DHBW
Duale Hochschule
Baden-Württemberg
Stuttgart

System Requirements Specification

(TINF21C, SWE)

Project: Modelling Wizard Improvements

Customer: Markus Rentschler
Christian Holder

Team:

Project Manager	– Robin Ziegler (inf21100@lehre.dhbw-stuttgart.de)
Developer	– Nils Hoffmann (inf21194@lehre.dhbw-stuttgart.de)
Test Manager	– Michael Grote (inf21111@lehre.dhbw-stuttgart.de)
System Architect	– Fabian Kreuzer (inf21106@lehre.dhbw-stuttgart.de)
Tech. Documentation	– Dana Frey (inf21099@lehre.dhbw-stuttgart.de)
Product Manager	– Maximilian Trumpp (inf21123@lehre.dhbw-stuttgart.de)

Change History

Version	Date	Author	Comment
0.1	07.10.2022	Dana Frey	created
0.2	14.10.2022	Dana Frey	Added ‘Introduction’ and ‘Use Cases’
0.3	17.10.2022	Dana Frey	Added ‚Features‘
0.4	19.10.2022	Dana Frey	reworks
0.5	20.10.2022	Dana Frey	Added ‚Enhancements‘ and did reworks
1.0	10.05.2023	Dana Frey	Updates for the final version

Table of Contents

1. Introduction

Our project deals with the already existing program called „Modelling Wizard“. The program can create a device model in a graphical user interface (GUI). It is possible to add device interfaces (such as physical ports) or file attachments to the created device model. If the user already owns descriptive files, those files can be used to create a device model as well. The current version of the Modelling Wizard is capable of handling AMLX, AML, EDZ, IODD and GSD import.

The goal of our project is to create a more user-friendly version of the Modelling Wizard which includes an improved version of the GUI as well as fixing bugs and refactoring the code by the team beforehand.

1.1 Product Environment

AutomationML (AML) is the short form of Automation MarkUp Language and is used to describe parts of automation plants as objects. These objects can consist of multiple other objects and can 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 the following 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.
2. COLLADA to describe the geometry and 3D Models of an object
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 behavior and states if objects, action-sequences and I/O connections are implemented via this format. An IODD (IO Device Description) file describes the sensor and actuator of a plant or component. It also contains information on identify, parameters, process data, communication and more. It is written in XML-format, same as AML, which ensures a conversion.

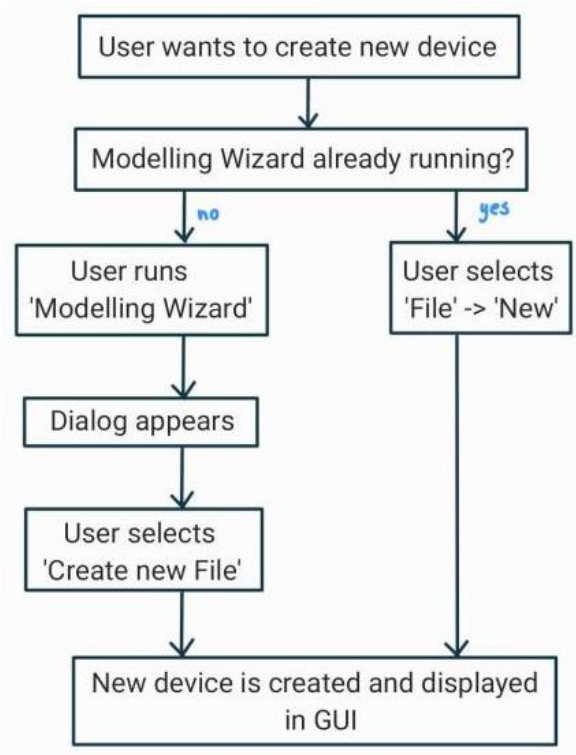
2. Use Cases

Our project doesn't include new added Use Cases and focuses more on the improvement of the given project. Therefore, the following Use Cases have been taken over by the previous team to still be able to visualize the current program's Use Cases.

2.1 <UC.001> Create new device

Use Case's Objective:	User wants to create a device
System Boundary:	The application itself

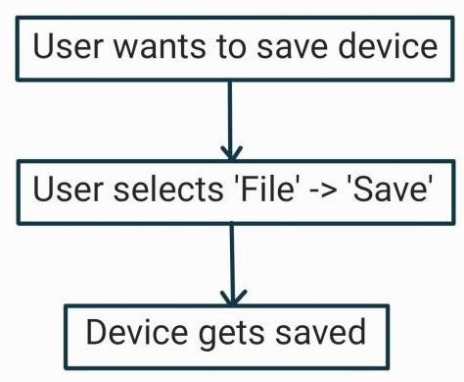
Precondition:	The user needs to have the minimal required data for the device on hand. The program needs to be installed on the user's system and opened.
Postcondition on success:	The entered data is displayed completely and correctly
Involved Users:	Every end-user of the application
Triggering Event:	When the user opens the application and uses the 'New' function to create a new device



