Introduction to the „RAMI 4.0 Toolbox“

Author:	Christoph Binder
Version:	0.1
Date:	2017-06-08

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.

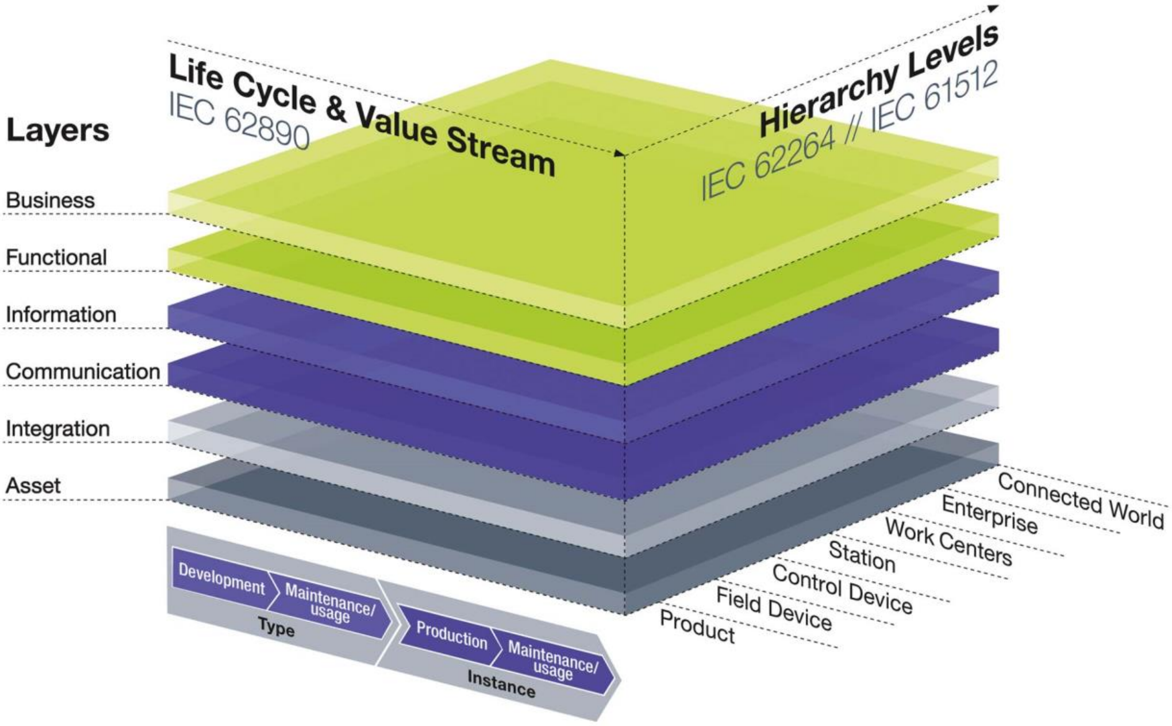


Figure 1 – Referenzarchitekturmodell Industrie 4.0 (RAMI 4.0)

2.1. The Interoperability Layer

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.

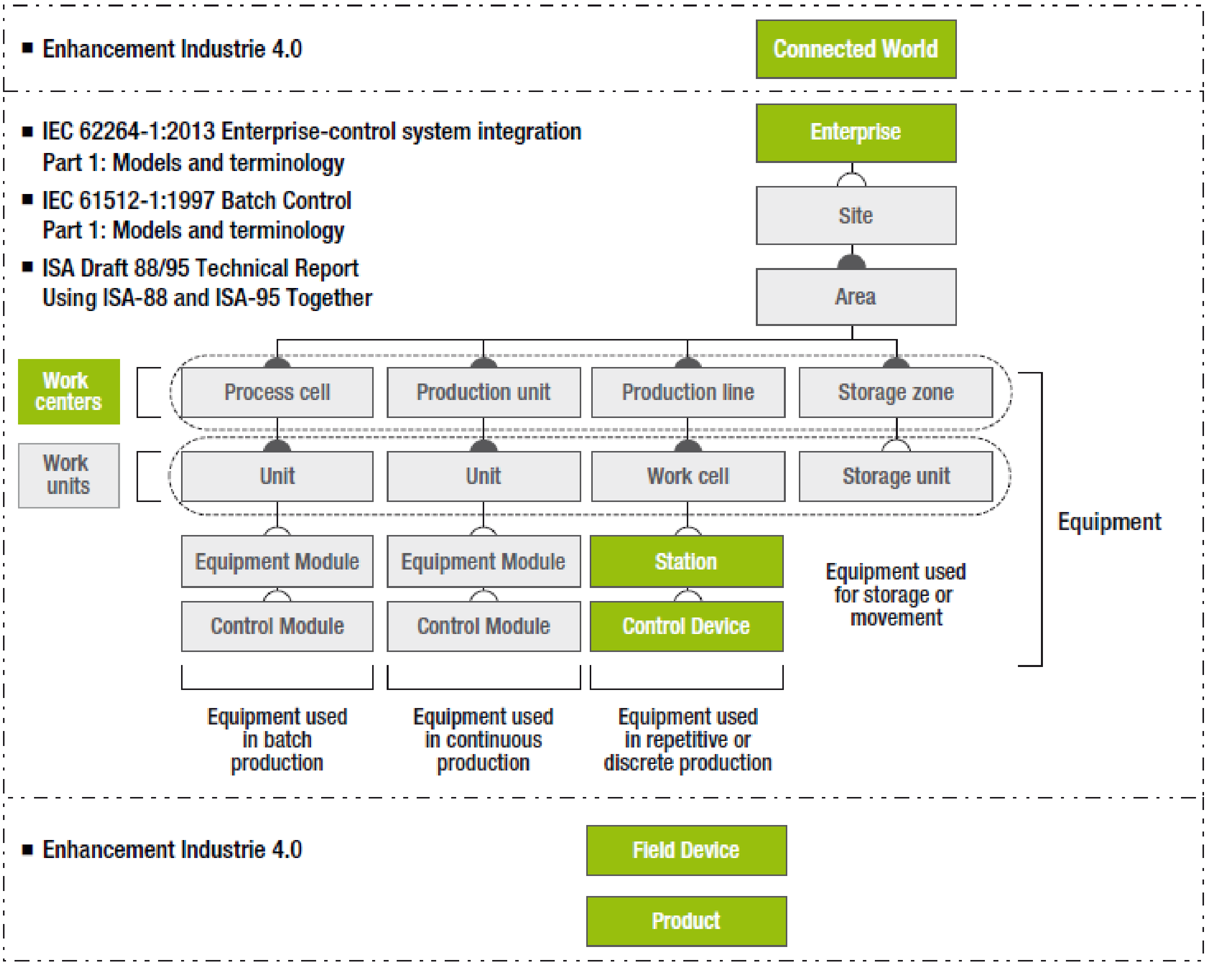
Life Cycle	Description
Type Development	This represents the first idea of a product. At this stage the every aspect around the product is displayed, from commissioning to development, testing and the generation of the first prototypes.
Type Maintenance	Representing the result from the development stage, 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.



■ Enhancement Industrie 4.0
Connected World
■ IEC 62264-1:2013 Enterprise-control system integration Part 1: Models and terminology
■ IEC 61512-1:1997 Batch Control Part 1: Models and terminology
■ ISA Draft 88/95 Technical Report Using ISA-88 and ISA-95 Together
Enterprise
Site
Area
Work centers
Work units
Process cell
Production unit
Production line
Storage zone
Unit
Unit
Work cell
Storage unit
Equipment Module
Equipment Module
Station
Control Module
Control Module
Control Device
Equipment used for storage or movement
Equipment
Equipment used in batch production
Equipment used in continuous production
Equipment used in repetitive or discrete production
■ Enhancement Industrie 4.0
Field Device
Product

Figure 2: RAMI 4.0 Hierarchy Levels

3. RAMI 4.0 Toolbox Architecture

The RAMI 4.0 Toolbox mainly consists of three components:

1. RAMI 4.0 MDG Technology
2. RAMI 4.0 Model Templates
3. RAMI 4.0 Reference Data

MDG Technology
The definition of the elements that can be used to model a RAMI 4.0 related project is done by the use of MDG Technology. The RAMI 4.0 MDG Technology holds these definitions.

Model Templates
To ease the work with the Toolbox some Model Templates have been created. They are based on the above mentioned definitions and can be accessed via the New Model Wizard.

Reference Data
The RAMI 4.0 Reference Data holds some Information concerning the representation of defined elements and some definitions for a Model-Import or -Export.

Metamodel
The most important element of the RAMI 4.0 Toolbox is the RAMI 4.0 Metamodel, which is derived from the RAMI 4.0 and delivers the available elements and their relations.

Figure 3 - The RAMI 4.0 Metamodel depicts the RAMI 4.0 Metamodel as rough overview without mentioning the individual attributes (implemented as Tagged Values).

