Automated control system design with model-based commissioning

Jiri Koziorek, Antonin Gavlas, Jaromir Konecny, Martin Mikolajek, Radim Kraut, Petr Walder

Abstract – A process of industrial control system design contains a set of steps. The result of the design process is significantly influenced by quality of execution of each step. An important phase of the design is a testing on different level of the design. The testing process influences a commissioning of the control system which follows the design process. The article describes innovative approach of automated control system design and analyses the possibilities of model based testing and commissioning. Both, the automated control system design and model based commissioning increase efficiency and quality of engineering process and of the designed control system. The model based commission is very close to digital twin approach which is one of important trends in automation.

Keywords – Control system design, Model based design, programmable controllers, industrial automation.

I. 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.

The 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]

This work was 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.

This work was also 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.

J. Koziorek is with Department of Cybernetics and Biomedical Engineering, VSB-Technical University of Ostrava, Czech Republic (corresponding author to provide phone: +420 597 325 950; e-mail: jiri.koziorek@vsb.cz).

J. Konecny is with Department of Cybernetics and Biomedical Engineering, VSB-Technical University of Ostrava, Czech Republic.

A. Gavlas is with Department of Cybernetics and Biomedical Engineering, VSB-Technical University of Ostrava, Czech Republic.

R. Kraut is with Department of Cybernetics and Biomedical Engineering, VSB-Technical University of Ostrava, Czech Republic.

P. Walder is with ELVAC a.s., Ostrava, Czech Republic.

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]

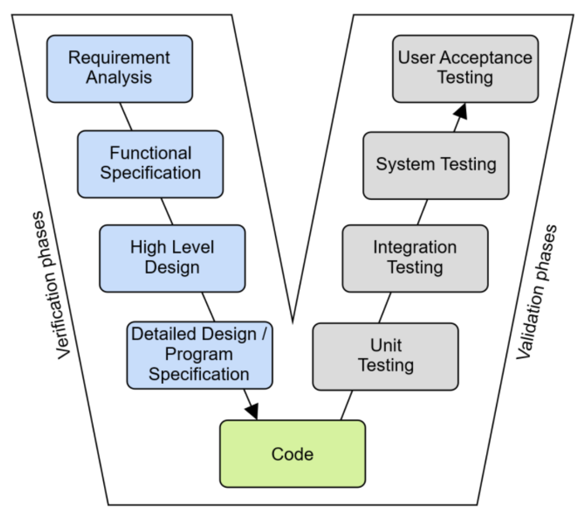

Fig. 1 Control system design by V-model.

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. To achieve proper functionality and quality of control applications, several levels of testing are necessary, as described in figure 1 – unit tests, integration tests, whole system tests and user acceptance tests. In case of control applications, the processing of the tests requires to connect control system to a controlled system, or to a model of controlled system. The control system could be tested if real or simulated technological signals are connected to the control system.

The subject of the research described in this article 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”. The other aim of the work is a demonstration how designed control system could be tested and commissioned using modelling and simulation.

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 solves 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.

II. 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.

III. 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 was 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 amends *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 show 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).

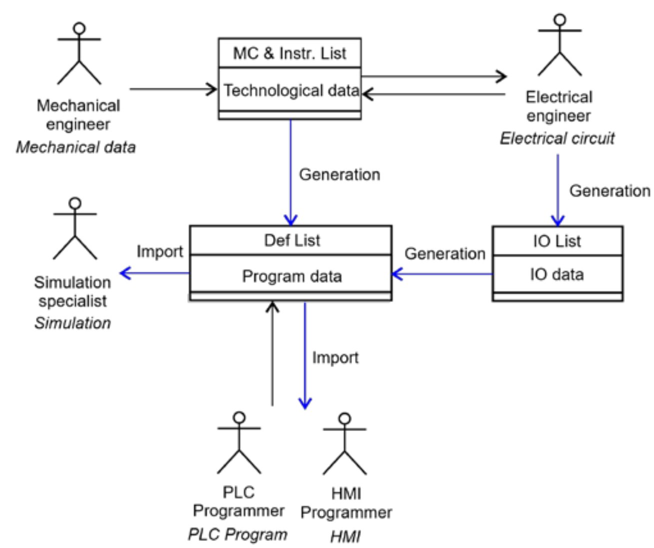

Fig. 2 Description of involved parties and dataflows within the design process of automation project.

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.