2.2 <UC.002> Save device

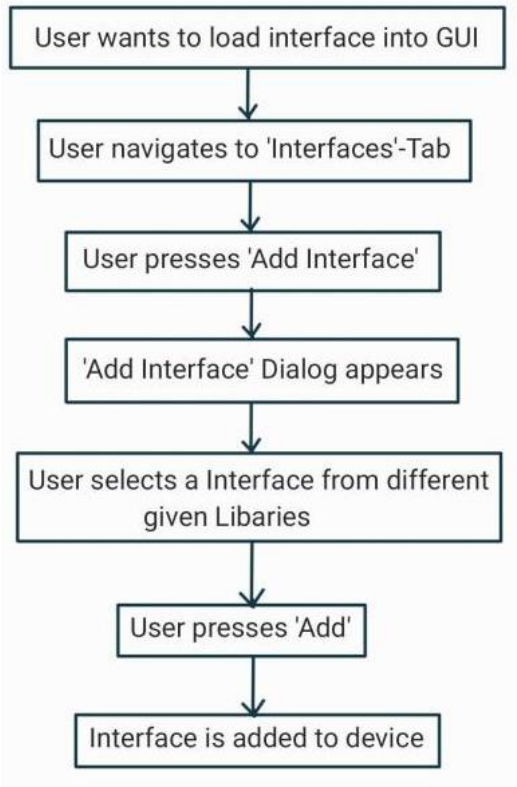
Use Case's Objective:	User wants to save a device by using the 'Save' function
System Boundary:	The application itself
Precondition:	The user created a device
Postcondition on success:	The user created or loaded at least one device successfully
Involved Users:	Every end-user of the application

Triggering Event:	When the user has an opened device, which the user wants to save



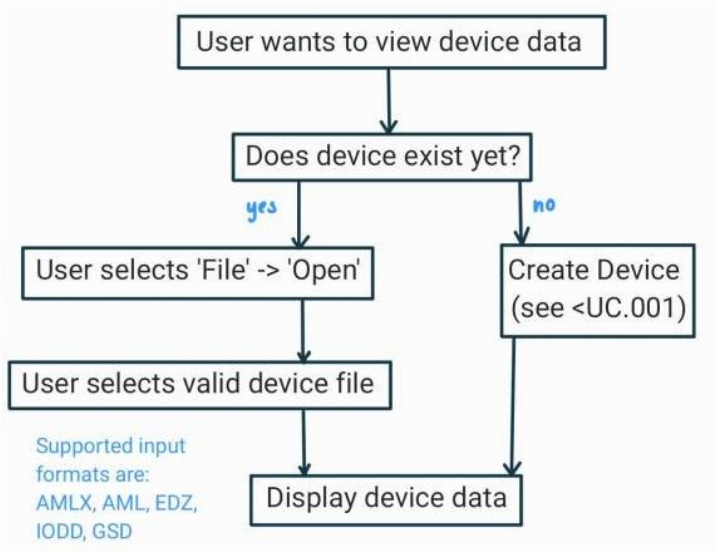
2.3 <UC.003> Create interface from existing library

Use Case's Objective:	User wants to create a device interface by adding 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 successfully added an interface to his device
Involved Users:	Every end-user of the application
Triggering Event:	When the user wants to add a device interface



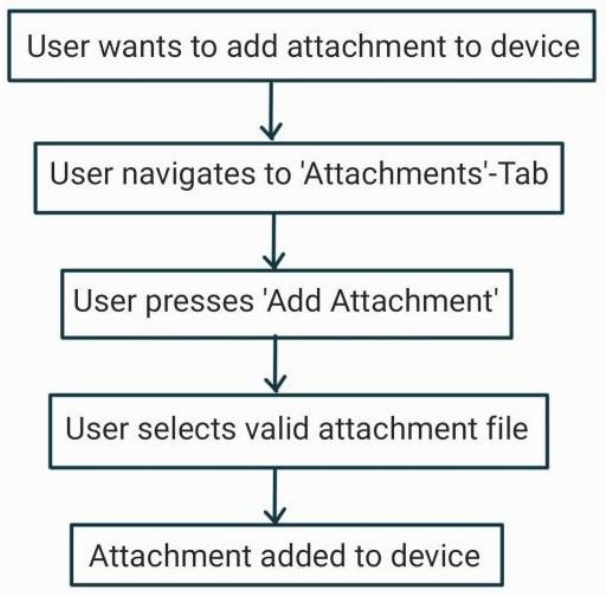
2.4 <UC.004> View device data

Use Case's Objective:	After at least one device was successfully created, the device data should be visible and editable on the user interface
System Boundary:	The application itself
Precondition:	The user created or loaded a device
Postcondition on success:	The user created or loaded at least one device successfully and its data is visible on the screen
Involved Users:	Every end-user of the application
Triggering Event:	When the user wants to view device data



