**Customer Requirements Specification**

**(Lastenheft)**

(TINF20C, SWE I Praxisprojekt 2021/2021)

Project: Modelling Wizard for Cable-Models

Customer: Rentschler & Holder

Rotebühlplatz 41

70178 Stuttgart

Supplier: Team 2 (Amtmann Leon, Thomé Fabian, Friedrich Calvin, Rausch Thorsten, Pauer Kevin, Sellemann Tim)

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Author** | **Comment** |
| 0.1 | 21.10.2021 | Max Gohlke | First Draft oft the CRS |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

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

CONTENTS

*1.* Goal 3

2. Product Environment 4

*3.* Product Usage 5

*3.1.* Business Processes 5

3.1.1. <BP.001>: <Name> 5

3.2. Use Cases 5

3.2.1. <UC.001> Use Case Name 6

*3.3.* Features 7

3.3.1. /LF10/ ….. 7

3.3.2. /LF20/ ….. 7

*4.* Product Data 8

4.1. /LD10/ ….. 8

4.2. /LD20/ ….. 8

*5.* Other Product Characteristics 9

5.1. /NF10/ ….. 9

5.2. /NF20/ ….. 9

5.3. System Environment 9

6. References 10

# Goal

A webbased application should be developed, that provides an accessible and easy GUI for the configuration of a cable and the addition of device-interfaces (for example physical ports) and file-attachements [1]. Such configurations can be found in the catalogues of almost every cable-provider. A usability-analysis should give insight to existing solutions, the following results then influence the development. The file-output format should be in form of a AutomationML-Package, which must apply to the rules for AML-Componentmodels (AML-DDs). Furthermore, the electrical interface-library should be expanded with new connector-types for single-pair-ethernet from the IEC63171-6 and M12-Push/Pull from the IEC 61076-2 010.

Following tasks must be completed:

1. Analysis of existing solutions (z.B. Balluff, Murr-Elektronik, Harting, Phoenix-Contact)

2. Mockup of a GUI and an appropriate usability-concept

3. Support for CAEX 2.15 and CAEX 3.0 as output format (customizable)

4. Input fields for all neccessary parameters

5. Defintion of an exchangable data-model for the product-logic

6. Use and expansion of the AML-interface-library for electrical connectors

7. Use of the ANGULAR-framework for the GUI-creation

8. Creation of an extensive user-documentation

[1] 2021\_Steckverbinderkongress\_Rentschler\_1v0.pdf

[2] <https://www.automationml.org/download-archive/>

[3] IEC 61076-2-010

# Product Environment

The usage environment of the modelling wizard for cable-models is characterized by its web-GUI for cable-onlineshops. This web-GUI is implemented with ANGULAR.

ANGULAR is a TypeScript (JavaScript deviation) based front-end framework for web development.

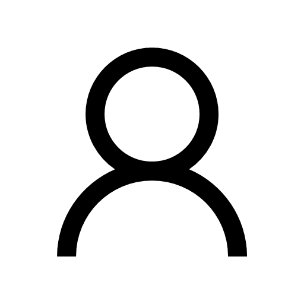
An important aspect of the software is the cable-configuration, that users use to configure their cables within the browser interface. Users can download the current configuration in form of an AutomationML-package that applies to the rules for AML-Componentmodels (AML-DDs).

AutomationML is a neutral, XML based data format for the storage and exchange of planning data and an open standard.

With this AutomationML-package that the users can export, they can store their configuration locally. Later when they need to access or share their configuration, they can import the AutomationML-package back into the web-interface or send it to another user.

The modelling wizard shall use the electrical interface-library for all the cables and connectors. Because the common new connector-types for single-pair-ethernet from the IEC63171-6 and M12-Push/Pull from the IEC 61076-2-010 are not in this library, we will expand it with these new connectors.

**Online Shop for Cables**



**Cable Modelling**

**Web-Interface for Cable Modelling**

**(Angular)**

**AutomationML-package**

**User**

Figure 1: Product Environment

# Product Usage

The Modelling Wizard for Cables should support the creation of new cable models with additional information for each created cable model, such as the number of contacts, layout of contacts and common connector names and descriptions.

It should be possible to easily create new cable models as soon as a new cable is registered in the inventory system of a business employing this Modelling Wizard. In this particular case, it does not matter whether the cable was acquired externally or created by R&D internally.

To this end, the Modelling Wizard has to support:

* the creation of new cable models,
* the displaying of existing cable models as an interactive list with details;
  + it will contain clickable entries to view the models in detail,
* the deprecation and deletion of cable models from the display list,
* searching and filtering the list of cables, and
* downloading of cable specifications as an AutomationML-package.

## Business Processes

### <BP.001>: New Cable is Registered in Inventory System

|  |  |
| --- | --- |
| Triggering Event: | Inventory system triggers messaging system to inform Distribution Manager of presence of new cable in the system |
| Result: | New cable model describing the newly added cable is created by the Inventory Control Manager and uploaded to the web server to be indexed and displayed. |
| In Roles: | Inventory Control Manager, Distribution Manager |

### <BP.002>: Cable is Marked as Deprecated in Inventory Management System

|  |  |
| --- | --- |
| Triggering Event: | Inventory system triggers messaging system to inform Distribution Manager of deprecation of cable |
| Result: | Inventory Control Manager deletes the defunct cable from the Cable Wizard web interface. |
| In Roles: | Inventory Control Manager, Distribution Manager |

### <BP.003>: Customer Lookup of Cable Information

|  |  |
| --- | --- |
| Triggering Event: | Customer visits Cable Wizard display page |
| Result: | Customer leaves happily, having found a fitting cable and the right information for their use of the cable |
| In Roles: | Customer |