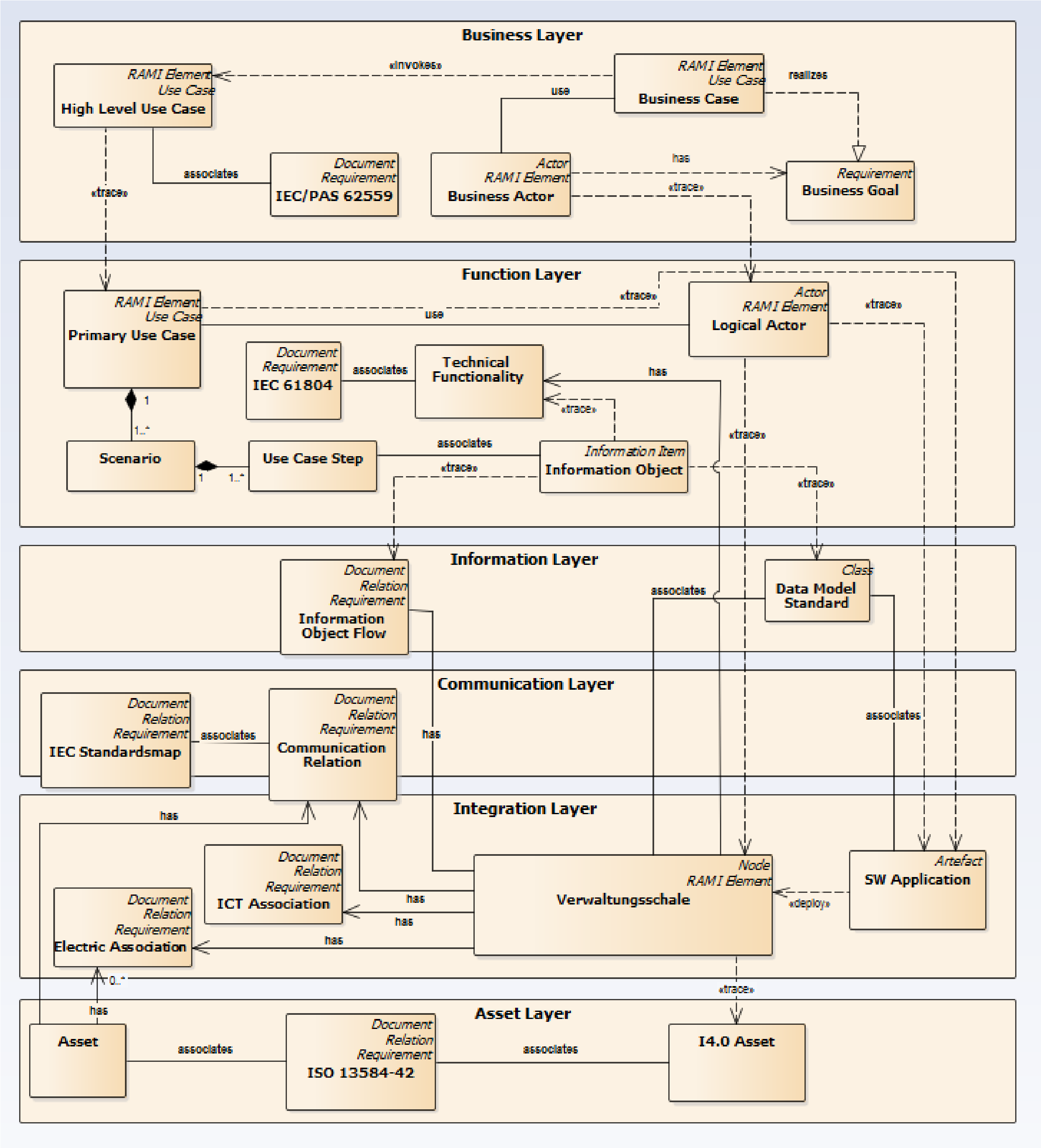
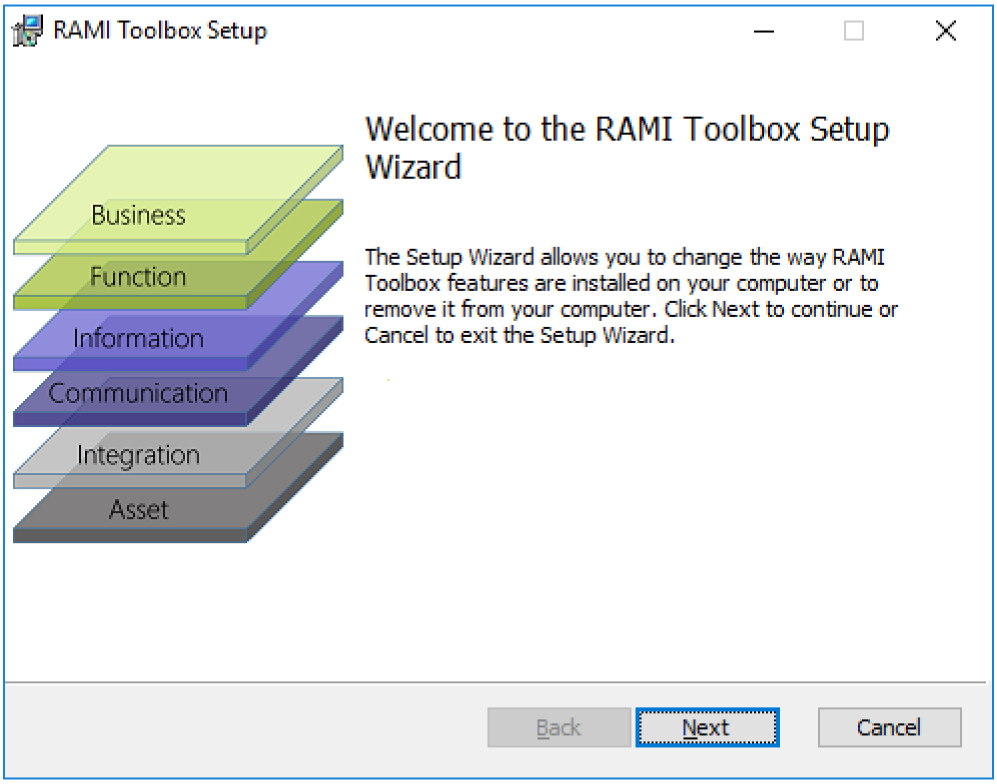


Figure 3 - The RAMI 4.0 Metamodel

4. 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.



RAMI Toolbox Setup

Welcome to the RAMI Toolbox Setup Wizard

The Setup Wizard allows you to change the way RAMI Toolbox features are installed on your computer or to remove it from your computer. Click Next to continue or Cancel to exit the Setup Wizard.

Figure 4: RAMI 4.0 Installation Wizard

5. Proposed Development Process

The proposed development process is inspired from the Use Case Mapping Process (UCMP). It basically consists of three phases, where the first two phases reflect the main tasks of the UCMP, but in an adopted schedule.

First, during the System Analysis Phase, the systems functionality is to be described. This is done by executing a Use Case Analysis, developing the RAMI 4.0 Business Layer and the RAMI 4.0 Function Layer. After this, the Integration Layer is being developed in the Design Phase. From there, the other Layers can be developed like the Asset Layer or during the System Architecture Phase, the development of the Information and the Communication Layer is done. The Design & Development Phase, which addresses the realization of individual systems, is not RAMI 4.0 specific and hence can be done by means of classic systems engineering methods. It is conceptualized as iterative phase to highlight the idea of suggested agile development.

The following sections describe the steps to be performed in more detail.

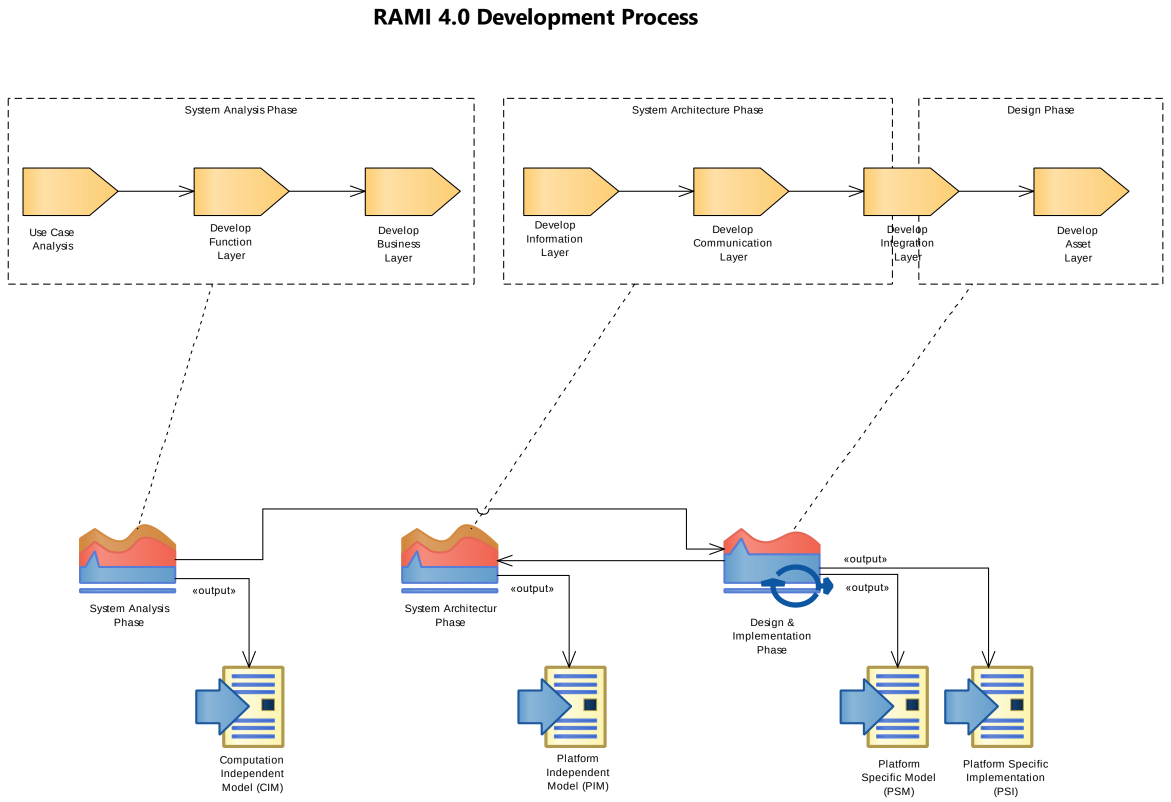

RAMI 4.0 Development Process
System Analysis Phase
System Architecture Phase
Design Phase
Use Case Analysis
Develop Function Layer
Develop Business Layer
Develop Information Layer
Develop Communication Layer
Develop Integration Layer
Develop Asset Layer
System Analysis Phase
System Architectur Phase
Design & Implementation Phase
«output»
«output»
«output»
«output»
Computation Independent Model (CIM)
Platform Independent Model (PIM)
Platform Specific Model (PSM)
Platform Specific Implementation (PSI)

