Customer Requirements Specification

(Lastenheft)

(TINF20C, SWE I Praxisprojekt 2021/2021)

Project: Modelling Wizard for Device Descriptions

Customer: Rentschler & Holder

Rotebühlplatz 41

70178 Stuttgart

Supplier: Team 1 (Linus Eickhoff, Lukas Ernst, Florian Kaiser, Florian Kellermann, Malte Horst)

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Date | Author | Comment |
| 0.1 | 30.10.2021 | Malte Horst | created |
|  | 31.10.2021 | Linus Eickhoff | Goal description added |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Allgemeine Hinweise:

Alles, was in dieser blauen Schriftart gesetzt ist, dient nur zur Erläuterung und sollte im fertigen Lastenheft nicht mehr auftauchen!

Der Umfang dieses Dokuments darf sechs Seiten nicht überschreiten.

Ein Lastenheft enthält eine grobe Beschreibung aller fachlichen Anforderungen, die das zu entwickelnde Produkt erfüllen muss. Die Inhalte des Lastenheftes (CRS) dienen als Grundlage für das Pflichtenheft und können -wenn sinnvoll- im Pflichtenheft (SRS) wieder verwendet werden.

#### ****Offene Punkte****

In diesem Abschnitt sollen alle Probleme und offenen Fragen gesammelt werden. Bei einem fertigen Lastenheft sollte er leer sein, aber bei Zwischenversionen kommt diesem Abschnitt besondere Bedeutung zu!

CONTENTS

*1.* Goal 4

2. Product Environment 5

*3.* Product Usage 7

*3.1.* Business Processes 7

3.1.1. <BP.001>: Device creation 7

3.1.2. <BP.002>: Edit Device 7

3.1.3. <BP.003>: Export Device 8

3.2. Use Cases 9

3.2.1. <UC.001> Create new device 9

3.2.2. <UC.002> Create interface or load interface from Library 10

3.2.3. <UC.003> View device data and device interface data 10

3.2.4. <UC.004> Add Attachments for the Device 10

3.2.5. <UC.005> Format output as CAEX version 2.15/3.0 11

*3.3.* Features 12

3.3.1. /LF10/ ….. 12

3.3.2. /LF10/ Import 12

3.3.3. /LF20/ File validation 12

3.3.4. /LF30/ Error handling 12

3.3.5. /LF40/ GUI 12

3.3.6. /LF50/ Display device in a readable way 12

3.3.7. /LF60/ Edit device 12

3.3.8. /LF70/ Create device 13

3.3.9. /LF80/ Export device 13

*4.* Product Data 14

4.1.1. /LD10/ AML-DD 14

4.1.2. /LD20/ Save and load files 14

*5.* Other Product Characteristics 15

5.1.1. /NF10/ Installation Wizard 15

5.1.2. /NF20/ GUI 15

5.2. System Environment 15

6. References 16

# Goal

The goal is to overhaul an existing plugin for the AutomationML editor by developing a Windows standalone application, that does not depend on the editor, but still uses the AML Engine. Furthermore, the Windows stand-alone application should be designed to be more user-friendly and easier to use, by enhancing the GUI and reducing redundant or irrelevant content, that could distract the user.

Dieser Abschnitt hat die Aufgabe als Einleitung zu dienen. Beschrieben wird die Hauptaufgabe des Systems. Meist kann man von der Aufgabenstellung bzw. Auftragsanfrage abschreiben. Wichtig ist es, den Grund für die Systementwicklung (Probleme oder Geschäftsideen) und damit ihre Ziele herauszuarbeiten.

Benennen Sie auch die Zielgruppe, die später mit dem System arbeiten soll.

# Product Environment

AutomationML (AML) is short for Automation MarkUp Language and is used to describe parts of automation plants as objects. These objects can consist of multiple other objects and be part of a larger

assembly of objects. That way AML can be used to describe a single screw or an entire robot with the

necessary level of detail.

AML makes use of various standards to describe plant components:

1. CAEX (Computer Aided Engineering Exchange) to describe attributes of objects and their relations in a hierarchical structure. This is called a system topology. In this respect, CAEX forms

the overarching integration framework of AutomationML. [1]

2. COLLADA to describe the geometry and 3D models of a objects

3. COLLADA also integrates motion planning. It describes the connections and relations of moveable objects, which is called Kinematics.

4. PLCopen XML describes the logic. Internal behaviour and states if objects, action-sequences

and I/O connections are implemented via this format.

An IODD (IO Device Description) file describes the sensors and actuator of a plant or component. It also contains information on identity, parameters, process data, communication and more. It is written in XML-format, same as AML, which ensures a conversion.

Dieser Abschnitt hat die Aufgabe den Einsatzbereich des zu entwickelnden Systems klarzustellen. Dazu gehören Erläuterungen der notwendigen Fachbegriffe und deren Zusammenhänge ebenso wie die Darstellung der systemrelevanten Abläufe im Einsatzbereich.