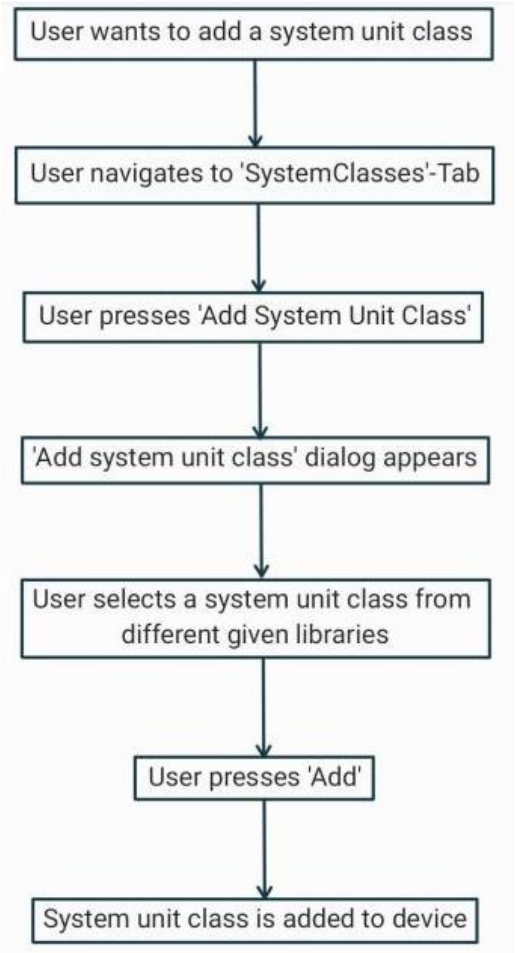
2.5 <UC.005> Add attachments to device

Use Case's Objective:	It is possible to add an attachment to the object, such as a manufacturer's icon
System Boundary:	The application itself
Precondition:	The user created or loaded a device
Postcondition on success:	The user has successfully added an attachment to his device
Involved Users:	Every end-user of the application
Triggering Event:	When the user has the need to edit device data and add attachments such as icons.



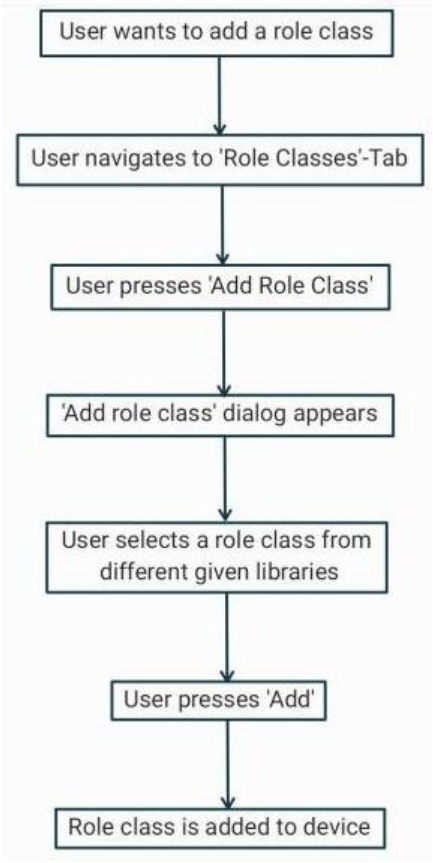
2.6 <UC.006> Add system unit classes to device

Use Case's Objective:	User wants to add a system unit class to existing device
System Boundary:	The application itself
Precondition:	The user created or loaded a device
Postcondition on success:	The user has successfully added a system unit class to his device
Involved Users:	Every end-user of the application
Triggering Event:	When the user has the need to add a system unit class to his device



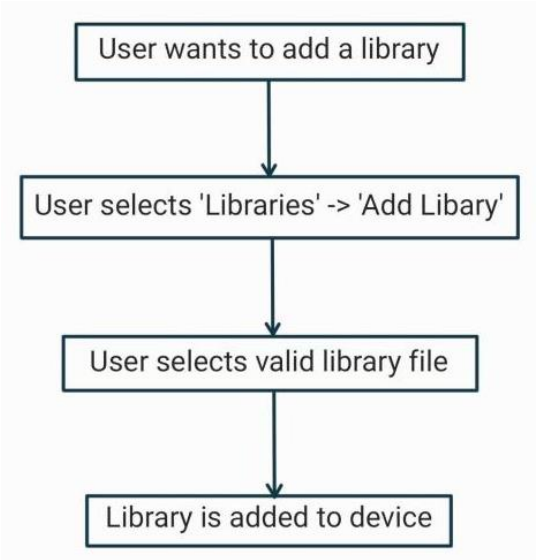
2.7 <UC.007> Add role class to device

Use Case's Objective:	User wants to add a role class to existing device
System Boundary:	The application itself
Precondition:	The user created or loaded a device
Postcondition on success:	The user has successfully added a role class to his device
Involved Users:	Every end-user of the application
Triggering Event:	When the user has the need to add a role class to his device



2.8 <UC.008> Add individual library

Use Case's Objective:	User wants to add an individual library to application
System Boundary:	The application itself
Precondition:	The user created or loaded a device
Postcondition on success:	The user has successfully added an individual library to application
Involved Users:	Every end-user of the application
Triggering Event:	When the user has the need to add an own library



3. Functional Features

3.1.1 <LF11> Import

The application should be able to import a file by the absolute path to the file. This import supports files of the file types AMLX, AML, EDZ, IODD and GSD.

3.1.2 <LF12> File validation

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

3.1.3 <LF13> Error handling

The system shall be able to handle errors (unexpected shut down, wrongly formatted files, ...) and throw an error to the user. If an error appears to be in a specific part of the program, the cursor should be pointed to that exact location.

3.1.4 <LF14> Unsaved changes

When the user has edited a device without saving yet, he should be informed, that there are non-saved changes.

3.1.5 <LF15> 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.