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

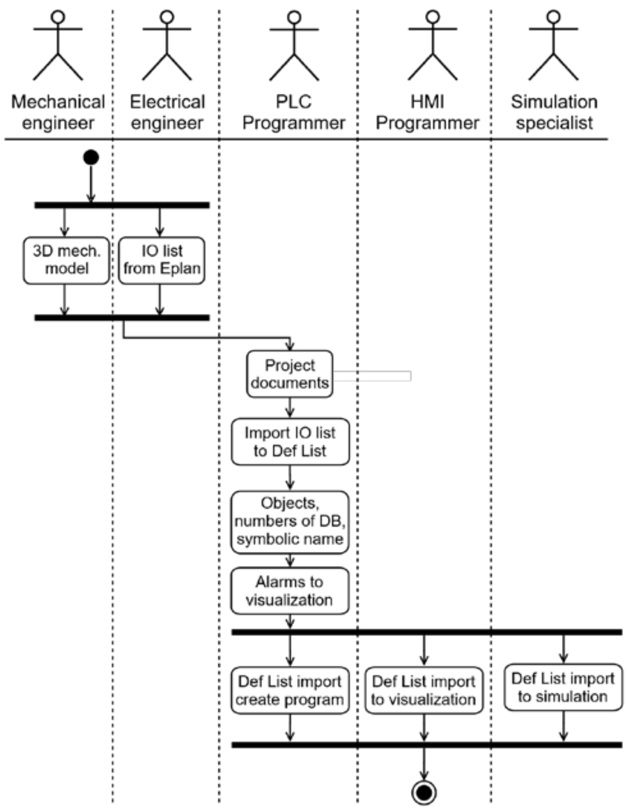

Fig. 3 Activity diagram of automation project.

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.

B. Linkage with control algorithms

The approach described in this article consider the *Definition List* as a basic source of information to create structure, variables, signals, data structures in control application. It reduces 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 sybsystem which is used in different type of controlled systems, machines, the engineering companies could have already prepared control algorithms for such sybsystems. 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]

C. 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 sybsystem.
- Function block to control a sybsystem.
- Visual component to display a subsytem 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.

One of possible tool is 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] An 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]

Therefore, TIA Openness is a suitable tool for implementing described approach in which all suggested functionalities could be tested. 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. 4. 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*.

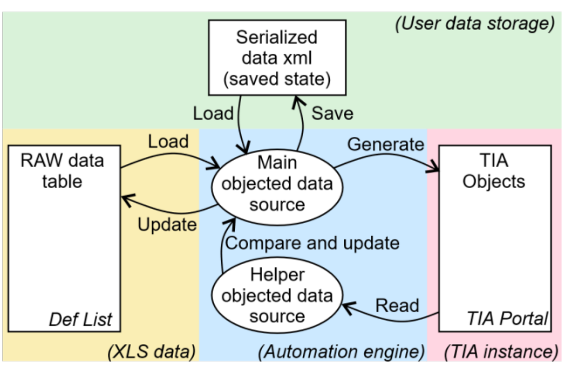

Fig. 4 Dataflow in developed tool for generation of control application.

IV. MODEL-BASED DESIGN

Testing and commissioning based on modelling and simulation could reduce the time of design process and could help in verification phase especially when testing on real controlled system is not possible. The approach in which are models used in systems design is generally called model-based design (MBD). [26] The MBD are usually considered set of methods describing specific problems associated with design of complex control including signal processing and communication systems.

The model helps people to know, understand, or simulate a subject the model represents. The model is a representation of controlled system providing technological signals and behavior of controlled system which is then used for testing of control system functionality. Generally, the model could be physical (represented by real system) or mathematical (represented by mathematical description). The mathematical model consists of an abstract system of mathematical relations that describe the properties of examined device. An implementation of mathematical models and experimenting with them is the subject of the simulation. [18]-[19]-[20]

The models are used instead of real controlled systems in the phase of testing of control application and can help especially in situations when real systems are not available or when testing activities could cause a damage on real systems. The process of using a model for testing can be divided into following stages [21]:

- Analysis and design of a model of a system.
- Verification and tuning the model.
- Analysis and implementation a control algorithms/application.
- Verification and tuning controller.
- Testing the control application by simulation on the model.

