

## Analysis of dataflows within industrial control system design

Jiri Koziorek<sup>1,\*</sup>, Jaromir Konecny<sup>1</sup>, Antonin Gavlas<sup>1</sup>, Radim Kraut<sup>1</sup>, and Petr Walder<sup>2</sup>

<sup>1</sup>Department of Cybernetics and Biomedical Engineering, VSB-Technical University of Ostrava, Ostrava-Poruba 70833

<sup>2</sup>ELVAC a.s., Hascicka 53, 700 30 Ostrava, Czech Republic

**Abstract.** A design of industrial control system is a complex process which is usually divided into specific design steps. Results from antecedent step are generally inputs for the next step and there could be also other influences among steps. So there is a strong dataflow within an automation project especially if the range of designed system is large from the point of view of the signals count, components, variables, alarms and other messages etc. The ability of effective and reliable data transfer between the steps has significant influence on effectivity and quality of the design of an industrial control system. The main goal of the paper is an analysis of the design steps during the industrial control system design, definition of the dataflows between particular steps and description of a concept of automated system for data transmission within the design. The proposed concept is presented and evaluated on pilot industrial control system design project.

### 1 Introduction

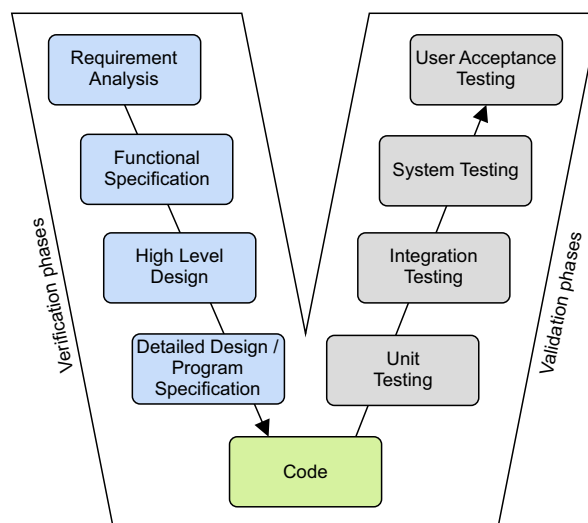
Industrial control system design is a complex process which contains a number of steps. The result of design is strongly influenced by this process, especially properties as quality of control applications, efficiency of design, system reliability, sustainability and further extension of the control system.

A control systems are generally defined as a subset of the general concept of the system, which is defined in the IEEE standard glossary as “a collection of components organized to accomplish a specific function or set of functions.” [1] The main specifics that distinguish them from other systems interaction with environment (represented by other systems, hardware and software entities, people, the physical world, etc.) and the fact that controllers typically have to react with the surrounding environment in real time.

The majority of resources in this area is focused on design of control application. There are many approaches how to proceed with the design of control application. The aim of these approaches is an effective design of a high quality application from the perspective of the structure, readability, extendibility, level of hidden faults, efficiency of commissioning etc. [3], [4]

A typical example of control application design is so called V-model which is used for standard control applications as well as for functional safety control applications. V-model is a graphical representation of a process system design used from facilitating the understanding of complex systems to detailed description of the various stages of the system life cycle. It has a characteristic V-shape in which the left side represents the decomposition and the

specification of the system, while the right side represents a system integration and validation. [2]



**Fig. 1.** Control system design by V-model.

The control application and its quality are a very important aspects but from the point of view of the entire industrial control system design, this is the last step.

The subject of the research described in this paper is based on requirements of the industrial engineering companies that need to increase their efficiency work and increase the quality of the projects.

The aim of the research is to look at the design process on a wider scale and include all design steps which are typically solved by these engineering companies during industrial control system design during “automation project”.

\*e-mail: jiri.koziorek.vsb.cz

## 2 Design steps in automation project

It is necessary to look at the design process in complex view. In following text, authors will describe design process in a machinery application area, but it could be applicable also in other application areas, especially in the process automation and others.