3.1.6 <LF16> Create device

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

3.1.7 <LF17> Export device

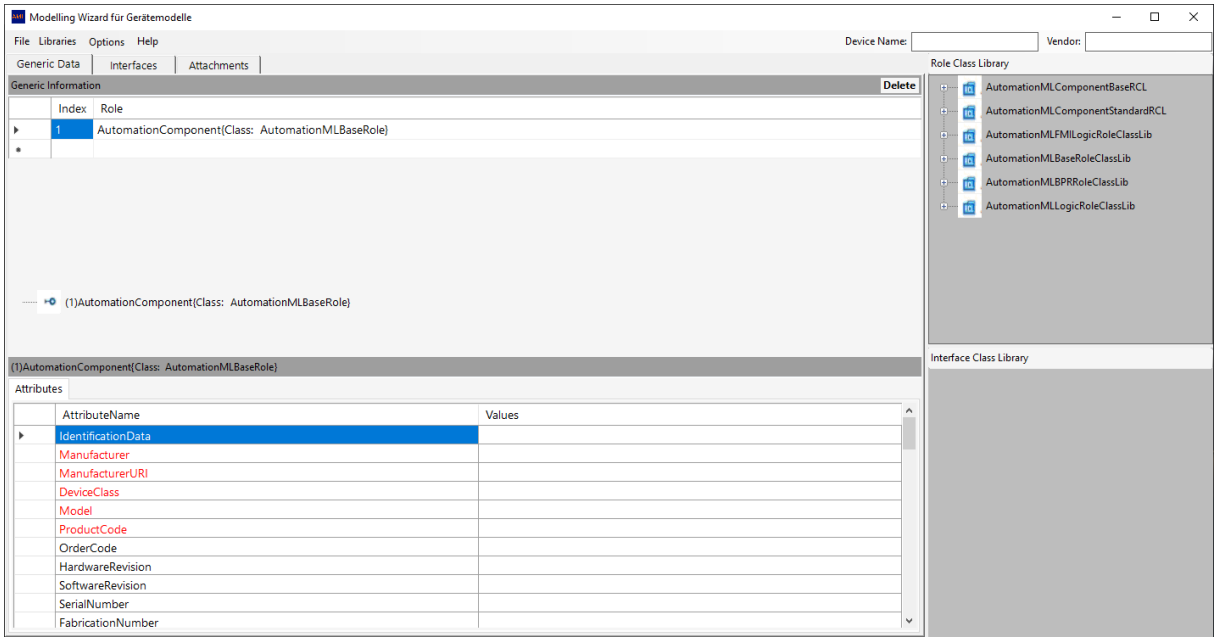
When the user has edited a device, he should be able to save the device to a file.

4. Non-functional Features

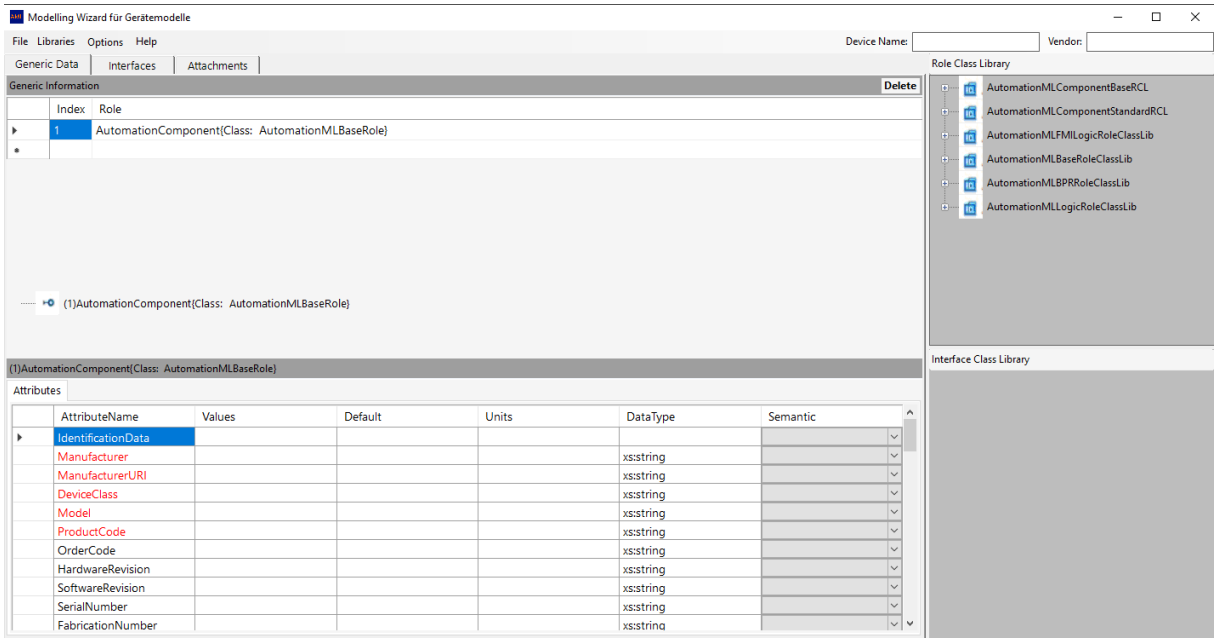
4.1.1 <NF11> GUI

The system should display a graphical user interface after startup of the standalone application. The user will interact with this GUI for every other functionality of the application. The first prototype includes the changes in the existing UI. The final prototype is based on a complete redesign of the old interface.