Figure 5 – RAMI 4.0 Development Process

Element	Notes
System Analysis Phase	During the System Analysis Phase the specification for the system to be built has to be developed. Therefor it is necessary to identify the project-roles, the stakeholder and their individual requirements and the interrelation of the project with its environment and its project context.
Computation Independent Model (CIM)	The Term "Computation Independent Model" is taken from Model Driven Architecture (MDA). It focuses on the desired behaviour and function of a system without mentioning how the system is implemented (Separation of concerns).
System Architecture Phase	The System Architecture Phase is used to map the prior defined functionalities to a basic architecture. This basic architecture consists of the major function blocks and the concerning interfaces of the whole system.
Platform Independent Model (PIM)	The "Platform Independent Model" aims to identify major function blocks without relating them to a concrete implementation
Design & Implementation Phase	The Design & Implementation Phase is designed as iterative process. The underlying idea is to support agile development aspects. During this phase, a prior developed architecture is designed and implemented.
Platform Specific Model (PSM)	The "Platform Specific Model" describes the design for the specific components. The design is done with respect to the selected platform.
Platform Specific Implementation (PSI)	The "Platform Specific Implementation" is the last Layer of the MDA deliverables. As the name mentions, it depicts the implementation itself.

6. System Analysis Phase

6.1. Use Case Analysis

The execution of the Use Case Analysis is one of the key tasks that has to be performed. It focuses on the identification of involved actors and goals and on the decomposition of the Product-Owner inputs into single Use Cases. As the identified model elements state the basis for all further modeling, this is a very crucial task and should obtain the necessary attention.

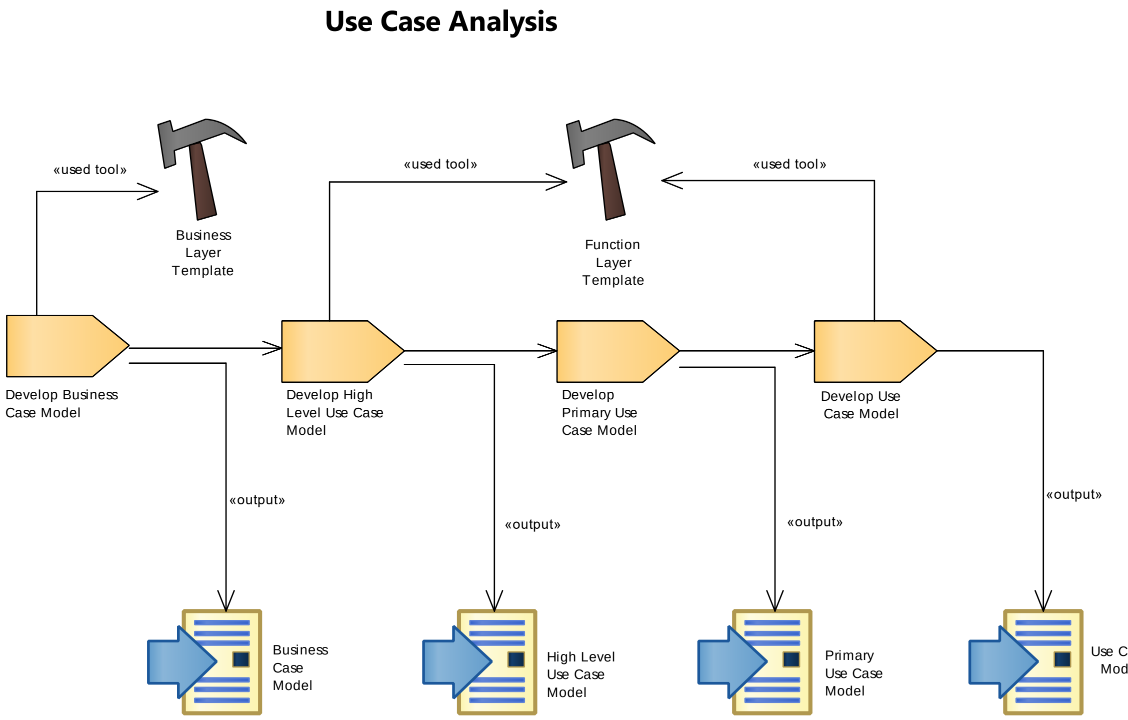


Figure 6 - Use Case Analysis

The four tasks of the Use Case Analysis are described in the following subsections in detail.

6.1.1. Develop Business Case Model

Focus of this task is the identification of involved *Business Actors (BA),* their assigned *Business Goals (BG)* and individual *Business Cases (BC)* that are performed by the BA in order to reach the individual BG. In addition, *High Level Use Cases (HLUC)* are identified and modeled that are included by the BC.
After creating a new Enterprise Architect project with an empty model you can make use of the *Model Wizard* to get a template for the development of the Business Case Model.

1. *Start the Model Wizard (Menu → Project → New Model) and select "RAMI" as Technology and "RAMI Business Layer" as model template. Of course you can also create the single packages and diagrams manually, but for the beginning the template is useful to get an idea about the concept.*

This step creates a new package named "RAMI Business Layer" within your model. The content of this package is an example that illustrates, how a Business Case Model looks like.

Developing your own Business Case Model comprises the following steps:

2. *Model the identified **Business Actors** by making use of the Model Element "Business Actor". This element can be found in the View "Toolbox" and placed by drag and drop onto the diagram.*

3. *Model the individual **Business Goals** for every Business Actor by using the element "Business Goal". Bring them into relation with the concerning Business Actor by making use of the "dependency" relation.*

4. *Model the **Business Cases** that are performed by the Business Actors in order reach the individual Business Goals. The relation between Business Actor and Business Case is of type "use"; the relation between Business Case and Business Goal is of type "realize". Typically numerous Business Cases exist. You can make use of all UML Use Case relations to model the dependencies between them. It is also a good practice to make diagrams composit in order to reduce complexity.*

5. *Identify and model specific **High Level Use Cases** by making use of the element "High Level Use Case". Bring them in relation with individual Business Cases by making use of the "invoke" relation. Say, a Business Case invokes High Level Use Cases.*

Please note that every Business Case from the template is linked to an individual RAMI Business Layer Diagram. These diagrams will be modeled after finishing the RAMI Function Layer, as there will be defined which domains and zones are affected by each related High Level Use Case.

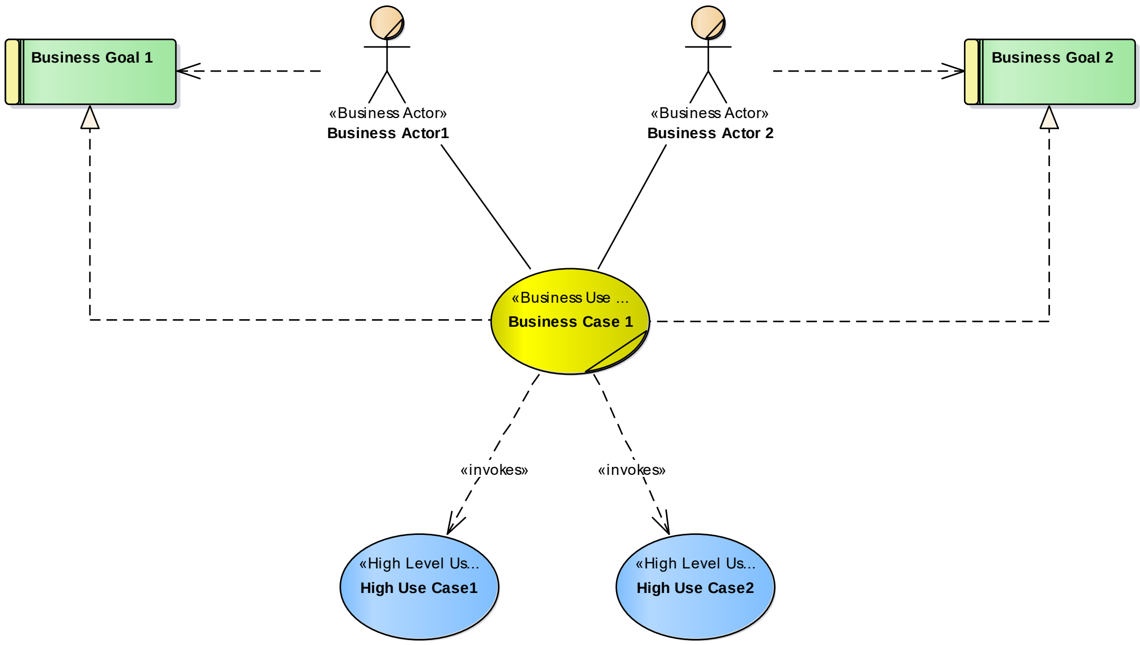


Figure 7 - Business Case Model (Example)

6.1.2. Develop High Level Use Case Model

After the identification of the involved High Level Use Cases (HLUC) in the prior executed task they need to be described in more detail. The description of the HLUCs consists of two basic steps. First, model the relations between the individual HLUC's by making use of the standard Use Case relationships from the UML. Next, decompose every single HLUC in more granular Primary Use Cases (PUC) as well as Use Cases (UC) and describe their cooperation.

