Chalmers Publication Library

Automatic generation: A way of ensuring PLC and HMI standards

(article starts on next page)

Automatic Generation: A way of ensuring PLC and HMI standards

Petter Falkman* (petter.falkman@chalmers.se), Erik Helander†, Mikael Andersson‡
*Chalmers University of Technology, Department of Signals and Systems, Gothenburg, Sweden
†Control Systems, Body FTE Gothenburg, Volvo Car Corporation, Göteborg, Sweden
‡Robot Simulation BiW & FA, Volvo Car Corporation, Göteborg, Sweden

Abstract

Preparing an automatic production system takes a lot of time and to be able to decrease this time virtual simulation studies are used more and more frequently. However, even if more work is performed in a virtual environment a problem is still that the same work is done more than one time in different software tools due to the lack of integration between them.

The present paper presents a case study that investigates how a newly developed tool called SIMATIC Automation Designer can be used in order to close the gap between the mechanical design and the electrical design. SIMATIC Automation Designer is a Siemens software that can generate PLC code and HMI screens. The result shows that by generating PLC code and HMI screens automatically, it is possible to get the same structure and naming standard in every PLC and HMI project. This will ensure a corporate standard and will be a quality assurance of the PLC code and HMI screens.

1. Introduction

Preparing an automatic production system takes a lot of time and to be able to decrease this time virtual simulation studies are used more and more frequently. However, even if more work is performed in a virtual environment a problem is still that the same work is done more than one time in different software tools due to the lack of integration between the them. For a virtual simulation study the right equipment must be chosen and named to do a proper simulation of the real system in order to be able to make a reliable investigation of the systems behavior.

The integration between the traditionally mechanical engineering and electrical engineering activities is not yet achieved. This means that information about mechanical descriptions such as system layout and behavior can normally not be transferred and used in a satisfactory way by the electrical engineer and programmers. A manual transfer of the information is therefore required that leads to both double work and possible human induced errors. One way to decrease human errors and ensure the structure and standards for both the control programs and HMI screens is to use automatic generation.

PLCs (Programmable Logic Controllers) are computers specially designed for industrial control applications. A PLC is programmed in special PLC languages that follows the standards set in IEC 61131-3 [4]. To enable an operator to supervise and manipulate the system controlled by the PLC a HMI (Human Machine Interface) panel is commonly used.

Automatic generation of PLC code has been done several times before as for example in [5] who generated ladder code or PLC code generation for industrial robot cells [1]. The focus of many of these reports has been to generate code to handle interlocks in a general way. The focus of the present paper is to investigate the possibilities of generating PLC programs and HMI screens according to a company standard. A case study has been performed at Volvo Car Corporation, Gothenburg, Sweden, and will be presented in Section 2.

The present case study is investigating how newly developed tools can be used in order to close the gap between the mechanical design of a production system and the electrical design of the automation solution. The software tools used in this case study are Process Simulate and SIMATIC Automation Designer. Process Simulate is a software for 3D kinematic simulation and assembly path planning for production facilities [7]. Automation Designer includes tools for automatic generation of PLC code and HMI screens and are able to do this by integrating data about mechanics, electrics and automation of devices used in the factory [8]. The main focus of this paper is to show how PLC code and HMI screens can be generated automatically from library function blocks and simulation studies and through this ensure the that the PLC programs follow the Volvo Car Corporation standards. An additional aim is to show how much information that can be transferred from a simulation study in Process Simulate into the software Automation Designer.

2. Case study

Three stations which are part of the production line creating the underbody of a car are used for the case study. Figure 1 shows the three stations that are completely automated and controlled by one PLC.
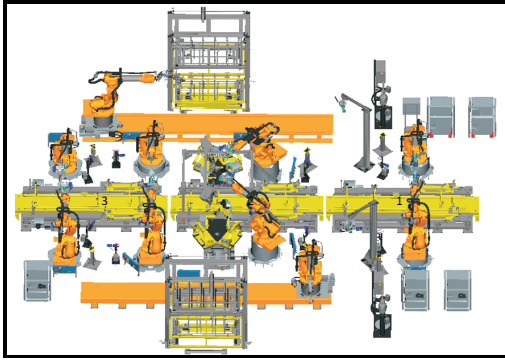
Figure 1. The three stations included in this case study.

To be able to generate code in a desirable way clear standards must be available. The PLC standards differ a lot between companies when it comes to program structure and programming language. Different PLC languages such as ladder, function block diagrams, sequential function charts and instruction list are widely used by the production companies for programming their PLCs [2]. Another standard which differs between the companies is also how to structure the PLC code. The code is usually structured in a flat or a hierarchical structure. A flat structure is often easier to program but for larger projects it tends to get very complex and hard to overview, therefore it is also harder to maintain. The hierarchical structure works in an object oriented way and requires more planning but it allows for a wider reuse of PLC code. All PLC code for this case study is generated in a function block diagram structure and the HMI screens are generated to an XML file which can be imported to the HMI software.
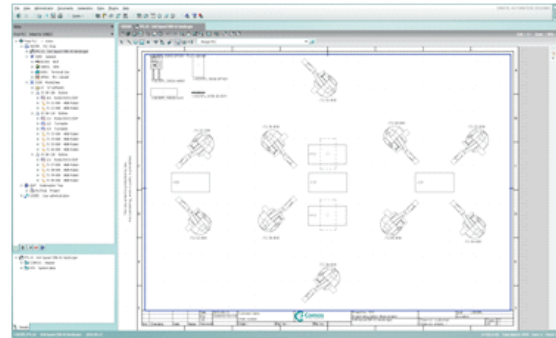
2.1 Automation Designer

SIMATIC Automation Designer is a software tool from Siemens that is developed on the base of Comos [8]. The software is an engineering platform, mainly for the manufacturing industry, to decrease the time for engineering and start commissioning earlier. Automation Designers main purpose is to help the engineers to take the step from the digital factory to the real plant and to reuse as much information as possible [3].