Unter dem Produkteinsatz versteht man sowohl den direkten Problembereich, in dem das zu entwickelnde System eingesetzt werden soll, als auch die umgebenden Geschäftsprozesse.

Hier also den Problembereich des Projektes benennen und erläutern, ob es zu unterstützende Abläufe im Einsatzbereich (Geschäftsprozesse) gibt und wo sie zu finden sind.

Dieser Abschnitt muss so geschrieben sein, dass er, den Laien mit der Terminologie und den Zusammenhängen im Problembereich vertraut macht. Daher muss die Beschreibung möglichst allgemein sein. Außerdem sollte der Text gut strukturiert sein. Auch der Einsatz von erläuternden Graphiken ist manchmal sinnvoll.

Exhibitb4-2

Figure x: Product Environment

Wichtig ist es auch noch, gemachte Annahmen sauber von den oben beschriebenen Fakten getrennt aufzulisten. Dies erleichtert eine spätere Fehlersuche, wenn das System nicht die Erwartungen erfüllt.

# Product Usage

The following business processes, use cases and features shall be supported by the system.

Dieser Abschnitt hat die Aufgabe, die Anwendung des zu entwickelnden Systems sowohl überblicksartig als auch detaillierter aus Benutzersicht zu beschreiben. Abschliessend sollen die vom Produkt erwarteten Features beschrieben werden.

## Business Processes

Falls notwendig, sind hier die identifizierten Geschäftsprozesse näher zu beschreiben. Jeder von ihnen erhält einen eigenen Unterabschnitt gemäß dem Template. In diesem Abschnitt wird der Ablauf der Geschäftsprozesse des vorigen Abschnittes genauer beschrieben. Diese Abläufe sind es, die das zu entwickelnde System ausschnittsweise unterstützen soll.

### <BP.001>: Device creation

|  |  |
| --- | --- |
| Triggering Event: | User wants to create a new device |
| Result: | The system will create an empty new device file the user can then edit to his liking. This file can be saved and reedited later on. |
| Involved Roles: | User and Device Modelling Application |



Durch eigenes Aktivitätsdiagramm ersetzen.

Figure 2.2: <BP.001> Activity Diagram

### <BP.002>: Edit Device

|  |  |
| --- | --- |
| Triggering Event: | The user wants to edit a device file |
| Result: | The file will be opened in the application and the user can edit it. Changes made can be saved in the file. |
| Involved Roles: | User and Device Modelling Application |



Durch eigenes Aktivitätsdiagramm ersetzen.

Figure 2.3: <BP.002> Activity Diagram

### <BP.003>: Export Device

|  |  |
| --- | --- |
| Triggering Event: | The user wants to export a created/edited device |
| Result: | The device in the application will be saved to a file |
| Involved Roles: | User and Device Modelling Application |



Durch eigenes Aktivitätsdiagramm ersetzen.

Figure 2.3: <BP.002> Activity Diagram

## Use Cases

Aufgabe dieses Abschnittes ist es, einen Überblick über die Produktfunktionen zu geben. Dazu wird ein Use Case Diagramm eingesetzt, das eine abstrakte Sicht auf die Produktfunktionen und die externen Beteiligten an diesen Funktionen gibt.



Durch eigenes Use Case Diagramm ersetzen.

Figure x: Use Case Overview Diagram

### <UC.001> Create new device

Dieser Abschnitt muss für jeden Use Case wiederholt werden. Hier soll der Use Case mit einer ID versehen und kurz textuell sowie ggf. mit einem Aktivitätendiagramm erläutert werden.

|  |  |
| --- | --- |
| **Related Business Process:** | <BP.001>: Create new device |
| **Use Cases Objective:** | User wants to create a device by inserting the data manually into the user interface of the application |
| **System Boundary:** | The application itself |
| **Precondition:** | The user needs to have the minimal required data for the device on hand.  The program needs to be installed on the user system and opened. |
| **Postcondition on success:** | The entered data is displayed completely and correctly. |
| **Beteiligte Nutzer:** | Every end-user of the application |
| **Triggering Event:** | When the user opens the application and uses the ‘New device’ function to create a new device |

### <UC.002> Create interface or load interface from Library

|  |  |
| --- | --- |
| **Related Business Process:** | <BP.002>: Edit device |
| **Use Cases Objective:** | Creating a device interface by inserting the data manually into the user interface. Or to add an interface from one of the existing libraries. |
| **System Boundary:** | The application itself |
| **Precondition:** | The user needs to have the minimal required data for the device or interface to be added. |
| **Postcondition on success:** | The user has submitted the specific data completely and correctly |
| **Beteiligte Nutzer:** | Every end-user of the application. |
| **Triggering Event:** | When the user has the need to add/create a device interface. |

### <UC.003> View device data and device interface data

|  |  |
| --- | --- |
| **Related Business Process:** | <BP.002>: Edit device |
| **Use Cases Objective:** | After at least one device was successfully added, the device data should be visible and editable on the user interface. |
| **System Boundary:** | The application itself |
| **Precondition:** | The user added/loaded a device. |
| **Postcondition on success:** | The user added at least one device successfully. |
| **Beteiligte Nutzer:** | Every end-user of the application. |
| **Triggering Event:** | When the user has the need to view device data and device interface data. |