These modeling tasks are performed at the level of the RAMI Function Layer. For the ease of use, the RAMI 4.0 Toolbox provides some adequate templates which can be used for modeling as described below:

1. *Start the Model Wizard (Menu → Project → New Model) and select "RAMI" as Technology and "RAMI Function Layer" as model template. Of course you can also create the single packages and diagrams manually, but for the beginning the template is useful to get an idea about the concept.*

This step creates a new package named "RAMI Function Layer" within your model. The content of this package is an example that illustrates, how a model of HLUCs and PUCs/UCs could look like. Besides the model itself the package shows the best practice for structuring your model: Inside the package is a RAMI Function Layer diagram located that is used to depict the relations between the single HLUCs. For every HLUC an individual Sub-Package with the name of the corresponding HLUC and the stereotype "*HLUC"* is integrated.
Each of these Sub-Packages holds two diagrams of type RAMI Function Layer. The first diagram is used to decompose the HLUC into more granular PUCs/UCs and to describe their relations. The second diagram is used to locate the involved PUCs/UCs within the RAMI plane, consisting of domains and zones. Again, for every PUC/UC an individual Sub-Package with the name of the PUC/UC and the stereotype *"PUC*/UC*"* exists.
To create your own High Level Use Case Model follow the steps below:

2. *Place all HLUC's you identified during the prior task on the Top-Level Diagram in the RAMI Function Layer Package and model the relations between them.*

3. *To keep your model clean and to enhance the readability of the later on generated reports, create a package structure as described above and move the HLUCs from the RAMI Business Layer package to the corresponding Sub-Package in the RAMI Function Layer Package.*

4. *For every HLUC Sub-Package create a diagram of type "RAMI Function Layer"*

5. *Decompose the HLUC into more granular PUCs/UCs and describe their cooperation. It is good practice to make use of Activity Diagrams to do so. To create compact information you can also place the Activity Diagram itself on this diagram. Depict functional interrelations between single PUCs/UCs by making use of the "Functional Interrelation" relation.*

6. *Hint: You can automatically generate an Activity Diagram from the HLUC. To do so, you have to invoke the single PUCs/UCs within the "Scenario" description of the HLUC. Please learn more about "Scenarios" and "Generate Diagrams" in the Enterprise Architect Help.*

6.1.3. Develop Primary Use Case Model and Use Case Model

The aim of this task is a more detailed description for every single Primary Use Case (PUC) and Use Case (UC) developed in the prior task. Basically it is valid to make use of any possibility that helps understanding the functionality of the PUC/UC. In praxis it is useful to describe all PUCs/UCs in a similar way. A proven method is to describe the PUCs/UCs by using Activity and Sequence Diagrams. Enterprise Architect comprises a very powerful possibility for an automated generation of these diagrams. Creating the descriptions that way is a very efficient approach for modeling the PUCs/UCs. To make use of this possibility follow the steps below:

1. *Create a RAMI Function Layer Diagram within every PUC/UC Sub-Package and place the corresponding PUC/UC in it.*

2. *Describe the functionality of the PUC/UC in any text editor (or use the available Product-Owner description). Write in a way, that every sentence represents one step or Activity of the PUC/UC and write every sentence in a new line. Try to create sentences that hold the name of every involved actor and every exchanged information. For example instead of "Actor A sends Voltage Measurement to Actor B.*

Actor B responds with the new operation point" write "Actor A sends Voltage Measurement to Actor B. Actor B transmits Operation Point to Actor A.". Now, copy the written text to the clipboard.

3. Open the properties dialog of the corresponding PUC/UC and select "Scenarios" from the "Rules" menu on the left. Here, select the "Structured Specification" tab and right click into the Action description field. Now you can select to "create structure from clipboard text". Doing this, creates for every sentence written before a single step in the Use Case description. After this, you can close the property.

4. Create for every Actor involved the corresponding Model Element (RAMI Actor) and link it to the PUC/UC. Take care to name the actors the same way as they are called in the textual description. Of course, if the Actor already exists in your model, do not create it again but place and link the existing one. If you open the Scenario Description within the PU/UC C properties dialog again, you will see that the name of the connected actor is now underlined.

5. Having all involved Actors linked to the PUC/UC, you can generate an Activity and Sequence Diagram automatically. In the menu bar of the "Structured Specification" tab you can find the Icon for "Generate Diagrams". Here you can select which type of diagrams you want to generate.
   Please Note: The described functionality is very powerful and enables you to describe Use Cases in a very efficient way. You can manipulate the activity diagram and update the structured description with the changes. Of course it is also possible to create alternate and exception paths and much more. Please refer to the very valuable Enterprise Architect Help to learn more about these capabilities.

6. In the generated Activity Diagram you can analyze the PUC/UC and create model elements for every Information Item transmitted. Use the element "Information Object" for this task. As this diagram is of type "Activity Diagrams" you won't find this element in the toolbox. You have to select the RAMI Function Layer toolbox by clicking on "More tools…" at the top of the toolbox and selecting "RAMI -Toolbox / RAMI Function Layer". Depict the relation between a single Activity (Use Case Step) and the Information Item by means of the "dependency" relation.

7. In the generated Sequence Diagram you can attach the created Information Items to the corresponding sequence. Right click on the corresponding connection and select "Advanced/Information Flows Realized". In the appearing dialog click on "click to create new information flow…". A package browser opens where you can select the referred Information Item. Again, please refer to the Enterprise Architect Help to learn more.

8. You can place the generated Activity and Sequence Diagrams directly in the PUC/UC diagram simply by using "drag and drop". This helps, as you have all relevant description – the PUC/UC and it's involved actors, the algorithmic aspect (activity diagram) and the communication aspect (sequence diagram) – in one diagram that will be used during reporting.

6.2. Develop Business Layer

The focus of the RAMI Business Layer is to show the affected domains and zones for every Business Case and hence has to be modeled for each of them. It is suggested to use the involved High Level Use Cases as model elements. The steps below describe how to model the RAMI Business Layer for a single Business Case.

1. *Use the RAMI Business Layer template to create a new RAMI Business Layer diagram for every Business Case (New Diagram/ RAMI Toolbox/ RAMI Business Layer)*

2. *To make the RAMI Plane (domains and zones) visible, open “Diagram” in the pull down menu and select the “Swimlanes and Matrix” dialog. Within the register card “Matrix” you can select a Model Profile. Select the “RAMI Business Layer” Profile and activate the Check-Box “activate” at the top.*

3. *For every invoked HLUC open the RAMI Function Layer diagram and analyze which domains and zones are affected.*

4. *Go to the RAMI Business Layer diagram and place the corresponding HLUC in it. Switch the appearance of the HLUC to rectangular (right click / Advanced / Use Rectangular Notation)*

5. *Locate and resize the HLUC according to the affected domains and zones. Having done this for all HLUCs, you see which domains and zones are affected by the Business Case in total.*

6. *Place the Business Case in the diagram (at the bottom of the Z-Order), switch to rectangular notation and resize and locate it according to the totally affected domains and zones.*

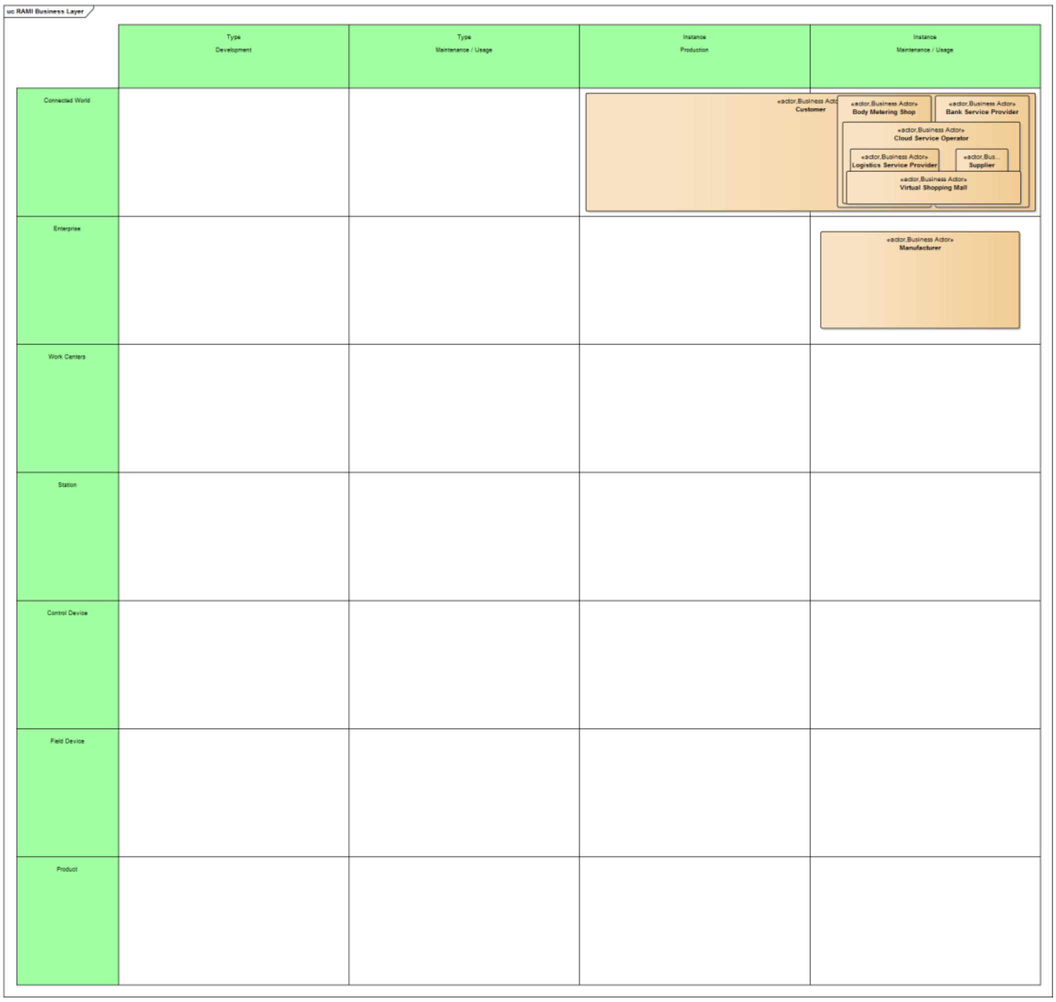


Figure 8 - RAMI Business Layer (Example)

6.3. Develop Function Layer

The RAMI Function Layer describes the location of all involved Primary Use Cases (PUC), Use Cases (UC) and Actors for a specific High Level Use Case (HLUC). Hence, it has to be developed for every HLUC separately. The steps below describe how to create the RAMI Function Layer for a single HLUC.