The standard approach of software design which is not based on modelling doesn't contains a creation, a verification and a tuning of the model. These steps are additional and consume an engineering time. So engineering company need to consider whether potential benefits outweigh the disadvantages. [22]

A. Strategies of model based design

The approaches are called as model-based design could be divided to several categories according a way of realization and using model in a design process. [26] Basic categories are following:

- Model in the loop (MIL) – the control system as well as controlled systems are simulated. So the design and testing process are performed without any physical hardware components - virtual representation of both parts.
- Software in the loop (SIL) – developed control blocks are replaced in the simulation by software code representation which means the inclusion of compiled production software code into a simulation model. There is possible to use automatic code generation functions from specific control blocks.
- Processor in the loop (PIL) – the control algorithm is operated on target control device. The model of the system is built by a specific tool, usually based on a software modeling tool on PC computer. The simulation is then running on this computer and the target control device is connected by communication network.

- Hardware in the loop (HIL) – similar like PIL but target control device and the simulation device are interconnected by real technological IO signals.
- Rapid Control Prototyping (RCP) – this approach enables to design, test, verify and tune control strategies on a target hardware or a RT simulator connected to real (not simulated) prototype of controlled system.

The SIL, PIL simulation strategies are widely used in engineering practice of companies focused on industrial automation. The HIL simulations are usually advantageous in more demanding applications. [23] [24]

B. *The digital twin*

The digital twin concept extends an approach of modelling the system. The digital twin concept is much more complex. It can be used to understand, predict, and optimize performance of the system in order to achieve improved business outcomes. As mentioned above, the digital twin is created during system design and remains connected to a system during whole its lifecycle. [25]

The digital twin is usually described as a complex of three components: a data model, a set of analytics or algorithms, and a knowledge. [26]

The digital twin is more complex but gives more features. It brings complete virtual design of product or system and is used for number of other purposes then only testing and commissioning.

V. Model Based Testing and Commissioning Application Example

The application example is based on design of a laboratory production that performs an assembling of defined products built from Lego bricks. The production line contains different types of subsystems – conveyor belt, drives, robotic arms, pneumatic components etc. The control system is based by a programmable controller and operator panel. Robotic subsystems have their own controllers (in Figure 5).

As the production line were designed virtually in selected CAD system the data from this mechanical design were then used for automated control system design as well as for testing of control application. [26]

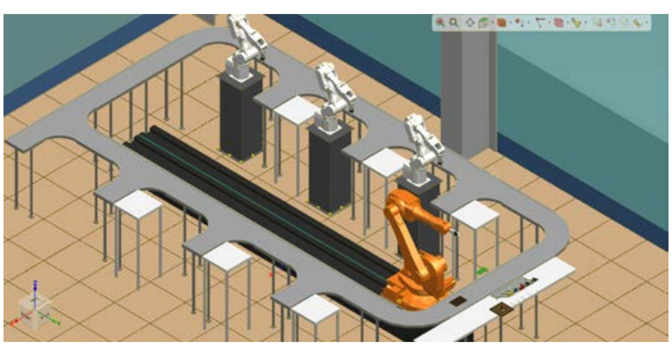

Fig. 5 CAD design of the production line.

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 sybsystems.
- Composition of basic structure of PLC TIA Portal project.
- Composition of basic structure of HMI TIA Portal project.

A. *Automated control system design*

The *MC&Instr. List* and the *IO List* are based on mechanical and electrical design. The data are compiled by software tool and the *Definition List* is automatically created. When the *Definition List* is completely created and verified, the automatic generation of the control application could be done. 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. The additional control algorithms which are not generated automatically are completed by programmer. The automated generation of the basis of automation project reduces routine tasks which could causes errors in control application.

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

B. *Testing and commissioning*

The testing and commissioning of control application is based on PIL model of controlled system. The inclusion of the model to a structure of control system is described in Figure 6.

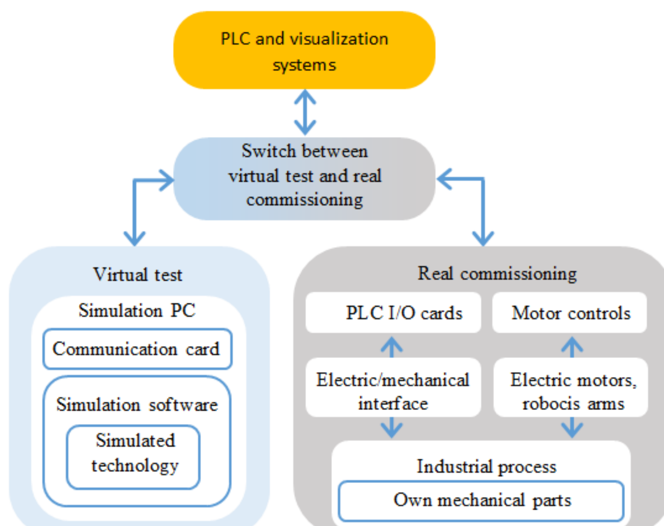

Fig. 6 Testing and commissioning using by PIL/HIL model.