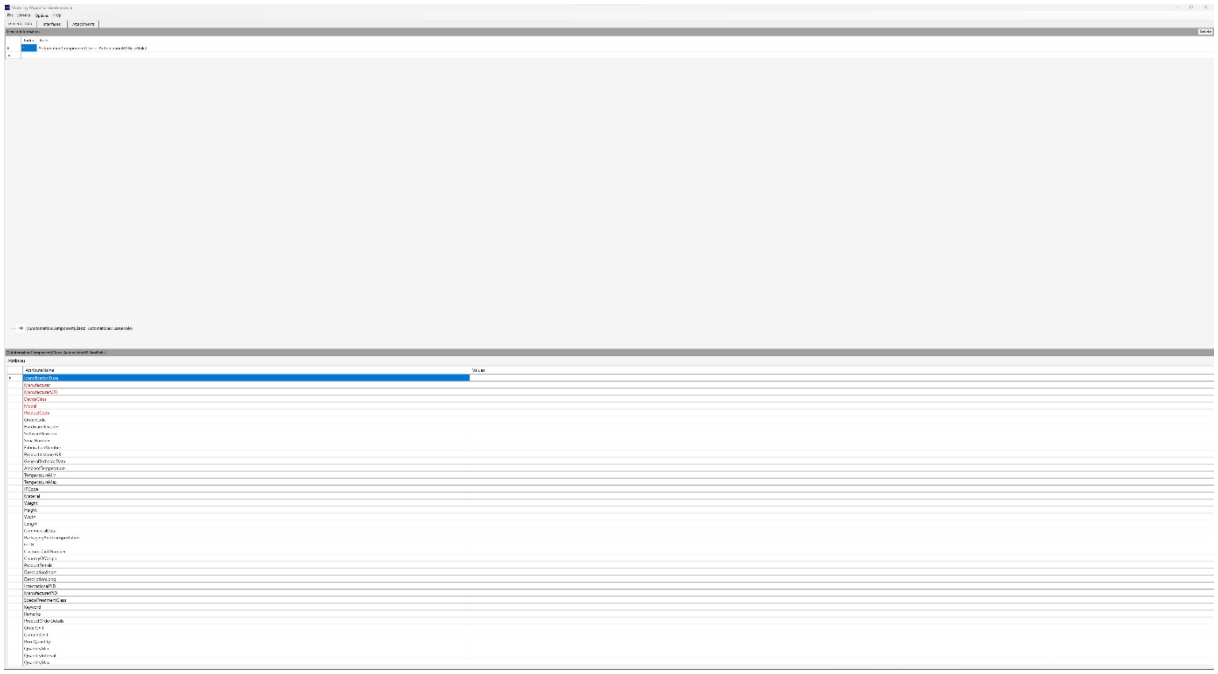
Old GUI from the previous Modelling Wizard team (in easy mode):



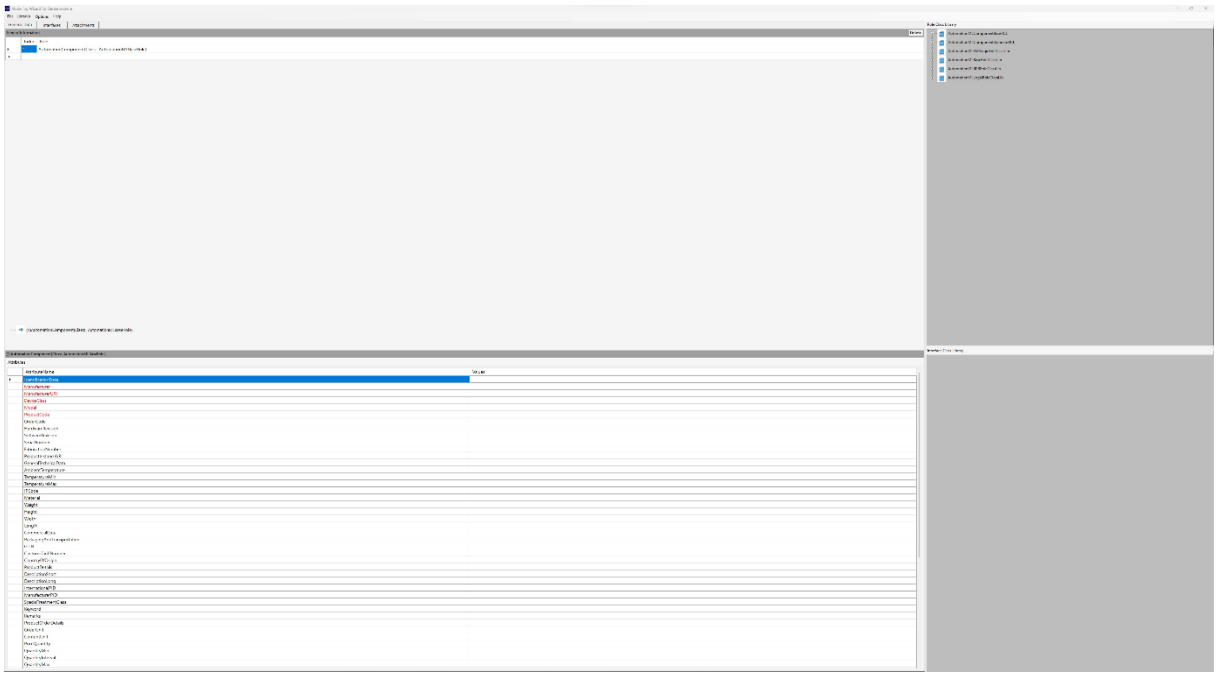
Old GUI from the previous Modelling Wizard team (in "advanced" mode = expert mode):