1. *Use the RAMI Function Layer Template to create a new RAMI Function Layer Diagram inside the “HLUC” package.*

2. *To make the RAMI Plane (domains and zones) visible, open “Diagram” in the pull down menu and select the “Swimlanes and Matrix” dialog. Within the register card “Matrix” you can select a Model Profile. Select the “RAMI Function Layer” Profile and activate the Check-Box “activate” at the top.*

3. *Place all involved PUCs/UCs and Actors in the diagram and arrange them in respect to the corresponding domains and zones.*

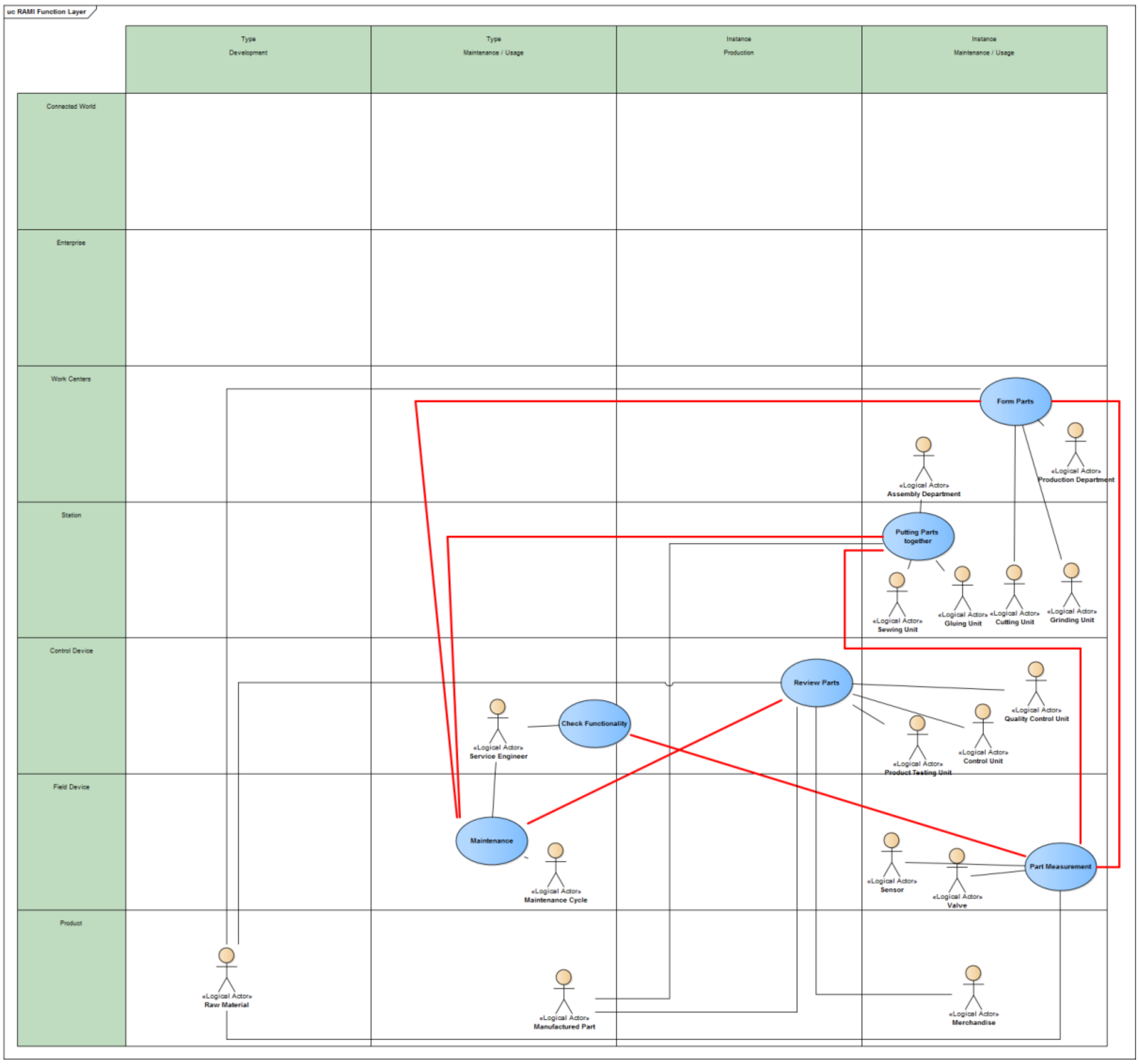


Figure 9 - RAMI Function Layer (Example)

7. Design Phase

7.1. Develop Integration Layer

By developing the RAMI Integration Layer, in a first step a mapping from logical actors to physical components has to be done. Next, the identified physical actors are utilized to create the RAMI component layer. Both steps can be executed by usage of the supplied Integration Layer Template.

7.1.1. Map Actors to Administration Shells

To perform the mapping of the logical Actors (as created in the Function Layer) to administration shells follow the steps below:

1. *Use the RAMI Integration Layer template to create a new RAMI Integration Layer diagram*
2. *Place all involved logical Actors to the diagram*
3. *Create a virtual representation (administration shell) for every Actor. This virtual representation can either be a specific component or a Software Application. You can use the appropriate model elements from the toolbox for this step. Use the "trace" relation to depict the model transformation step. You can also create new Components by using the general "Component" element. It is also possible to use your own images for these Components. Please use the Enterprise Architect Help to learn more about this possibility.*
4. *If a Software Application is introduced, you need to create a "Computer" Component as well. Use the "hosts" relation to show, which computer hosts which Software Application.*

7.1.2. Develop Integration Layer

The development of the RAMI Component Layer can be done by following the steps below:

1. *Use the RAMI Integration Layer template to create a new RAMI Integration Layer*
2. *To make the RAMI Plane (domains and zones) visible, open "Diagram" in the pull down menu and select the "Swimlanes and Matrix" dialog. Within the register card "Matrix" you can select a Model Profile. Select the "RAMI Integration Layer" Profile and activate the Check-Box "activate" at the top.*
3. *Place all prior created physical components within the plane in the corresponding domain and zone.*
4. *Create additional necessary components, like ICT-Networks (represented as clouds), Electric Networks (e.g. "Medium Voltage Network") or electric components like Transformers. Following the modeling-concepts you can of course make single components like for example ICT-Networks "composite" and describe them in more detail. Please refer to the Enterprise Architect Help to learn more about this possibility.*
5. *Model the relations between the individual components. You can use the "Electric Association" relation to depict electric connections, especially along the electric conversion chain in the process zone, and "ICT Association" to model ICT connections.*

