System Requirements Specification

(TINF19C, SWE I, 2020 / 2021)

Project: PLCopen-Editor

Customer: Rentschler & Holder
Rotebühlplatz 41
70178 Stuttgart

Supplier: Team 1 (Franziska Kopp, Leonie de Santis, Mouaz Tabboush, Elian Yildirim)

Version	Date	Author	Comment
0.1	30.10.2020	Elian Yildirim	Created
0.2	01.11.2020	Elian Yildirim	Added Introduction
0.3	02.11.2020	Elian Yildirim	Added functional Requirements
0.4	03.11.2020	Elian Yildirim	Added Non-functional Requirements
0.5	03.11.2020	Elian Yildirim	Added Product Data

Contents

1. Introduction

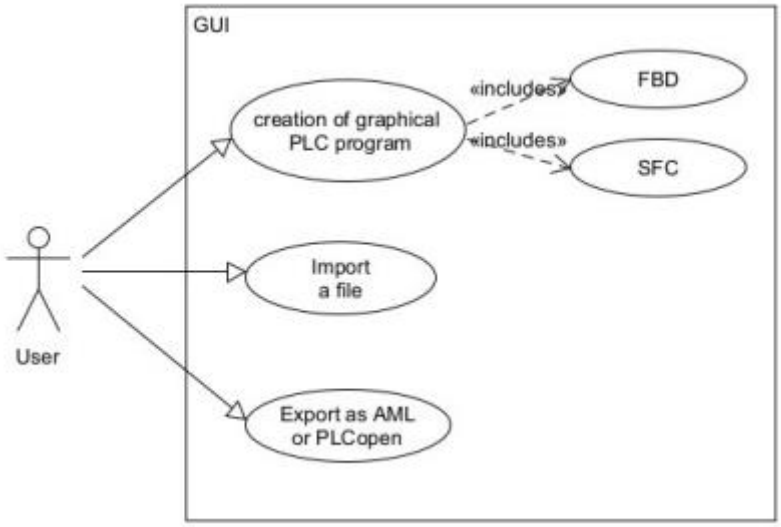
The PLCopen-Editor is an application for the web and is used to create programs for the Open-PLC-Runtime with a graphical user interface. The programs can be compiled by different graphical components and are based on the IEC 61131-3 standard and support the two languages LD (Ladder Logic), FBD (Function Block Diagram), IL (Instruction List), ST (Structured Text) and SFC (Sequential Function Chart). XML files can be exported after the creation of a program. These files compile the assembled components on the user interface to an executable and compatible program. XML files can also be imported into the editor and displayed graphically.

1.1 Product environment

The PLCopen-Editor is used in automation technology for the control and regulation of processes. The programming languages LAD, FBD, AQL, ST and SFC should be supported. The focus, however, should be on the languages FBD and SFC, since these are the graphical representation for describing sequential processes.
The PLCopen-Editor can also display a program in the XML language and thus both export and import corresponding XML files. In addition, a created program can also be exported as AutomationML.

1.2 Use Cases

The users of the PLCopen-Editor use various functions and areas of the editor, which are summarized in the following graphic:



1.2.1 Create PLC program

Related Business Process:	PLC programming
Use Cases Objective:	User wants to create a graphical PLC program. The User can choose between the programming languages FBD and SFC
System Boundary:	Program itself is the system boundary.
Precondition:	The GUI has to run without errors.
Postcondition on success:	If an error occurs, an easily understandable error message should be communicated.
Involved Roles:	User, Editor
Triggering Event:	User opens the GUI