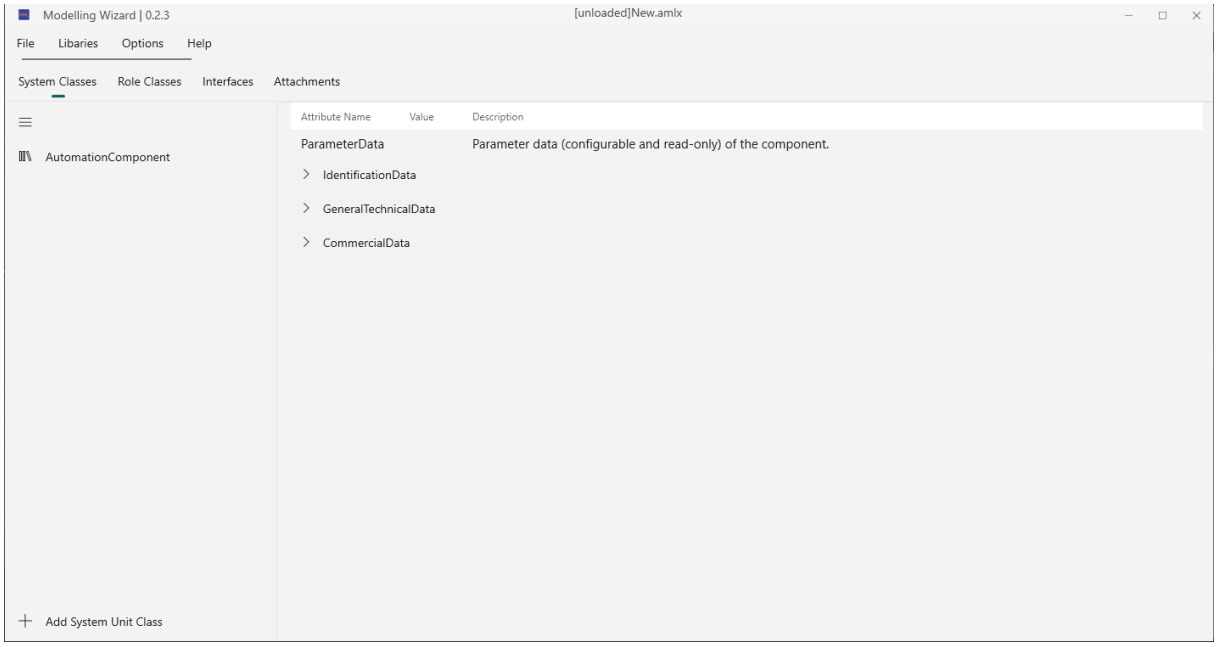
First prototype of GUI (in easy mode):



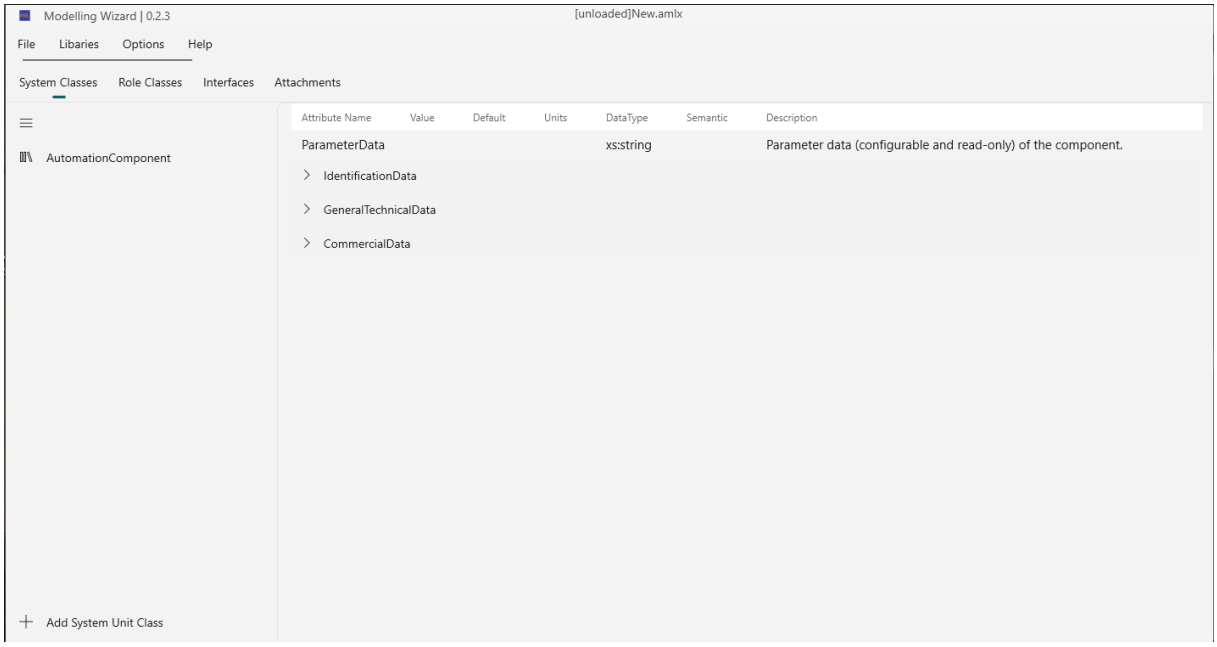
First prototype of GUI (in expert mode):



Final GUI version (in easy mode):



Final GUI version (in expert mode):



The graphical user interface should be user friendly. There should not be an overwhelming amount of information, button areas should easily be clickable, there should be no unnecessary buttons, etc..