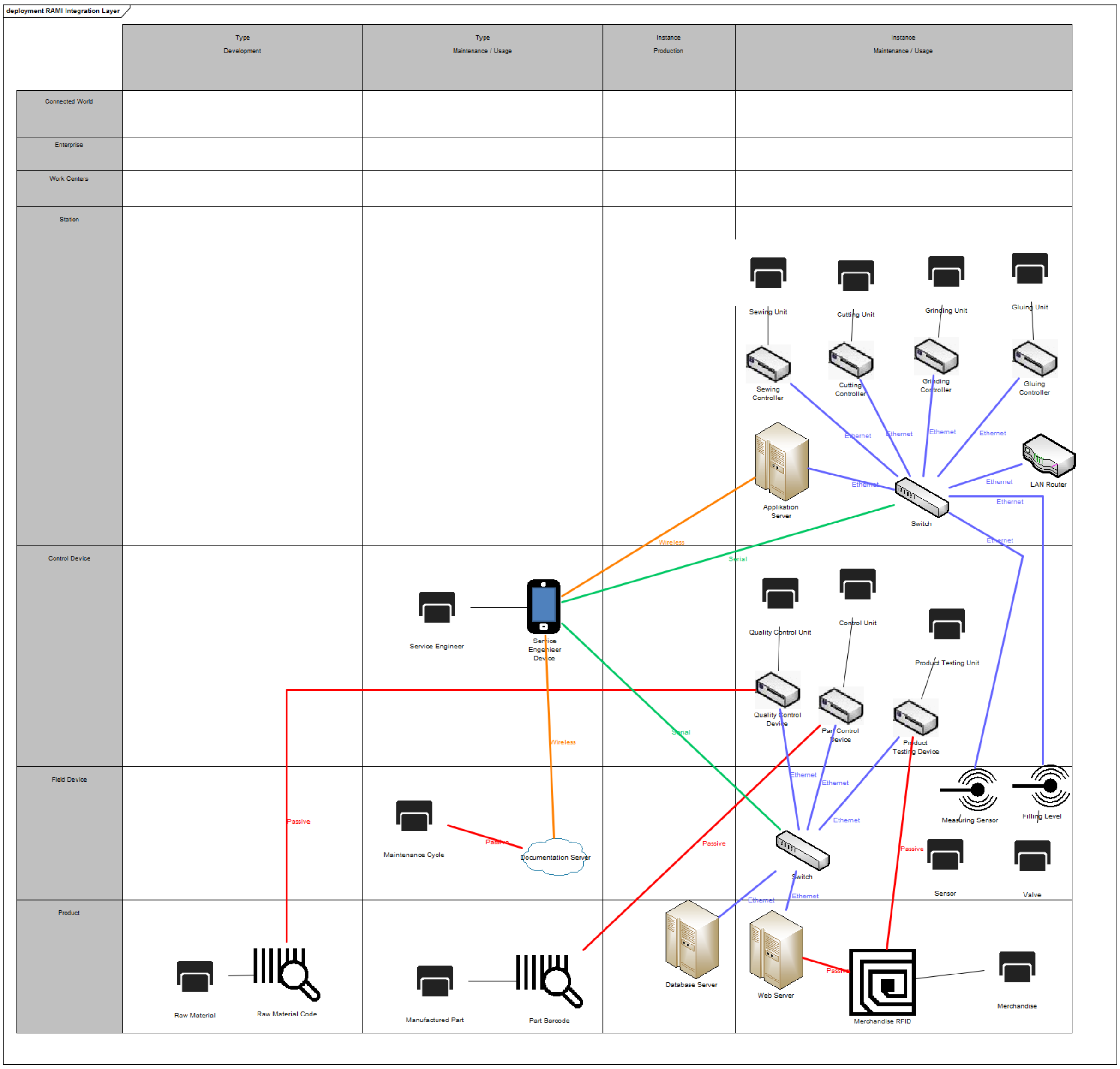


Figure 10 - RAMI Component Layer (Example)

7.2. Develop Asset Layer

The Asset Layer represents all physical representation of actors, administration shells, etc. In a first step a mapping from administration shells to physical components has to be done. Next, the identified physical actors are utilized to create the RAMI Asset layer. Both steps can be executed by usage of the supplied Asset Layer Template.

7.2.1. Map Administration Shells to Physical Components

To perform the mapping of the logical Actors (as created in the Function Layer) to physical components follow the steps below:

1. *Use the RAMI Asset Layer template to create a new RAMI Integration Layer diagram*

2. *Place all involved administration shells to the diagram*

3. *Create a physical representation (one or more assets) for every administration shell. This physical representation can either be a specific component or a Software Application. You can use the appropriate model elements from the toolbox for this step. Use the "trace" relation to depict the model transformation step. You can also create new Components by using the general "Component" element. It is also possible to use your own images for these Components. Please use the Enterprise Architect Help to learn more about this possibility.*

7.2.2. Develop Asset Layer

The development of the RAMI Component Layer can be done by following the steps below:

1. *Use the RAMI Asset Layer template to create a new RAMI Asset Layer*

2. *To make the RAMI Plane (domains and zones) visible, open "Diagram" in the pull down menu and select the "Swimlanes and Matrix" dialog. Within the register card "Matrix" you can select a Model Profile. Select the "RAMI Asset Layer" Profile and activate the Check-Box "activate" at the top.*

3. *Place all prior created physical components within the plane in the corresponding domain and zone.*

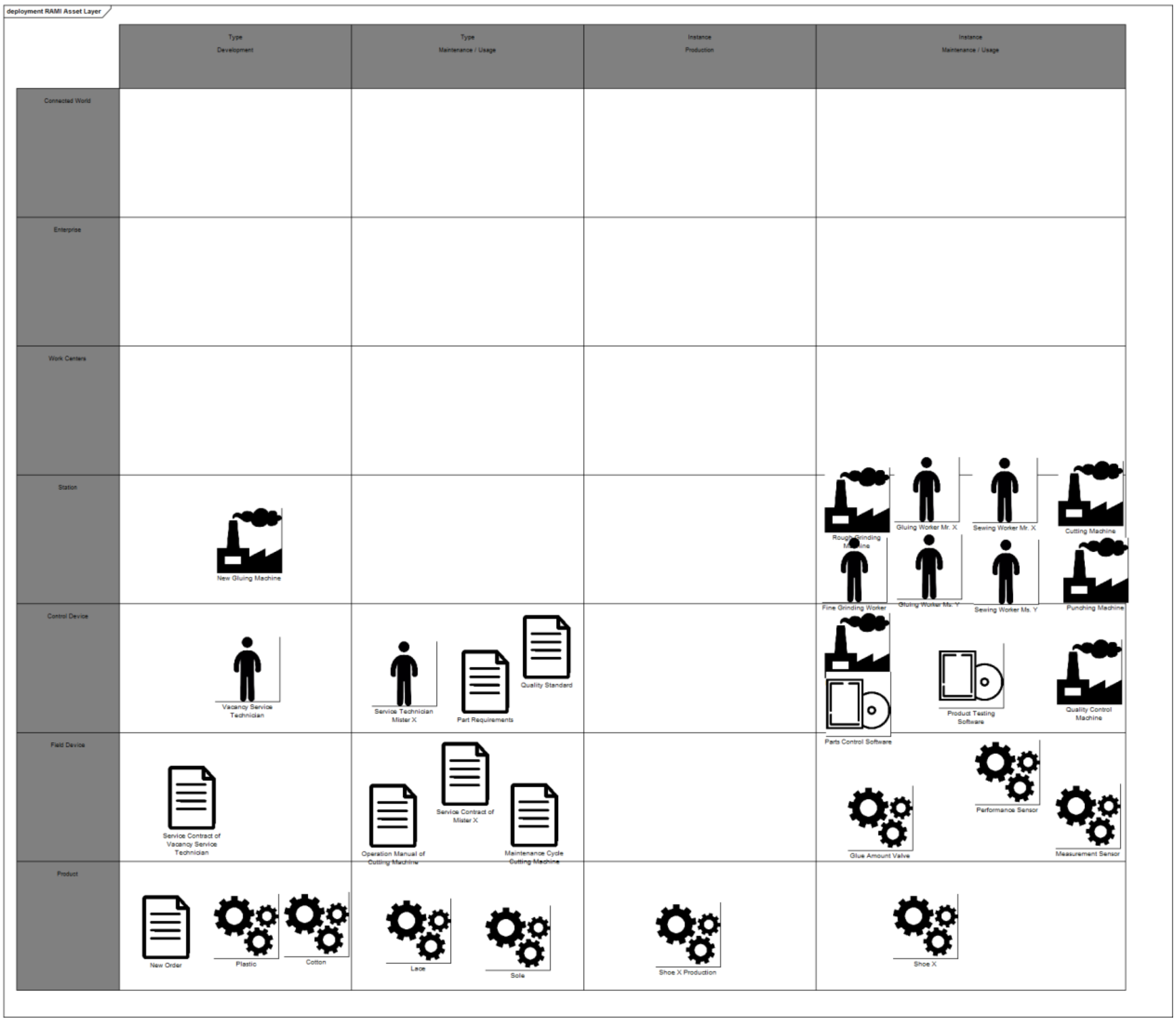


Figure 11: RAMI Asset Layer (Example)

8. System Architecture Phase

8.1. 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.
The development of the Information Layer consists of three consecutive tasks as described in the following subsections.
You can make use of the Model Wizard to create the involved diagrams automatically:

1. *Start the Model Wizard (Menu → Project → New Model) and select "RAMI" as Technology and "RAMI Information Layer" as model template. Of course you can also create the single packages and diagrams manually, but for the beginning the template is useful to get an idea about the concept.*

This step creates a new package named "RAMI Information Layer" within your model. The content of this package is an appropriate sub package for each of the three necessary steps.

8.1.1. Develop Business Context View

The Business Context View models the information object flows between individual components. It can be developed following the steps below:

1. *Open the “Business Context View” Diagram as created from the Model Wizard or create this diagram manually*

2. *Place all components in the diagram. Hint: If you copy the elements from the Component Layer, you will have arranged them in the same way as in the Component Layer*

3. *Hide Relations. After placing the components in the diagram, all relations (e.g. ICT Associations, Electric Associations) as introduced so far will be visible. As we are not interested in these relations within the Information Layer, you can make them invisible. Open the “Diagram” Drop-Down Menu, select “Advanced” → “Visible Relations” and deselect all relations. Do not accidently delete these relations instead of making them invisible. Deleting them will delete them not only from the diagram but from the model, so you won’t see them in the original diagrams anymore.*

4. *Create Information Object Flows. In the Primary Use Case Diagram we identified which Information Objects are to be communicated between logical actors. Use the “Information Object Flow” Relation to model these flows here as well and select the appropriate “Information Objects” in the Pop-up window. Note: This step is some “copy” task, in a future version of the RAMI Toolbox this will be automated.*

