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
0.2	31.10.2021	Linus Eickhoff	Goal description added
0.3	11.11.2021	Malte Horst	Removed comments
1.0	06.05.2022	Linus Eickhoff	Checked

CONTENTS

1. 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.

2. 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.

3. Product Usage

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

3.1. Business Processes

3.1.1. <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

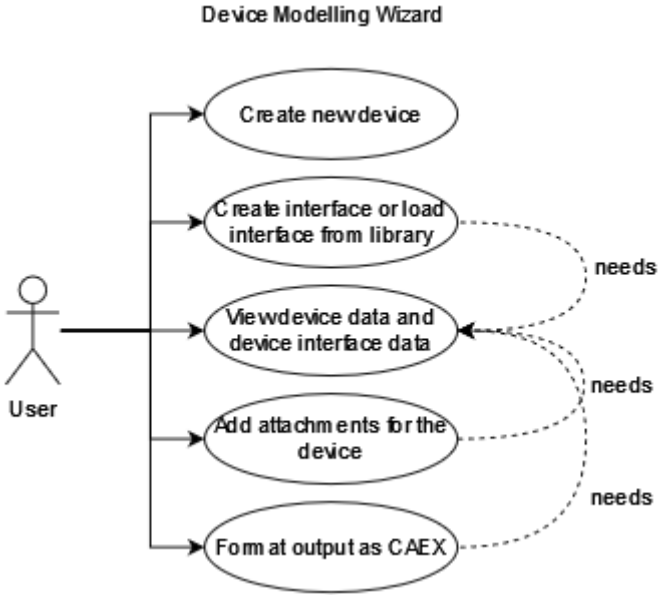
3.1.2. <BP.002>: Edit Device

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

3.1.3. <BP.003>: Export Device

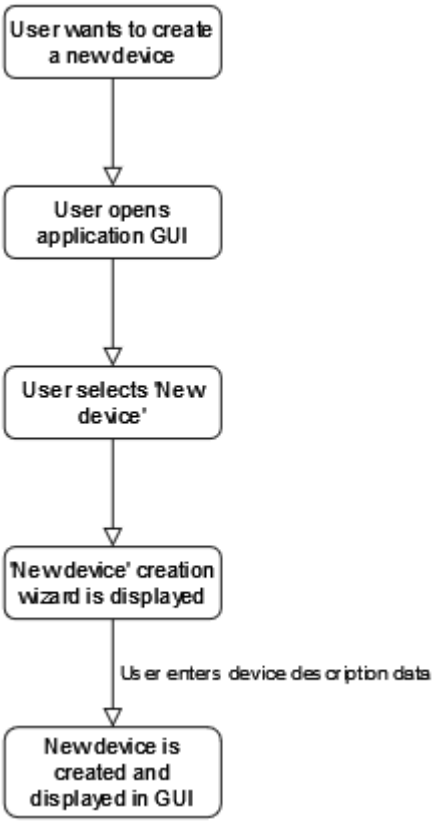
Triggering Event:	The user wants to export a created/edited device
Result:	The device in the application will be saved to a file
Involved Roles:	User and Device Modelling Application