4.1.2 <NF12> 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.

4.1.3 <NF13> Easy Mode

When the user wants to hide additional attribute information visible in the "System Classes"-Tab.

4.1.4 <NF14> Expert Mode

When the user wants to display additional attribute information, which was not visible in the "System Classes"-Tab before.

4.1.5 <NF15> Portable

The program should be runnable without any installation.

4.1.6 <NF16> Performance

The application should respond instantly after a user's action.

4.1.7 <NF17> Compatibility

The application should be executable on every current system such as Windows 10 or higher. Furthermore, the application is only executable on the Windows platform.

5. Bug fixes

5.1 <BUG10> Data Models

https://github.com/robinziegler/TINF21C_Team4_Modelling_Wizard_Improvements/issues/12

5.2 <BUG20> Open Saved Files

https://github.com/robinziegler/TINF21C_Team4_Modelling_Wizard_Improvements/issues/5

5.3 <BUG30> Delete without Confirmation

https://github.com/robinziegler/TINF21C_Team4_Modelling_Wizard_Improvements/issues/27

5.4 <BUG40> Deleted Interface

https://github.com/robinziegler/TINF21C_Team4_Modelling_Wizard_Improvements/issues/28

5.5 <BUG50> List View for Attributes

https://github.com/robinziegler/TINF21C_Team4_Modelling_Wizard_Improvements/issues/75

5.6 <BUG60> Separate Menu Bar

https://github.com/robinziegler/TINF21C_Team4_Modelling_Wizard_Improvements/issues/72

5.7 <BUG70> Visual Displacement

https://github.com/robinziegler/TINF21C_Team4_Modelling_Wizard_Improvements/issues/71

5.8 <BUG80> Text Colours removed

https://github.com/robinziegler/TINF21C_Team4_Modelling_Wizard_Improvements/issues/68

5.9 <BUG90> Theme changes Dialog

https://github.com/robinziegler/TINF21C_Team4_Modelling_Wizard_Improvements/issues/66

5.10 <BUG100> Wrong Colouring

https://github.com/robinziegler/TINF21C_Team4_Modelling_Wizard_Improvements/issues/62

5.11 <BUG110> Semantic

https://github.com/robinziegler/TINF21C_Team4_Modelling_Wizard_Improvements/issues/60

5.12 <BUG120> Content Dialog size

https://github.com/robinziegler/TINF21C_Team4_Modelling_Wizard_Improvements/issues/52

5.13 <BUG130> Visual Feedback

https://github.com/robinziegler/TINF21C_Team4_Modelling_Wizard_Improvements/issues/134

5.14 <BUG140> Crash while Edit

https://github.com/robinziegler/TINF21C_Team4_Modelling_Wizard_Improvements/issues/130

5.15 <BUG150> Attribute Highlight

https://github.com/robinziegler/TINF21C_Team4_Modelling_Wizard_Improvements/issues/115

5.16 <BUG160> Library Visualization

https://github.com/robinziegler/TINF21C_Team4_Modelling_Wizard_Improvements/issues/114

5.17 <BUG170> Local Storage

https://github.com/robinziegler/TINF21C_Team4_Modelling_Wizard_Improvements/issues/97

5.18 <BUG180> Wrong Text

https://github.com/robinziegler/TINF21C_Team4_Modelling_Wizard_Improvements/issues/82

5.19 <BUG190> Window Refactoring

https://github.com/robinziegler/TINF21C_Team4_Modelling_Wizard_Improvements/issues/132

5.20 <BUG200> Dialog Cancel

https://github.com/robinziegler/TINF21C_Team4_Modelling_Wizard_Improvements/issues/133

5.21 <BUG210> Change Filename

https://github.com/robinziegler/TINF21C_Team4_Modelling_Wizard_Improvements/issues/135

5.22 <BUG220> Data Grid size

https://github.com/robinziegler/TINF21C_Team4_Modelling_Wizard_Improvements/issues/136

5.23 <BUG230> Update Status

https://github.com/robinziegler/TINF21C_Team4_Modelling_Wizard_Improvements/issues/137

6. Enhancements

6.1 <ENH10 > Data Models

https://github.com/robinziegler/TINF21C_Team4_Modelling_Wizard_Improvements/issues/13

6.2 <ENH20 > Device Name

https://github.com/robinziegler/TINF21C_Team4_Modelling_Wizard_Improvements/issues/11

6.3 <ENH30 > Easy Mode

https://github.com/robinziegler/TINF21C_Team4_Modelling_Wizard_Improvements/issues/10

6.4 <ENH40 > Error Handling

https://github.com/robinziegler/TINF21C_Team4_Modelling_Wizard_Improvements/issues/9

6.5 <ENH50 > Remove Input Fields

https://github.com/robinziegler/TINF21C_Team4_Modelling_Wizard_Improvements/issues/

6.6 <ENH60 > Save Buttons

https://github.com/robinziegler/TINF21C_Team4_Modelling_Wizard_Improvements/issues/7