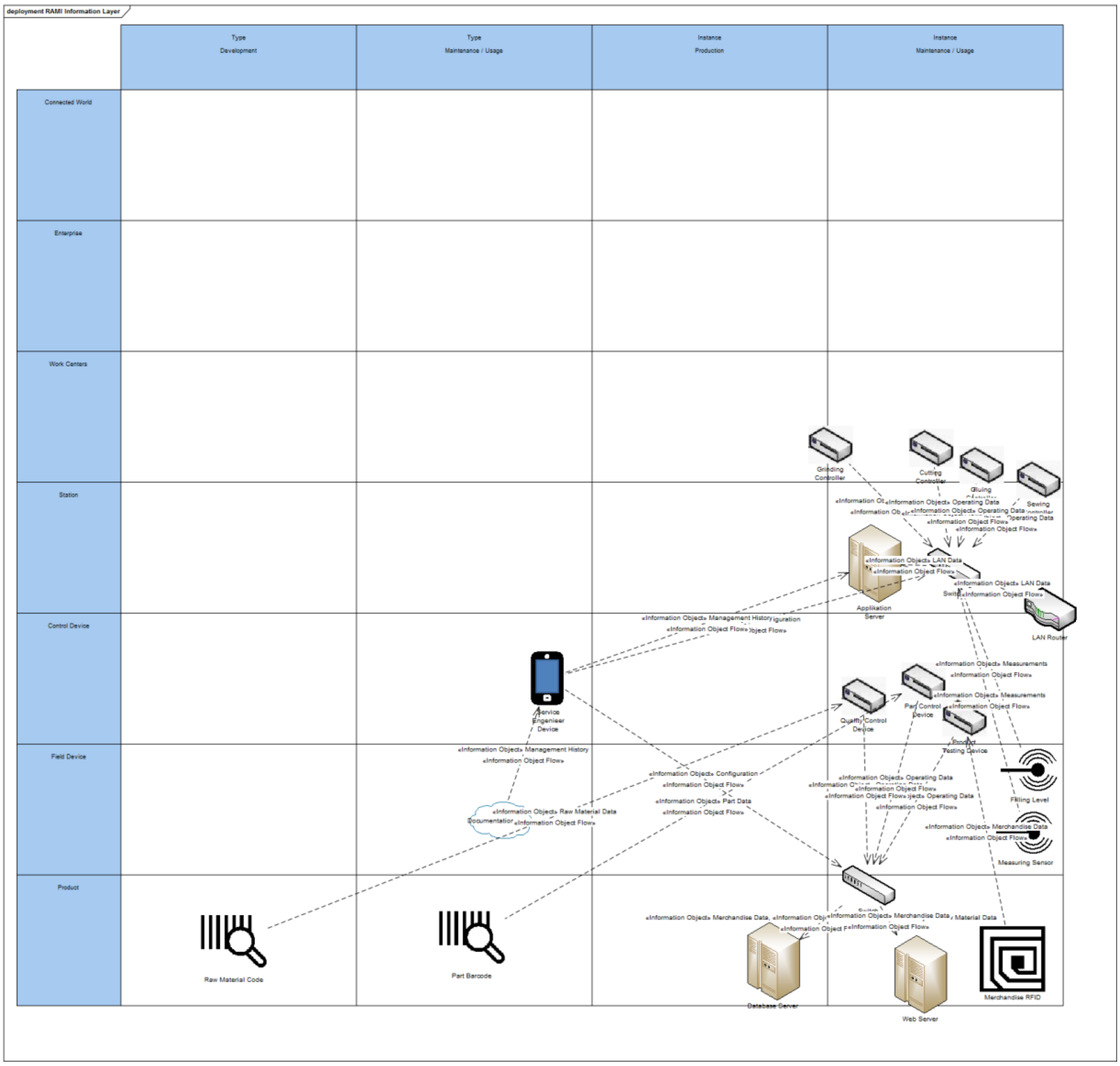


Figure 12 - Business Context View (Example)

8.1.2. Perform Standard and Information Object Mapping

1. *Open the “Standard and Information Object Mapping” Diagram as created from the Model Wizard or create this diagram manually*

2. *Place all components in the diagram. Hint: If you copy the elements from the Component Layer, you will have arranged them in the same way as in the Component Layer*

3. *Turn the visibility of all relations – except the Information Object Flows – off*

4. *Create appropriate “Data Model Standard” Elements and associate the related components with them by using the “Associates Data Model Standard” relations.*

5. *Place all used Information Objects in the diagram and state the relations between Data Model Standards and Information Objects by using the “Provides Information Object” relations.*

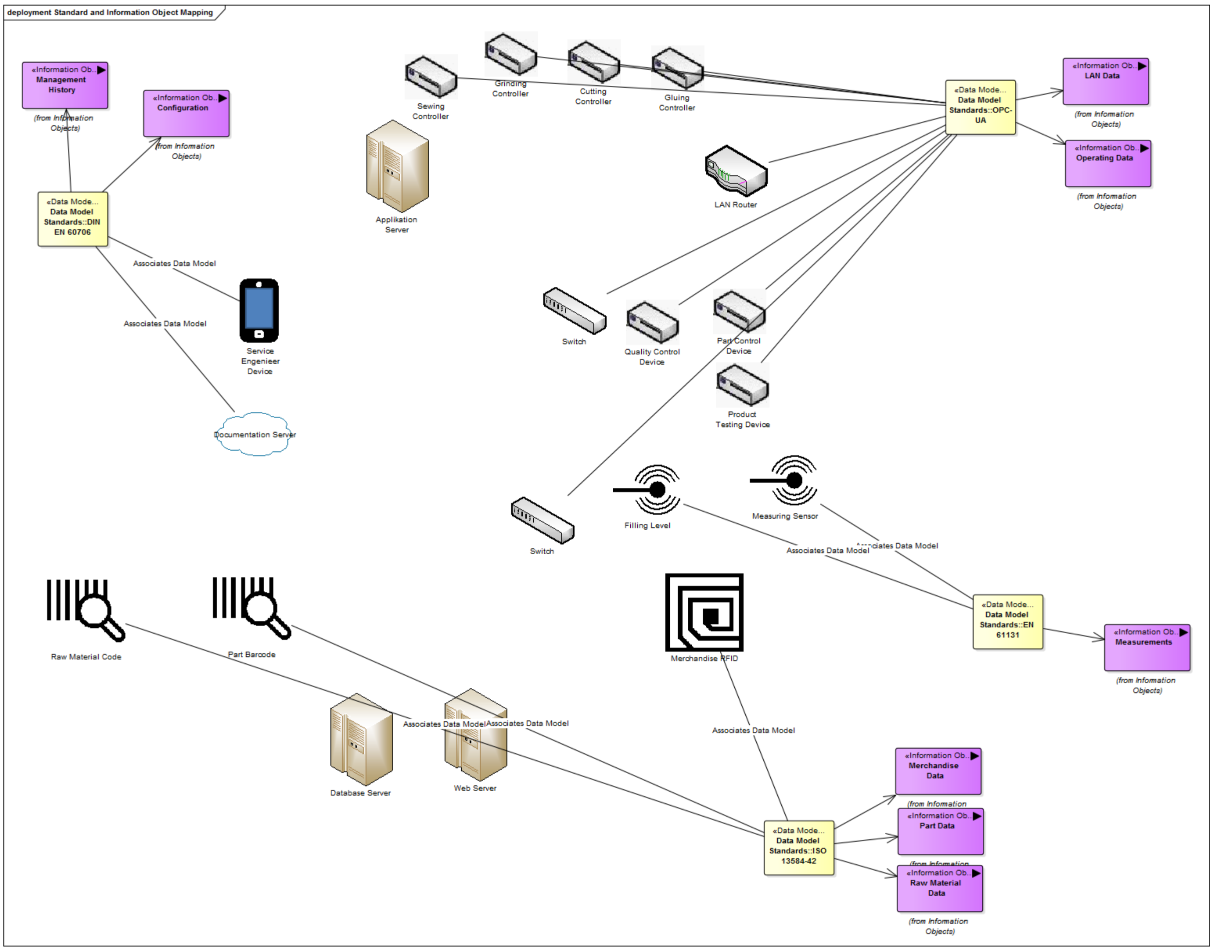

deployment Standard and Information Object Mapping
«Information Ob.. Management History
(from Information Objects)
«Information Ob.. Configuration
(from Information Objects)
«Data Mode... Data Model Standards::DIN EN 60706
Associates Data Model
Associates Data Model
Service Engenieer Device
Documentation Server
Sewing Controller
Grinding Controller
Cutting Controller
Gluing Controller
Applikation Server
«Data Mode... Data Model Standards::OPC-UA
«Information Ob.. LAN Data
(from Information Objects)
«Information Ob.. Operating Data
(from Information Objects)
LAN Router
Switch
Quality Control Device
Part Control Device
Product Testing Device
Switch
Filling Level
Measuring Sensor
Associates Data Model
Associates Data Model
«Data Mode... Data Model Standards::EN 61131
«Information Ob.. Measurements
(from Information Objects)
Raw Material Code
Part Barcode
Merchandise RFID
Database Server
Web Server
Associates Data Model
Associates Data Model
Associates Data Model
«Data Mode... Data Model Standards::ISO 13584-42
«Information Ob.. Merchandise Data
(from Information
«Information Ob.. Part Data
(from Information
«Information Ob.. Raw Material Data
(from Information Objects)

Figure 13 - Standard and Information Object Mapping (Example)

8.2. Develop Communication Layer

The focus of the Communication Layer is to depict the used technology and protocols for the communication between single components.
You can model this layer by following the steps below.

