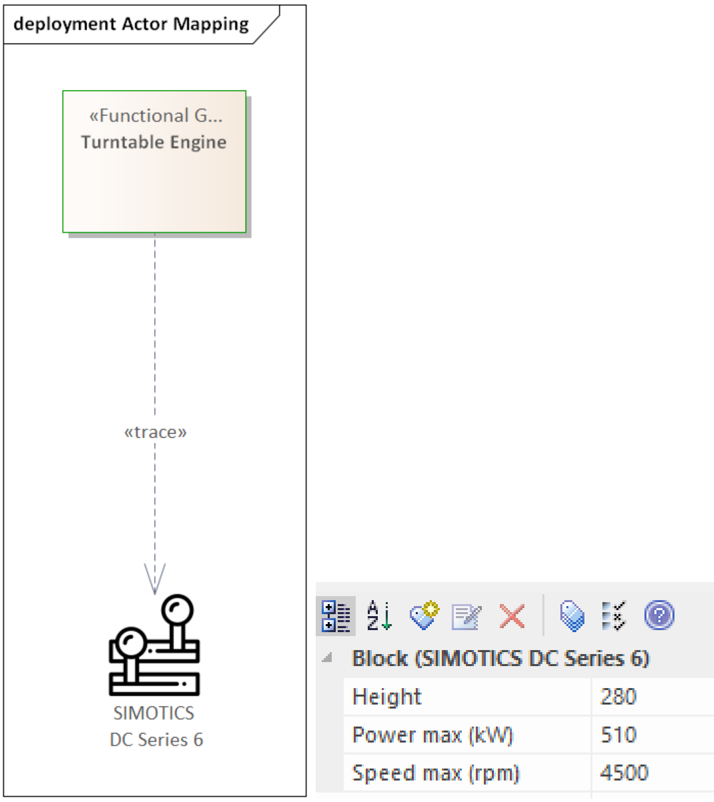
7. *In the Actor Mapping Model, trace any component (ideally an Actor) to the just defined Functional Group*



8. *With a right click on the Functional Group or with the RAMI Toolbox at the Specialize-Tab, you can execute the function "PIM -> PSM".*

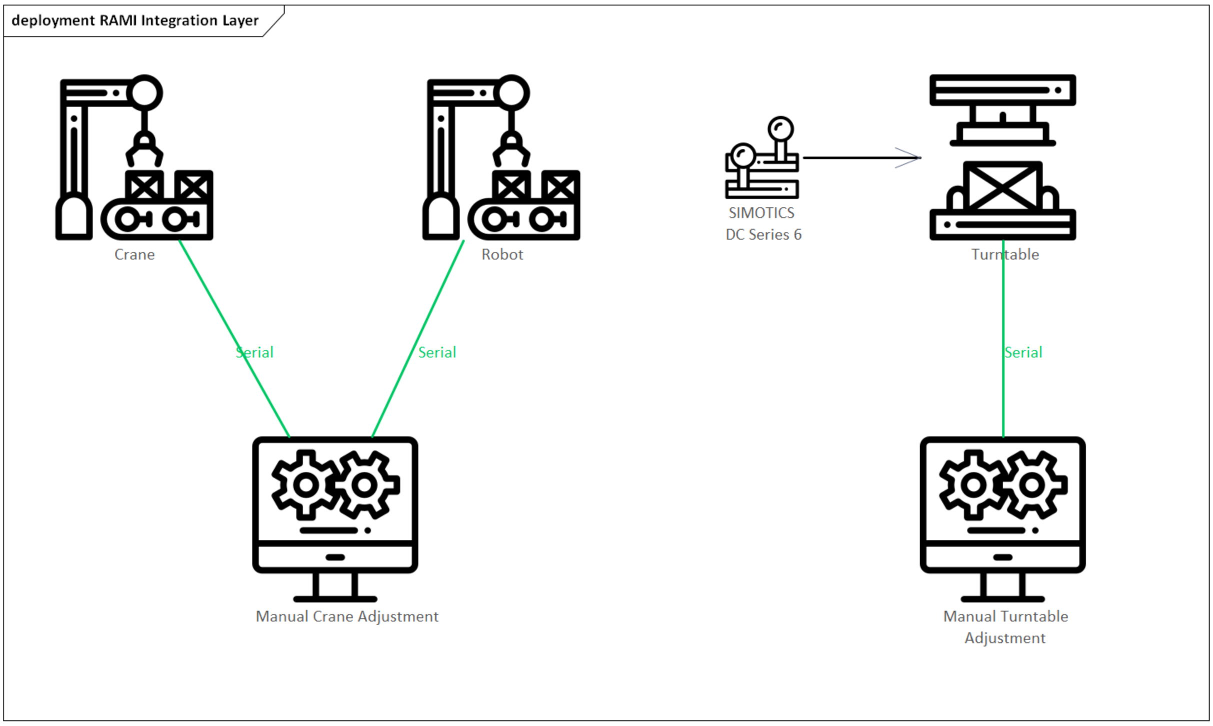


If everything is installed correctly, this will search in the AutomationML library for the best possible solution and automatically sets the Values of the traced component to this solution.



In this case study, the Siemens SIMOTICS DC Series 6 is the best fitting engine with a height of 280 cm, a power supply of 510 kW and a rotation speed of 4500 rpm.

9. Model the technical architecture of the system with the chosen engine



9. Acknowledges and Future Work

The financial support of the Josef Ressel Center by the Austrian Federal Ministry of Economy, Family and Youth and the Austrian National Foundation for Research, Technology and Development is gratefully acknowledged.

As the RAMI 4.0 Toolbox appears to be a very useful help for architecting Industrie 4.0-based Systems continuous development is planned. The features for the next iteration will be selected according to the gained experience during application in various projects. However, a few feature requests already exist as you can find below:

- Architecture Refinements
- Element Attribute Refinements
- Model Verification & Validation
- FMEA Integration
- …