### <UC.004> Add Attachments for the Device

|  |  |
| --- | --- |
| **Related Business Process:** | <BP.002>: Edit device |
| **Use Cases Objective:** | It is possible to add an Attachment to the object, such as a manufacturer icon. |
| **System Boundary:** | The application itself |
| **Precondition:** | The user has added/loaded a device. |
| **Postcondition on success:** | The user added/loaded at least one device successfully. |
| **Beteiligte Nutzer:** | Every end-user of the application. |
| **Triggering Event:** | When the user has the need to edit device data and add attachments such as icons. |

### <UC.005> Format output as CAEX version 2.15/3.0

|  |  |
| --- | --- |
| **Related Business Process:** | <BP.003>: Export Device |
| **Use Cases Objective:** | Make the export to CAEX formats possible for devices. |
| **System Boundary:** | The application itself |
| **Precondition:** | The user has added/loaded a device. |
| **Postcondition on success:** | The user added/loaded at least one device successfully. |
| **Beteiligte Nutzer:** | Every end-user of the application. |
| **Triggering Event:** | When the user wants to save a device in the CAEX format. |

## Features

In diesem Abschnitt sollen die bereits definierbaren funktionalen Features in „Balzert-Notation“ aufgelistet werden und nach den Regeln der Anforderungsschablone ausformuliert werden. à (<https://www.sophist.de/fileadmin/SOPHIST/Puplikationen/Broschueren/SOPHIST_Broschuere_MASTeR.pdf>)

### /LF10/ Import

The application should be able to import a file by the absolute path to the file.

### /LF20/ File validation

The system shall be able to detect wrongly formatted imported files and throw an error to the user.

### /LF30/ Error handling

The system shall be able to handle errors (unexpected shut down, wrongly formatted files, ...)

and throw an error to the user.

### /LF40/ GUI

The system should display a graphical user interface after startup. The user will interact with this GUI for every other functionality of the application

### /LF50/ Display device in a readable way

When a device is loaded or created the attributes of the element should be displayed directly and easily readable for the user.

### /LF60/ Edit device

When the attributes of a loaded device are displayed to the user, the user should be able to edit every attribute he wants to change.

### /LF70/ Create device

When the application is started, the user should be able to create a new and empty device model.

### /LF80/ Export device

When the user has edited a device, he should be able to save the device to a file.

# Product Data

In diesem Abschnitt werden die Hauptdaten und Datenschnittstellen beschrieben, mit denen das Softwareprodukt arbeiten soll und die bereits identifizierbar sind (siehe Abb. 1). Im Allgemeinen werden diese Hauptdaten eines Programms auch nonvolatil gespeichert.

### /LD10/ AML-DD

The system shall create a valid AML-DD with all the necessary information the original file contained. Including an AML root file complete with versioning header, a “SystemUnitClass” with logical description, identification and configuration parameters as well as a reference to the original file and pictures if there are any.

### /LD20/ Save and load files

The system shall make a device description editable. For this, the user can load existing or create new devices within the interface of the application. After editing them, these descriptions can be saved to a location selected by the user.

# Other Product Characteristics

This section describes the already known non-functional requirements for the product.

…

Die Aufgabe dieses Abschnittes ist die Beschreibung der nicht-funktionalen Anforderungen. Dabei handelt es sich um Charakteristiken oder Qualitäten, die das Produkt attraktiv machen und es von vergleichbaren Produkten unterscheiden.

In diesem Abschnitt werden die wesentlichen Eigenschaften des zu entwickelnden Produktes beschrieben, die nicht direkt die zu leistende Funktionalität betreffen.

In diesem Abschnitt sollen diese bereits definierbaren Anforderungen in „Balzert-Notation“ aufgelistet werden und nach den Regeln der Anforderungsschablone ([*https://www.sophist.de/fileadmin/SOPHIST/Puplikationen/Broschueren/SOPHIST\_Broschuere\_MASTeR.pdf*](https://www.sophist.de/fileadmin/SOPHIST/Puplikationen/Broschueren/SOPHIST_Broschuere_MASTeR.pdf)) ausformuliert werden.

### /NF10/ Installation Wizard

The software shall be installable by a graphical installer. The installer should be able to upgrade existing installations on the system.

### /NF20/ GUI

The system shall display a graphical user interface (GUI) to the user. This GUI has to display every function the application provides to the user. It will be the only way to interact with the application.

### 

### /NF30/ System Environment

The application shall be used on Windows 10. The limitations of the environment are identical to the limitations of the .NET Framework.

Hier sollten alle wesentlichen und notwendigen Parameter der Systemumgebung (Hardware, Software) beschrieben werden, soweit diese bereits festlegbar ist.

# References

[1] …

[2] …