Advantages of this approach:

- In this configuration, the real PLC could be used while remote I/O modules and signals connected to these modules are simulated. So a configuration of the control system hardware is the same as during the real commissioning.
- If the used software does not have a description of the peripherals used, it is possible add to this simulation mode additional information to describe the real remote I/O modules or motors drives, etc. This additional information could be in GSD files or similar.
- Comfortable testing of process software during construction of the final process line.
- The opportunity to train service technicians during virtual testing.
- Switching from simulated mode to real mode is made simply by replacing simulated periphery by real periphery.
- When the model is based on CAD design with exact dimensioning, the testing and commissioning is made in conditions which are very close to real system. [26]

CONCLUSION

The main goal of the article is description of an approach of automated design of control system followed by model based testing and commissioning. The steps of the design process were presented, the data generated in each step were analyzed and possibilities of automation of design process were summarized. All these parts were specified with cooperation with engineering company focused on machinery automation. The result of the design process was then tested by a simulation based on PIL model. The basic steps needed for simulation were described in application example. The described approach is now implemented to engineering procedures of mentioned company and tested in real applications.

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 978-1-4244-68508, 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 21582246

[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. PerezVazquez, J. Perez-Cruz, J. Martinez-Roman, M. Pineda-Sanchez, V. Perez-Vázquez, M. PerezVazquez, *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. Tkác, 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 19354576

[16] G. Danilevicius, 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

[18] B. Shen, T. Wang, J. Zhao, Z. Tang, *Application of virtual reality to simulation in equipment maintenance*, in *Proceedings of 2012 International Conference on Measurement* (2012), Information and Control, Harbin, China, 18-20 May, pp 524 – 528, doi: 10.1109/MIC.2012.6273356.

[19] O. Krejcar, I. Spicka, R. Frischer, *Implementation of full-featured PID regulator in microcontrollers*, in *Elektronika ir Elektrotechnika* (2011), (7), pp 77-82. ISSN: 13921215.

[20] P. Michalik, J. Stofa, I. Zolotova, *The use of BPMN for modelling the MES level in information and control systems* (2013), in *Quality Innovation Prosperity*, 17 (1), pp. 39-47. DOI: 10.1234/QIP.V17I1.68

[21] Z. Papp, M. Dorrepaal, D. J. Verburg, *Distributed hardware-in-the-loop simulator for autonomous continuous dynamical systems with spatially constrained interactions* (2013), in *Proceedings of the IEEE International Parallel and Distributed Processing Symposium*, Nice, France, 22-26 April, doi: 10.1109/IPDPS.2003.1213235.

[22] J. Koziorek, S. Ozana, V. Srovnal, T. Docekal, *Modeling and Simulations in Control Software Design*, in *Analytic methods in systems and software testing*, Hoboken, NJ, USA: Wiley, pp. 287-326. ISSN: 9781119271505

[23] S. Ozana, M. Pies, R. Hajovsky, J. Koziorek, O. Horacek, *Application of PIL approach for automated transportation center* (2014), *Lecture Notes in Computer Science*, 8838, pp 501-513. ISBN 978-366245236-3.

[24] K. Rastocny, J. Zdansky, M. Franekova, I. Zolotova, *Modelling of diagnostics influence on control system safety* (2018), in *Computing and Informatics*, 37 (2), pp. 457-475. DOI: 10.4149/cai_2018_2_457.

[25] Q. Qi, F. Tao, Y. Zuo, D. Zhao, *Digital Twin Service towards Smart Manufacturing* (2018), in *ScienceDirect*, 72, pp 237-242.

[26] J. Koziorek, R. Kraut, A. Gavlas, M. Mikolajek, P. Papcun, I. Zolotova, *Effective control system design with commissioning based on digital twin* (2018), in 48th *International conference of computers and industrial engineering (CIE48)*. 2-5 December 2018, Auckland, New Zealand