1. *Start the Model Wizard (Menu → Project → New Model) and select "RAMI" as Technology and "RAMI Communication Layer" as model template. Of course you can also create the single packages and diagrams manually, but for the beginning the template is useful to get an idea about the concept.*

2. *Place all components in the diagram. Hint: If you copy the elements from the Component Layer, you will have arranged them in the same way as in the Component Layer.*

3. *Hide Relations. After placing the components in the diagram, all relations (e.g. ICT Associations, Electric Associations) as introduced so far will be visible. As we are not interested in these relations within the Information Layer, you can make them invisible. Open the "Diagram" Drop-Down Menu, select "Advanced" → "Visible Relations" and deselect all relations. Do not accidently delete these relations instead of making them invisible. Deleting them will delete them not only from the diagram but from the model, so you won't see them in the original diagrams anymore.*

4. *Use the "Communication Path" Relations to model the communication paths between the components*

5. *For every communication path define the protocol and the technology as tagged value*

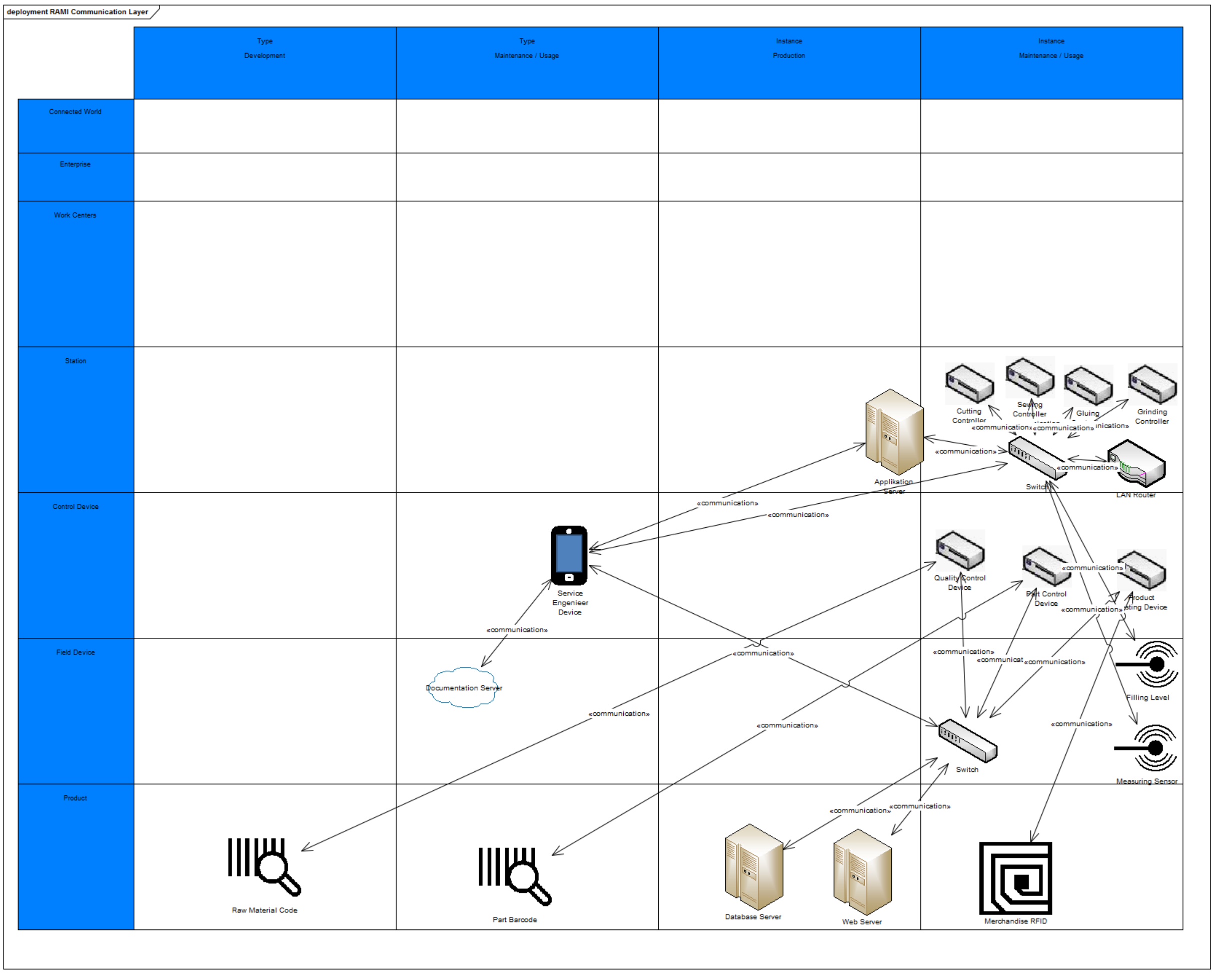


Figure 14 - RAMI Communication Layer (Example)

9. Design and Development of single Components

The development of all six RAMI layers delivers a big picture of a distributed Industrie 4.0-based system. For every component it is defined, which functionality it has to deliver and how it interacts with its surrounding components over all interoperability layers.
For the development of one, specific component out of this distributed system you could say that the systems' context and the systems' interfaces are defined. The system (component) itself is yet treated as black box. During the development of this system, the black box now is to be turned into a white box.
To start the development one component, it is a very valuable possibility, to show the functionality and all interfaces over all interoperability layers in one diagram. This can easily be achieved by following the steps below:

1. *Make the system (component) of interest composite and create a new diagram*
2. *Place the component in this diagram*
3. *Right click on the component and select "insert related elements". Now all related elements and the relations are placed into this diagram.*

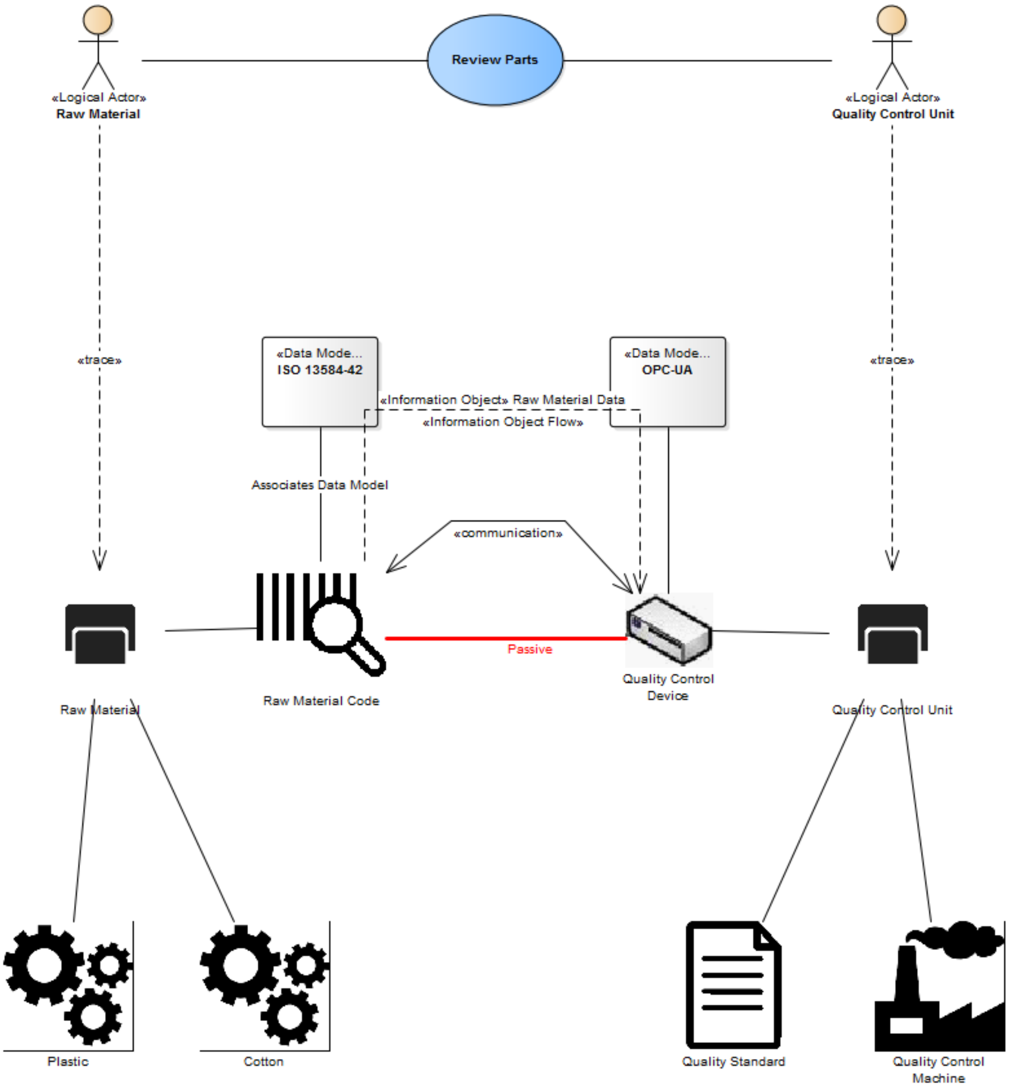


Figure 15 - Component Responsibility Model (Example)

10. Generating Reports

The generation of reports is a very powerful possibility delivered by Enterprise Architect. You can both, create HTML reports (Click through the whole model in an ordinary Internet Browser) or create documents.
The toolbox provides some useful templates which directly can be used for the generation of reports or which can be extended according to the individual needs. These templates are available in the “Resources” View under the section “Templates → Document Templates → RAMI-Toolbox”.

As the possibilities for automated generation of reports are vast and hence, introduce a certain complexity, the publication of a video tutorial according this task is scheduled in near future.

11. 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:

- Integration/Import of the Intelligrid Use Case Template
- Visualization
- Integration of Libraries
    - ENTSO-E Role Model
    - M/490 Use Case Management Repository
    - M/490 Actor List
    - ICT Component Library
    - Data Model Standards
    - Protocol/Technology Standards
    - …
- Logic Mapping of Elements to Domains/Zones
- Element Attribute Refinements