As the first step in system design could be considered a definition of requirements by a customer. This step is in fact outside of the design process but it is a very important clear understanding of customer demands by engineering company. The close cooperation of customer and engineering company in this phase is very advantageous for whole following process. The results of this steps are:

- Definition list of mechanical requirements.
- Functional analysis of whole machine.

The first step in engineering process is mechanical design. The mechanical design solve all mechanical aspects of the machine, especially:

- Construction concept assuring required operation.
- Dimensioning.
- Definition and construction design of all subsystems.
- Design of movable/solid parts.
- Functional safety aspects.
- Etc.

The mechanical design brings important results from the point of view of industrial control system design. The mechanical design specifies information about specific components of a machine that will be subjects of control. Especially, the following components should be mentioned:

- Motors and drives.
- Another machinery components.
- Requirements for sensing and actuating.
- Etc.

This information is the basis of *Motor Control and Instrumentation List* which is discussed in following section.

The next step that follows of that could run partially in parallel with mechanical design is an electrical design. The main goal of this step is:

- Specification of all components which will be a part of an electrical equipment of the machine.
- Scheme of an electrical circuit.
- Definition of technological signals.
- Functional safety aspects.

The electrical design is an essential step of a control system design. It brings very important information that have to be used during the design of a control application.

The combination of specific information from mechanical and electrical design are combined in *Motor Control and Instrumentation list*, which is a basic data source for following design steps. Another important result is a list of technological signals.

Steps mentioned above are preparatory from the point of view of a control application design but they bring an important information that have to be used during the control application design. The way of using this information could significantly influence the efficiency and the quality of control application design.

A typical industrial control system in machinery and also other industrial branches contains real time control system and human-machine interface. A real time control system in standard applications is typically based on programmable controllers in centralized or distributed architecture. Human-machine interface could be typically realized as operator panel or as PC computer with SCADA software.

As the PLCs and HMI operator panels are typical instruments in machinery automation, the following text will be mostly focused on this configuration nevertheless the described approaches are general and could be use also in another configuration and in another application areas. [5]

The design of control application for system based on programmable controllers is a discipline in which could be used a lot of approaches but there is not any approach required for example by international standards. The companies have usually their own internal standardized procedures to design applications. The example of well known general approach is described in Fig. 1, but different approaches could lead to well designed control application. In all cases, important initial conditions must be met to enable effective and quality design of the applications:

- Availability of complete and consistent data from mechanical and electrical design – list of components, subsystems, electrical scheme etc.
- Availability of complete and consistent list of technological signals including ranges, measurement requirements, processing requirements etc.
- Functional analysis of whole machine (system).
- Definition of internal information signals (e.g. statuses, commands, alarms) that are related to used components and subsystems.
- Required connection to human machine interface, signals and variables required for visualisation application.

The following text will deal with collecting of all mentioned information and their transfer within the design process.

## 3 Definition of links between steps

In previous section, the following main steps were defined based on analysis of design procedures of engineering companies:

1. Definition of requirements by a customer.
2. Mechanical design.
3. Electrical design.
4. Design of control application.
5. Design of visualization application.

The steps 1 – 3 are usually solved in different engineering CAD/CAE systems and another tools, and also in different engineering departments in company. These facts usually lead to situation that control engineer have several resources, from different software tools, from different departments. Data could have non-uniform format or format which is not suitable for direct use in control systems programming tools. [6]

Usually the first task for control system programmer is to compile all these resources to obtain initial information for control application design process. This task is essential for the control application design process but as it could bring in a lot of potential errors to control application in case it is done in wrong way. As this procedure contains a lot of manual work and the programmer usually has to handle with hundreds/thousands of items, the risk of faults is quite high. This procedure has also significant influence on efficiency of the control application design process.

#### 4 Analysis of dataflows

To be able to increase the effectivity and quality of the design process, the thorough analysis of design procedures of selected engineering company were done. The aim of the analysis was a definition of all involved parties, data structures and workflows within the design process of automation project.

The main parties involved are following:

- Mechanical engineer.
- Electrical engineer.
- PLC programmer.
- Human machine interface programmer.
- Other specialists involved to simulation, testing, commissioning and other activities.