6.7 <ENH70 > Open and Import

https://github.com/robinziegler/TINF21C_Team4_Modelling_Wizard_Improvements/issues/6

6.8 <ENH80> Load Library Repositioning

https://github.com/robinziegler/TINF21C_Team4_Modelling_Wizard_Improvements/issues/40

6.9 <ENH90> Mode Indicator

https://github.com/robinziegler/TINF21C_Team4_Modelling_Wizard_Improvements/issues/41

6.10 <ENH100> Remove About

https://github.com/robinziegler/TINF21C_Team4_Modelling_Wizard_Improvements/issues/42

6.11 <ENH110> Attributes Button

https://github.com/robinziegler/TINF21C_Team4_Modelling_Wizard_Improvements/issues/43

6.12 <ENH120> Remove Interface Class Library

https://github.com/robinziegler/TINF21C_Team4_Modelling_Wizard_Improvements/issues/44

6.13 <ENH130> Remove Column

https://github.com/robinziegler/TINF21C_Team4_Modelling_Wizard_Improvements/issues/45

6.14 <ENH140> Remove Table Display

https://github.com/robinziegler/TINF21C_Team4_Modelling_Wizard_Improvements/issues/46

6.15 <ENH150> Edit Data Types

https://github.com/robinziegler/TINF21C_Team4_Modelling_Wizard_Improvements/issues/47

6.16 <ENH160> Delete Option

https://github.com/robinziegler/TINF21C_Team4_Modelling_Wizard_Improvements/issues/48

6.17 <ENH170> Remove Textfields

https://github.com/robinziegler/TINF21C_Team4_Modelling_Wizard_Improvements/issues/49

6.18 <ENH180> Information Texts

https://github.com/robinziegler/TINF21C_Team4_Modelling_Wizard_Improvements/issues/50

6.19 <ENH190> Dialog

https://github.com/robinziegler/TINF21C_Team4_Modelling_Wizard_Improvements/issues/51

6.20 <ENH200> Adding Interfaces

https://github.com/robinziegler/TINF21C_Team4_Modelling_Wizard_Improvements/issues/53

6.21 <ENH210> Add Table

https://github.com/robinziegler/TINF21C_Team4_Modelling_Wizard_Improvements/issues/54

6.22 <ENH220> AML Files

https://github.com/robinziegler/TINF21C_Team4_Modelling_Wizard_Improvements/issues/57

6.23 <ENH230> Attachments

https://github.com/robinziegler/TINF21C_Team4_Modelling_Wizard_Improvements/issues/58

6.24 <ENH240> Dark Light Mode

https://github.com/robinziegler/TINF21C_Team4_Modelling_Wizard_Improvements/issues/61

6.25 <ENH250> Title Bar

https://github.com/robinziegler/TINF21C_Team4_Modelling_Wizard_Improvements/issues/63

6.26 <ENH260> Update Tables

https://github.com/robinziegler/TINF21C_Team4_Modelling_Wizard_Improvements/issues/64

6.27 <ENH270> Replace List View

https://github.com/robinziegler/TINF21C_Team4_Modelling_Wizard_Improvements/issues/70

6.28 <ENH280> About Dialog

https://github.com/robinziegler/TINF21C_Team4_Modelling_Wizard_Improvements/issues/73

6.29 <ENH290> Filename

https://github.com/robinziegler/TINF21C_Team4_Modelling_Wizard_Improvements/issues/76

6.30 <ENH300> Unsaved Changes

https://github.com/robinziegler/TINF21C_Team4_Modelling_Wizard_Improvements/issues/77

6.31 <ENH310> Value Validation

https://github.com/robinziegler/TINF21C_Team4_Modelling_Wizard_Improvements/issues/78

6.32 <ENH320> Start Dialog

https://github.com/robinziegler/TINF21C_Team4_Modelling_Wizard_Improvements/issues/79

6.33 <ENH330> Add Role Class

https://github.com/robinziegler/TINF21C_Team4_Modelling_Wizard_Improvements/issues/85

6.34 <ENH340> Visual Loading

https://github.com/robinziegler/TINF21C_Team4_Modelling_Wizard_Improvements/issues/88

6.35 <ENH350> Save Options

https://github.com/robinziegler/TINF21C_Team4_Modelling_Wizard_Improvements/issues/89

6.36 <ENH360> Automatic Expansion

https://github.com/robinziegler/TINF21C_Team4_Modelling_Wizard_Improvements/issues/112

6.37 <ENH370> NavBar Icons

https://github.com/robinziegler/TINF21C_Team4_Modelling_Wizard_Improvements/issues/113

6.38 <ENH380> Back Arrow

https://github.com/robinziegler/TINF21C_Team4_Modelling_Wizard_Improvements/issues/116

7. References

[1] https://github.com/H4CK3R-01/TINF20C_ModellingWizard_Devices/wiki/1.-Software-Requirements--Specification#UC3

[2] Software Engineering lessons by Markus Rentschler and Christian Holder

8. Glossary

AML	Automation Markup Language is an open standard data format for storing and exchanging plant planning data
AMLX	AML Package
CAEX	Computer-Aided Engineering Exchange
EDZ	EPLAN Electric P8 Data Archive Zipped File
GSD	General Station Description
GUI	Graphical User Interface
IODD	Input/Output Device Description