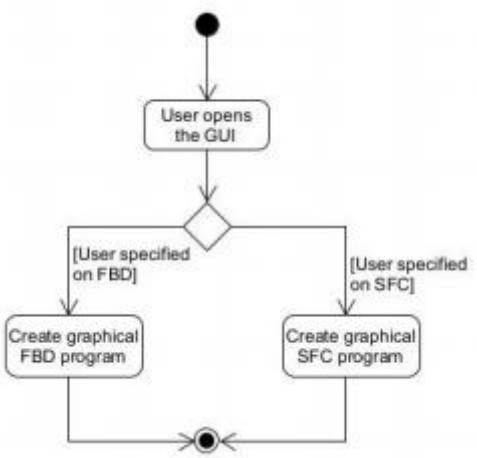


Figure 1: Create PLC program

1.2.2 Import file

Related Business Process:	File conversion
Use Cases Objective:	User wants to import an OpenPLC-XML or AutomationML file to load the PLC-program
System Boundary:	Program itself is the system boundary.
Precondition:	Imported file must be valid and without errors
Postcondition on success:	The PLC program was loaded completely.
Involved Roles:	User, PLCopen/AML file, Importer
Triggering Event:	User opens the GU and want to import a file

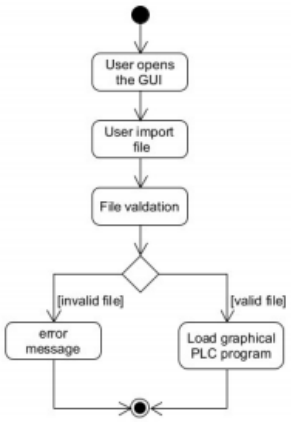


Figure 2: Import File

1.2.3 Export as PLCopen or AML

Related Business Process:	Data storage
Use Cases Objective:	User wants to export the PLC program as PLCopen XML or AutomationML to exchange their program between development environments
System Boundary:	Program itself is the system boundary
Precondition:	Created PLC program must be without errors
Postcondition on success:	The browser shall not be closed until the export is completed
Involved Roles:	PLC program, Exporter
Triggering Event:	User finisched the graphical PLC program and decide to save in PLCopen XML or AutomationML

Figure 3: Export as PLCopen or AML

2. Product Requirements

The following functionalities shall be supported by the PLCopen-Editor.

2.1 GUI Components

The GUI should have a homepage where you can create or load a new project. Furthermore there should be a page for the project overview, where you can add new sections to the project. These sections can then be edited in the editor and "filled" with blocks / respective modules.

2.2 Drag and Drop

The user interface offers through the integration of drag and drop functions a simple and clear editing and creation of the programs with the blocks / respective modules.

2.3 Error handling

The user should be informed via the user interface when the software detects an error so that the error can be corrected.

2.4 Plausibility check

The user should also receive a message via the user interface if the PLC program is faulty or incomplete. The user can then either continue the creation of the program or export the program.

2.5 File format analysis

Furthermore, the user should be informed via the user interface if he wants to import an incompatible XML file into the project.

2.6 File validation

The File validation shall check if the imported file is convertible and corresponds to the PLCopen XML schema or AML. In case of violation, an error message is displayed and the user is asked to import a new file. If the file is valid, it can be imported.

2.7 Languages support

For the time being, the GUI should be designed for the programming languages SFC and FBD. The user should be able to design, import and export programs in these two languages.

2.8 Import PLCxml and AML

The importer should create a graphical PLC program from the imported file. The imported format of the file has to be PLCxml or AML. Otherwise, there would occur an error message. If the file is imported completely, the user can edit the PLC program.

2.9 Export

The exporter should save the created PLC program as PLCopen XML or AML file. If an error occurs, a corresponding error message should be displayed. If the export was completed successfully, the web application can be closed or continued. The exported file can be imported into the PLCopen editor or the AutomationML editor.

3. Non-functional Requirements

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

3.1 Usability

The software should support a graphical user interface. It should be designed clearly and be easy to use, so that users can use the web application without difficulty even the first time they use it.

3.2 License

The software shall be published under the _____ license.

3.3 System Environment

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

4. Product Data

4.1 PLC

The editor shall support at least the PLC programming languages FBD and SFC.

4.2 FB

The editor shall provide all current logic blocks as graphical library elements. These can be dragged into the editor window and connected to each other at the inputs and outputs.

4.3 File types

The system shall work with the file types, PLCopen XML and AML. It must transfer these files into a graphical PLC program or be able to export a PLC program in one of these file types.