The mechanical engineer, electrical engineer and PLC programmer have an insight and understanding of controlled system and they could specify data structures and dataflows and they are responsible for data content. [7, 8]

The result of the analysis is described in Fig. 2. The mechanical engineer is at the beginning of the process. The result of his activity is (except of own mechanical project) so called *Motor Control and Instrumentation List (MC&Instr. List)*. The *MC&Instr. List* is filled by the data from mechanical project and it is one of the inputs for electrical project. Electrical engineer uses this result and work on electrical project. The electrical engineer amend *MC&Instr. List* and creates so called *IO List* e.g. list of technological signals. Composition of these results leads to creation of the *Definition List*. The *Definition List* is common data repository which is main structure for exchanging data from initial steps of automation project to steps dealing with PLC and HMI application design. The *Definition List* contains objects and others information that could be imported to the applications.

*Definition List* could be then used to generate a PLC program structure, to generate list of variables, IO signals, data structures as well as it could be finally used also for

generation of program code basis. *Definition List* could be also used for generation variables and structures to HMI application and to simulation. [9]

There are black and blue items in Fig. 2. The black items and flows shows the activities which are done especially by manual work, the blue items and flows shows activities which could be done automatically (despite of fact that they are usually done also manually in companies)

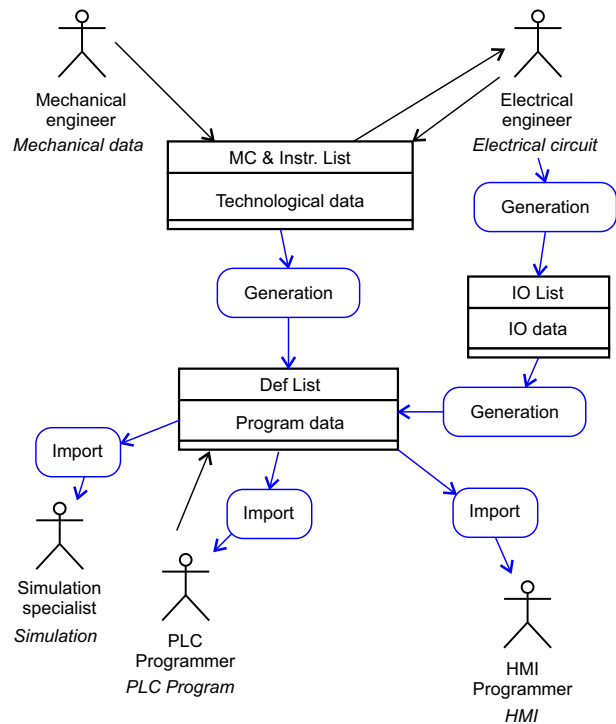


Fig. 2. Description of involved parties and dataflows within the design process of automation project.

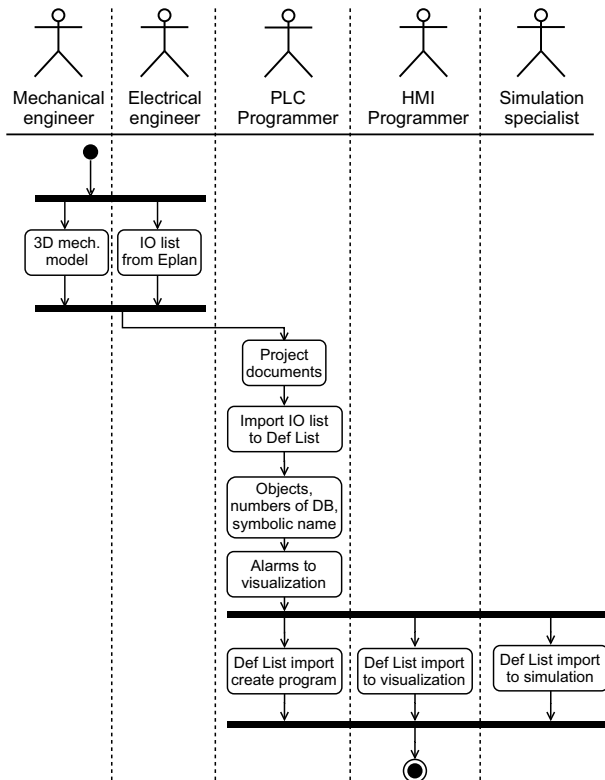
Next result of the analysis is activity diagram in Fig. 3 of parties involved in the automation project. The figure shows by different way the process that was described in the text above. There is better visible the key role of PLC programmer in the process. All the information from previous steps are collected in the beginning of control application design step in the form of *Definition List*. The way of using this information and way of providing them to HMI step and simulation step is crucial in whole automation project.