3.2. Use Cases



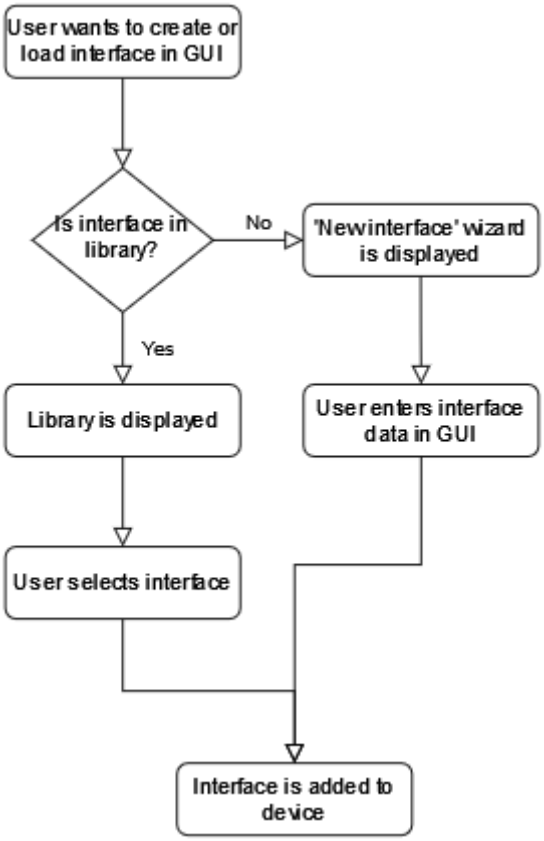
3.2.1. <UC.001> Create new device

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



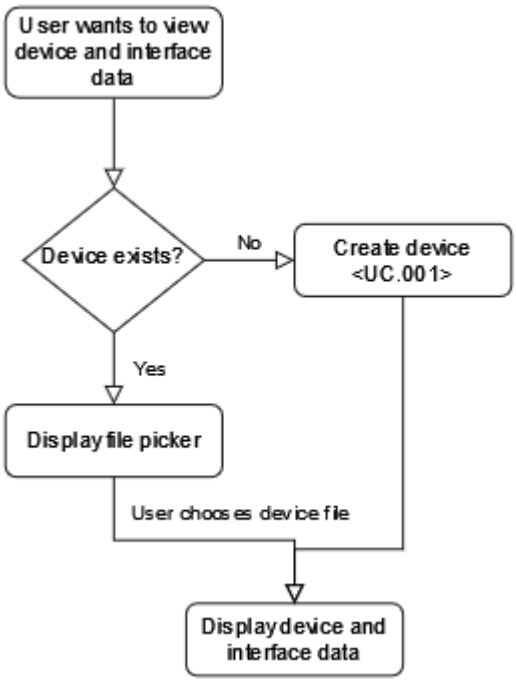
3.2.2. <UC.002> Create interface or load interface from Library

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.



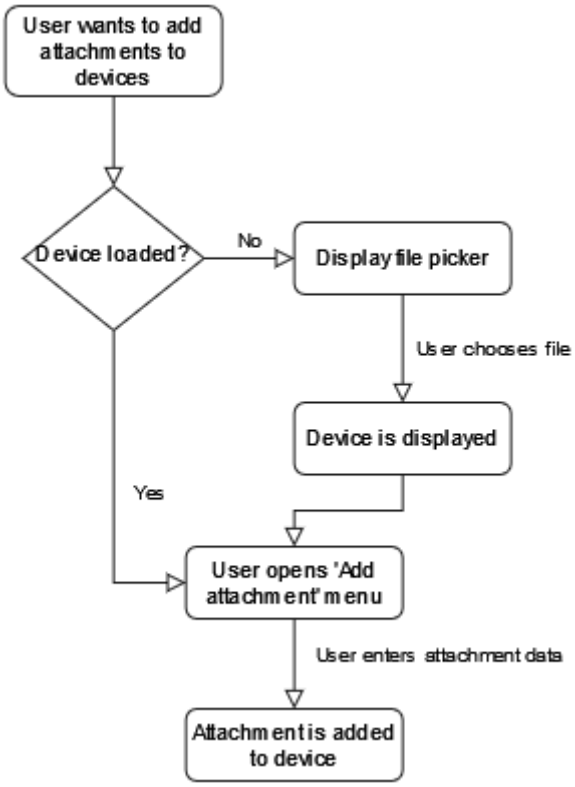
3.2.3. <UC.003> View device data and device interface data

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.



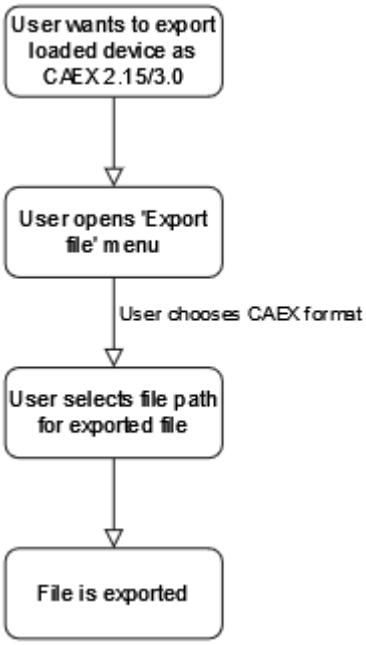
3.2.4. <UC.004> Add Attachments for the Device

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.



3.2.5. <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.


User wants to export loaded device as CAEX 2.15/3.0
User opens 'Export file' menu
User chooses CAEX format
User selects file path for exported file
File is exported

3.3. Features

3.3.1. /LF10/ Import

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

3.3.2. /LF20/ File validation

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

3.3.3. /LF30/ Error handling

The system shall be able to handle errors (unexpected shut down, wrongly formatted files, ...)
and throw an error to the user.

3.3.4. /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

3.3.5. /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.

3.3.6. /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.

3.3.7. /LF70/ Create device

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

3.3.8. /LF80/ Export device

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

4. Product Data

4.1.1. /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.

4.1.2. /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.

5. Other Product Characteristics

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

5.1.1. /NF10/ Installation Wizard

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

5.1.2. /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.

5.1.3. /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.