## Use Cases

Capability

Cable Wizard

Search & Filtering

Cable Creation

File Export

Cable Deprecation

### <UC.001> New cable registration

|  |  |
| --- | --- |
| **Related Business Process:** | <BP.001>: New Cable is registered in inventory system |
| **Use Cases Objective:** | User wants to store information about a new cable in a safe environment where it is easily accessible |
| **System Boundary:** | Inventory system |
| **Precondition:** | The cable must not be already registered, the program has to run without errors. |
| **Postcondition on success:** | The cable is successfully registered with all specifications |
| **Involved Users:** | User and inventory system |
| **Triggering Event:** | The user acquires a new cable which they want to be registered in the inventory system |

### <UC.002> Deprecated cable

|  |  |
| --- | --- |
| **Related Business Process:** | <BP.002>: Cable is marked as deprecated in inventory management system |
| **Use Cases Objective:** | User wants to remove a deprecated cable from the database as it is not longer necessary |
| **System Boundary:** | Inventory system |
| **Precondition:** | The user has to choose an existing cable they want to delete, the program has to run without errors |
| **Postcondition on success:** | The cable model is removed from the inventory system |
| **Involved Users:** | User and inventory system |
| **Triggering Event:** | Due to any reasons the need for a certain cable is no longer given |

### <UC.003> Lookup

|  |  |
| --- | --- |
| **Related Business Process:** | <BP.003>: Customer Lookup of Cable Information |
| **Use Cases Objective:** | User needs a cable to fit their requirements or needs the specifications of a cable they used for a project |
| **System Boundary:** | Search Engine |
| **Precondition:** | The user has to define their demands and there must be a fitting cable already stored in the database, the program has to run without errors |
| **Postcondition on success:** | A fitting cable model is found for the user to view and/or download |
| **Involved Users:** | User and search engine |
| **Triggering Event:** | The user has special requirements or wants to look up the details of an existing cable |

## Features

### /Löschgruppenfahrzeug 10/ Login & User Authentication

The Login & Authentication system shall check if the entered credentials match the credentials saved in the authentication storage location and, in case of failure to authenticate, inform the user and deny access to the cable wizards modification functions.

After too many successive authentication failures, the system administrator will be informed about an unauthorized access attempt.

### /LF20/ Displaying a List of Cable Models in Database

The Cable Wizard will check if there are any cable model files to be displayed and, in case there are none, give an appropriate message to the user. Otherwise, the cables will be displayed in alphabetical order of their respective specified names.

### /LF30/ Deprecation of Cable Model

If the Cable Wizard receives a command for deletion of a cable model, it will check if the command for deletion came from an authenticated user. If not, the cable is not deleted, otherwise the model is purged from the database and the display page is refreshed, now without displaying the deleted cable.

### /LF40/ Searching and Filtering the Cable List

The Cable Wizard will check the list of cables, dynamically purging those from the list that do not fit the search criteria specified in a text box. If the search criteria do not match any cable, the user will be informed. Otherwise, a list of cables matching the search criteria is returned.

### /LF50/ Downloading Cable Specifications

The Cable Wizard will convert the information about a particular cable in the database into a AutomationML-compliant format and transfer that data to the user. If the resulting AutomationML-package fails integrity verification or syntax-checks, the user will be informed. Otherwise the download on the users side will begin.

# Product Data

## Data

The data used for the modelling wizard is being provided by the AutomationML interface library for electrical connectors. AutomationML is a modelling language based on XML, which allows creation, storage and exchange of models used mainly for engineering purposes.

AutomationML uses the data format CAEX, which stands for Computer Aided Engineering Exchange. CAEX is built to store object data in hierarchical order and uses modules and components as base concepts to describe the models of real-life objects. Since AutomationML is built upon CAEX the data output format will support CAEX in both, Version 2.15 and its most recent Version 3.0. The data output itself will be an AutomationML-Package.

## Interfaces

Interfaces needed for the modelling wizard are most importantly the transfer of data provided by the AutomationML interface library to the framework being used, to display the data, which is ANGULAR.js, as well as the transfer of the user’s input back from the framework to the AutomationML interface library.

# Other Product Characteristics

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

## Usability

The website will be compatible with common browsers and devices without limitations in functionality and performance.

The interface shall consist of an easily readable and intuitively understandable layout, with elements conveying their meaning and given functionality to the user.

The interface shall enable an inexperienced to quickly understand how to access and use key features, with the meaning and given functionality of icons, buttons and other means of interaction being self-explanatory.

The software should be accessible for people with disabilities or impairments, such poor eyesight and colorblindness, by using distinguishable colors for highlighting, easily readable fonts and clearly visible icons and imagery.

## Efficiency

The software shall enable the user to put together their desired parts quickly, by requiring low amounts of steps taken by the user to archive their desired result.

The software shall perform individual steps in a minimal amount of time.

The software shall generate and provide the resulting files upon request within a minimal amount of time.

The software shall reduce mental effort required to achieve desired results to a minimum by utilizing a simple and intuitive layout and providing utility features.

## Performance

The software shall load and render on common browsers and devices without disruptive delays.

The software shall occupy low amounts of memory and computing power on user devices to avoid disrupting other applications the user may be running.

The interface shall respond to user inputs without noticeable delay.

The software shall have the capacity to handle large amounts of users interacting with the system simultaneously.

## System Environment

This section describes the system environment required to operate the product.

The website will be able to run on any browser supporting the HTML5 standard.

The system will require a computer with network access to communicate with its users.

# References

[1] …

[2] …