#### 4.1 Definition list structure

The *Definition List* contains objects. An object is a particular subsystem or component of controlled system, such as drive, hydraulic pump, valve etc. The object could be also a general subsystem related to controlled system, as diagnostic subsystem or processing analog signals.

Each object must have assigned identifier – symbol, data type, data area, description and it could be assigned also an algorithm/program code.

The control application must be able “to handle” with each subsystem represented by the particular object. It is necessary to know an object status and to influence the



**Fig. 3.** Activity diagram of automation project.

object. Usually it is also necessary to provide information about faults of the object which could be done by alarm messages.

So the object in *Definition List* is represented by:

- Symbolic name of the object (identification).
- Object description.
- List of statuses.
- List of commands and setpoints.
- List of faults/alarms.

The statuses, commands, setpoints are usually represented by variables of any elementary data types, while alarms are usually represented by a message text.

#### 4.2 Linkage with control algorithms

The approach described in this paper consider the *Definition List* as a basic source of information to create structure, variables, signals, data structures in control application. It reduce significantly the volume of manual work and reduce possible faults.

There is other possible use of *Definition List* that can further increase efficiency of control application design. If the object represents a typical subsystem which is used in different type of controlled systems, machines, the engineering companies could have already prepared control algorithms for such subsystems. The control algorithm could be represented for example by a function block created by using the programming languages according IEC 61131-3. [10, 11] Then the *Definition List* is also used to

generate the basic program code of a control application. The PLC programmer then add a specific algorithm and other aspects of the application which are necessary for particular machine. [12]

#### 4.3 Possibilities of automated programming

The systematic work with data which are generated during the whole process of automation project can lead to a real internal standardization of control system design. It increases efficiency and it could lead to a high level of automated programming. If the company is focused on a particular application area there is a high probability that a lot of subsystem are used repeatedly. Then the company could have a libraries of “macros” which contains:

- Typical mechanical construction of a subsystem as part of CAD project.
- Typical electrical configuration of a subsystem as part of CAE project.
- Statuses, commands, setpoints, alarms of subsystem.
- Function block to control a subsystem.
- Visual component to display a subsystem on HMI.
- Block to simulate a subsystem in a simulation software.

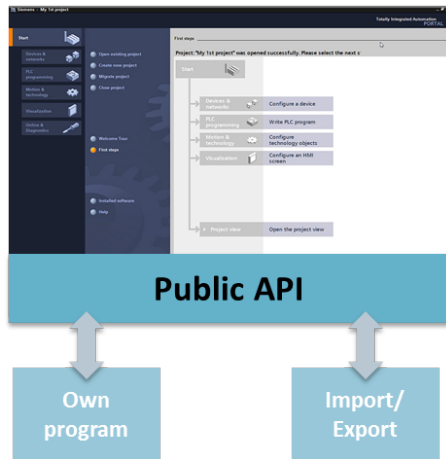
In such case, there is a real chance to solve number of steps within the automation project very effectively and automatically. The PLC programmer will do an adaptation and tuning of the application to particular controlled system condition.

The control application programming tools of main PLC producers have provided in last few years possibilities of automated programming or possibilities of external access to control application that can be used for automated programming. [13, 14]

As most of the steps regarding dataflows within automation project are general, vendor independent (*IO List, MC&Instr. List, Definition List*), the final goal of the process is automated generation PLC program or HMI application which are vendor specific.

We used TIA Portal V14(15) which is used for programming PLCs Siemens Simatic S7 to test proposed approach. TIA portal provide an interface TIA Openness built on .NET Framework. TIA Openness can be used for accessing the TIA Portal project by external applications. This fact can be used to automate the control application development. [15] [16]

In Fig. 4 is described the general concept of TIA Openness. A external application can access TIA Portal project. The data could be imported to or exported from TIA Portal project in \*.xml format. The selected component of TIA Portal application can be exported, modified in external application and imported back. Generally, TIA Openness enables opening/closing TIA Portal PLC project, open and modify hardware configuration, compare offline and online project, import and export of program blocks, data tables, data types technological objects etc. Similar functions could be used also for TIA Portal HMI project. [17]



**Fig. 4.** Elementary concept of access to process data.

Therefore, TIA Openness is a suitable tool for implementing described approach in which all suggested functionalities could be tested.

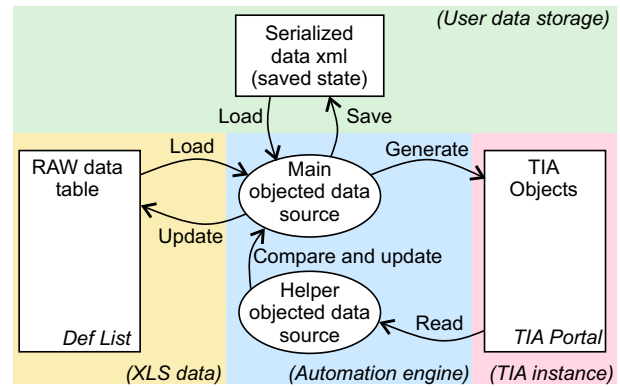
### 5 Concept of automated system for data transmission

Based on specified concept and on the possibilities of TIA Openness, the software tool for generation of control application data and program was developed. The structure of the developed tool is described in Fig. 5. The basic data source for the tool are data represented as objects. The source of the data, as mentioned above, is *Definition List* in \*.xls format. Raw data are read by *Load*. The data could be then saved to *User data storage* by *Save* in XML format and read back by *Load*. These functions enable saving and re-reading the current state of application. The data in XML format could be then used to generate *TIA Objects* and create by this way the basic structure of program in TIA Portal.

It is also possible to read current application from TIA Portal. To do it in a safe way, there is *Helper objected data source* to which data from TIA Portal are read and then compared with *Main objected data source*. The user can accept or do not accept detected changes. The data from *Main objected data source* could be exported back to *Definition list*.

### 6 Pilot design project

An single purpose machine was chosen as a pilot project in which engineering company had to do all design steps, from mechanical and electrical design to control and HMI application. The machine contains different types of subsystems – pneumatic, hydraulic components, conveyor belt, drives. The control system is composed by programmable controller and operator panel. The project deals also with functional safety aspects and distributed architecture design which are not subjects the testing of automated design approach. The typical projects of single purpose, production and testing machines usually have



**Fig. 5.** Dataflow in developed tool for generation of control application.

lower count of IO signals (from tenths to hundreds). So from this point of view are not too demanding. On the other hand, they require precise construction concept, short cycle and response times of programmable controller, strictly deterministic behavior etc.

The following aspects were considered within pilot project design:

- Obtaining data from *MC&Instr. List* and *IO List*.
- Creation of *Definition List*.
- Definition of statuses, commands, setpoints, alarms of subsystems.
- Composition of basic structure of PLC TIA Portal project.
- Composition of basic structure of HMI TIA Portal project.

The aim of this section is to compare manual approach without possibilities of automatic generation of application and suggested approach with automatically generated application.

#### Manual approach

The *MC&Instr. List* and the *IO List* prepared by mechanical engineer and electrical engineer are data sources for *Definition List* creation. This process is done manually by PLC programmer. This process means to copy different parts of *MC&Instr. List* and the *IO Lists* and manual composition of object structures. This manual approach could cause errors and wrong items. Of course, the programmer could use *MC&Instr. List* and the *IO List* and start to create control application directly but this approach is even more sensitive for mistakes and nonconceptual design.

Based on *Definition List*, the programmer creates manually the program structure, lists of technological signals, variables, data structures, statuses, commands, alarms lists etc. A big issue in the case of manually created *Definition List* is that when the programmer makes changes in control application it is complicated to keep *Definition List* in actual status. Within the project, there is usually a necessity to do upgrades of the application and *Definition List* grad-

**Table 1.** List of activities performed by control application programmer

Procedure definition	Manual approach	Consistency check	Automated approach	Consistency check	Repeat counts
Definition List creation and verification	manual	NON	automated with manual revision	YES	1 – 4
Creation of data structures for FBs in application (statuses, commands, setpoints, alarms)	manual	NON	automated	YES	15 – 30
Generation of IO list in application	manual (possible xls import)	NON	automated	YES	1 – 3
Upgrade Definition List according application changes	manual	NON	automated	YES	15 – 30
Creating basic structure of control application	manual	NON	automated	YES	1 – 3
Code generation	manual	NON	possible but not implemented yet (inserting of prepared FBs to application)	NON	1 – 3
Generation of HMI tags	manual (possible xls import)	NON	automated	YES	15 – 30
Generation of HMI alarms	manual (possible xls import)	NON	automated	YES	15 – 30

ually. This process is not only laborious but again could be a source of errors.

*Automated approach*

The *MC&Instr. List* and the *IO List* is again prepared by mechanical engineer and electrical engineer. The data are compiled by software tool and the *Definition List* is automatically created. During the automatic composition of *Definition List* the format, consistency and duplication of items are verified. The programmer then do a check of *Definition List*. The structure of objects and other aspects of the items could be modified. When the *Definition List* is completely created and verified, the automatic generation of the control application could be done. The the program structure, all the lists of technological signals, variables, data structures, data types, statuses, commands, alarms lists etc. are generated automatically. The programmer then during the process of implementation of application can modify the lists of items and all the changes will take effect also in *Definition List* and vice versa. So *Definition List* is real, basic, continuously upgraded data source for control application with minimal risk of errors and inconsistency.

Also connections between PLC application and HMI application are generated as well as alarm messages.

The table 1 contains list of activities performed by control application programmer. The table presents a comparison between manual and automated approach. The table could serve as introductory information for considering about increasing the efficiency of design and about application of approaches useful for internal standardization in engineering companies.

**7 Conclusion**

The main goal of the paper was analysis of dataflows in process automation project design. The steps of the design process and data generated in each step were specified with cooperation with engineering company focused on machinery automation. According the analysis, the software tool to automate the design process was developed.

The tool is now in initial version and it makes possible to automate basic activities in automation project. The tool is ready for further development to reach high level of automated code generation of control application.

Even though the actual version of application does not contain all possible functionalities, it could be already used well in automation project and brings a lot of benefits. The main benefits are:

- Increasing of work productivity by reduction of time consuming manual operation.
- Reduction of risk of errors in control application.
- Keeping the data source up-to-date for application in the form of Definition List.
- Increasing of level of standardized approach in engineering company.
- Increasing of readability and uniformity of applications that makes easier cooperation of several programmers on application development and commissioning.

Research activities described in this paper are still in process and system will be tested in various real projects. The final goal will be automatically generated code of control application and automatically generated basis of HMI application.

This work was supported by the European Regional Development Fund in the Modular System of Dynamic Drives Control, project number CZ.01.1.02/0.0/0.0/15\_019/0004315 within the Operational Programme Research, Enterprise and Innovations for Competitiveness.

This work was also supported by the European Regional Development Fund in the Research Centre of Advanced Mechatronic Systems project, project number CZ.02.1.01/0.0/0.0/16\_019/0000867 within the Operational Programme Research, Development and Education.

**References**

[1] IEEE Std 610.12-1990 pp. 1–84 (1990)

- [2] R.S. Pressman., *Software engineering*, 7th edn. (McGraw-Hill Education – Europe, London, United States, 2010), ISBN 9780071267823
- [3] S. Ozana, M. Pies, R. Hajovsky, J. Koziorek, O. Horacek, *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* **8838**, 501 (2014)
- [4] J. Kocian, J. Koziorek, *An Outline of Advanced Process Control and Self Tuning Techniques on PLC Background*, in *2010 IEEE CONFERENCE ON EMERGING TECHNOLOGIES AND FACTORY AUTOMATION (ETFA)* (IEEE, 2010), IEEE International Conference on Emerging Technologies and Factory Automation-ETFA, ISBN 978-1-4244-6850-8, ISSN 1946-0740
- [5] F. Basile, P. Chiacchio, D. Gerbasio, *Progress in PLC programming for distributed automation systems control*, in *2011 9th IEEE International Conference on Industrial Informatics* (2011), pp. 621 – 627, ISSN 1935-4576
- [6] S. Ulewicz, B. Vogel-Heuser, M. Ulbrich, A. Weigl, B. Beckert, *Proving equivalence between control software variants for Programmable Logic Controllers*, in *2015 IEEE 20th Conference on Emerging Technologies Factory Automation (ETFA)* (2015), pp. 1 – 5, ISSN 1946-0740
- [7] M. Obermeier, D. Schütz, B. Vogel-Heuser, *Evaluation of a newly developed model-driven PLC programming approach for machine and plant automation*, in *2012 IEEE International Conference on Systems, Man, and Cybernetics (SMC)* (2012), pp. 1552 – 1557, ISSN 1062-922X
- [8] H. Thaller, R. Ramler, J. Pichler, A. Egyed, *Exploring code clones in programmable logic controller software*, in *2017 22nd IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)* (2017), pp. 1 – 8
- [9] M. Obermeier, S. Braun, B. Vogel-Heuser, *IEEE Transactions on Industrial Informatics* **11**, 790 (2015)
- [10] M. Jamro, B. Trybus, *IEC 61131-3 programmable human machine interfaces for control devices*, in *2013 6th International Conference on Human System Interactions (HSI)* (2013), pp. 48 – 55, ISSN 2158-2246
- [11] R. Ramanathan, *The IEC 61131-3 programming languages features for industrial control systems*, in *2014 World Automation Congress (WAC)* (2014), pp. 598 – 603, ISSN 2154-4824
- [12] H. Prähofer, F. Angerer, R. Ramler, F. Grillenberger, *IEEE Transactions on Industrial Informatics* **13**, 37 (2017)
- [13] A. Sapena-Bano, R. Puche-Panadero, M. Perez-Vazquez, J. Perez-Cruz, J. Martinez-Roman, M. Pineda-Sanchez, V. Perez-Vázquez, M. Perez-Vazquez, *Automatic translation of Programmable Logic Controllers (PLC) control programs in packaging machinery*, in *2014 9th International Microsystems, Packaging, Assembly and Circuits Technology Conference (IMPACT)* (2014), pp. 445 – 448, ISSN 2150-5934
- [14] P. Seman, M. Juhás, L. Tkáč, M. Honek, *New possibilities of industrial programming software*, in *2013 International Conference on Process Control (PC)* (2013), pp. 474 – 479
- [15] J. Hufnagel, B. Vogel-Heuser, *Data integration in manufacturing industry: Model-based integration of data distributed from ERP to PLC*, in *2015 IEEE 13th International Conference on Industrial Informatics (INDIN)* (2015), pp. 275 – 281, ISSN 1935-4576
- [16] G. Danilevičius, D. Ezerskis, L.B. Kaunas, *Data structures and interfaces for automated PLC applications development process*, in *Proceedings of the 6th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems* (2011), Vol. 2, pp. 546 – 549
- [17] *Tia portal openness: Introduction and demo application*, [https://cache.industry.siemens.com/dl/files/692/108716692/att\\_920046/v1/108716692\\_TIA\\_Openness\\_GettingStartedAndDemo\\_V14SP1\\_en.pdf](https://cache.industry.siemens.com/dl/files/692/108716692/att_920046/v1/108716692_TIA_Openness_GettingStartedAndDemo_V14SP1_en.pdf)