In Automation Designer there is the possibility to generate PLC code for Siemens PLC programming tool Simatic Step 7 [9] and HMI screens for the HMI creation software WinCC flexible [10] by the use of templates created and stored in a database. A template in Automation Designer represents a real world object e.g a robot and contains information about the object such as hardware information, PLC code and HMI screens [6].

When constructing a new line or station an engineering project is typically created in Automation Designer. The engineering project is built up in a tree structure, see Figure 2, by combining templates to create a representation

of the real line. The information in the templates, such as variable names, is modified by scripts to fit the current project. By the use of the tree structure and the possibility to link objects together the structure for the generated PLC program is created. The code used for the generation of the PLC can either be written inside Automation Designer in the STL (Statement List) format or be imported from a Step 7 library [6].

Figure 2. Tree structure and 2D layout in Automation Designer.

Automation Designer can generate HMI screens for an object in four different levels, three levels where the screen is a combination of objects and one detail level containing only information about the specific object. In Automation Designer objects are placed on a 2D drawing, see Figure 2, where polygons representing the different levels are drawn around the objects to assign which objects that should be in the same screens [8]. The design for the object in the different levels is made in WinCC flexible and transferred to Automation Designer using an XML file.

2.2 Case study description

PLC code and HMI screens are automatically generated for the case study stations to test how the Volvo Car standards can be ensured in each generated project. The generations is made by Automation Designer, however to be able to generate the desired information, templates have been created inside the software for the equipment in the stations. All essential PLC code and HMI icons have been included for each created template, but to get the right naming of the objects and variables some small scripts have been added to the templates as well. The scripts are executed when a template is used in an engineering project and are needed to be able to follow the standards. The standards include rules for; naming, structure of PLC code, structure inside the generated function blocks diagrams, numbering of function and data blocks, alarm handling, required HMI screens and number of levels, HMI tags. The present case study also takes into consideration the possibility to reuse the information such as function block libraries and previous created simulation studies.

2.3 Procedure in Automation Designer

The work performed in Automation Designer to get the results for this case is divided into four different steps; create templates, build tree structure based on the templates, import simulation study, and generate PLC code and HMI screens.

In the first step three different types of templates were created for the equipment included in the case study stations. The three types of templates were device templates, S7 programs templates and WinCC flexible templates. The reason for creating three templates instead of one for the equipment is due to naming and also to get the possibility of only generating PLC code. This is harder to do if all information is within one template. However the device template, which is the main template for equipment, is connected to the other two template types and will import the information from theses in the engineering project.

The second step was to build up a tree structure with the templates created in the first step. This was done in an engineering project in Automation Designer. After creating a tree structure a configuration of the project was performed where connections between equipment was set amongst other things.

An XML file containing the simulation study of the three stations is imported into an engineering project in Automation Designer in the third step.

The last step was to open the generators in Automation Designer and generate code to STEP 7 and WinCC flexible screens to an XML file.

2.4 PLC code generation

Automation Designer is able to generate and export a PLC project to STEP 7. Required Function blocks were transferred into the generated PLC project. The internal code of the rest of the function blocks, which were not collected from the library, was generated with a structure according to the Volvo Car standard. The main rule for the standard is that the structure should be according to the order of execution items inside the function block. A calling structure between the function blocks has also been achieved in the generated PLC project.

The coordinating sequence is still done manually. However, which signals to manipulate for each equipment have been put inside the functions blocks and set to a default value, see Figure 3, to support this task.

A standard when using Siemens PLC software is that every function block should have a data area called data block (DB) attached to it. The data blocks include all variables which are used by its function block. These have been generated into the PLC project and are set to a name and DB number. All variables inside the data blocks have also been set to a default value according to its data type.

An important topic is the handling of alarms between the function blocks. In the end of every block there should be a summation of the alarm signals and this should then be sent up through the calling structure. This summa-
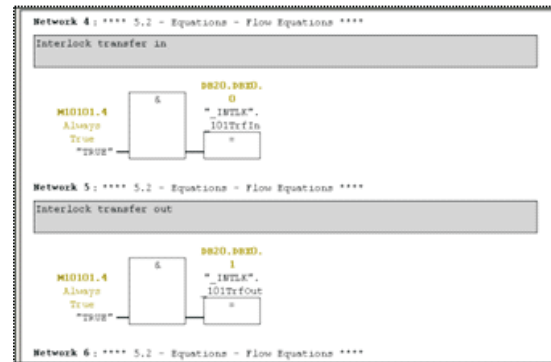

Figure 3. Interlock signals inside one of the generated function blocks.

tion was generated by Automation Designer and alarms are sent to the right level of the PLC structure where it is decided what to do with the alarm.

2.5 HMI screen generation

The HMI screen generation from Automation Designer includes the desired screens and is based on the design set in the 2D layout inside the software (see Figure 1). A tag list containing all tags used in the screens is also included in the generated project. The screen and tag names are created by Automation Designer and are based on the search paths for each object in Automation Designer The generated screens and tags are not exported to WinCC flexible directly as for the PLC project. It is instead generated as an XML file which can be imported into the HMI software. Figure 4 shows the top level screen for the present case study.
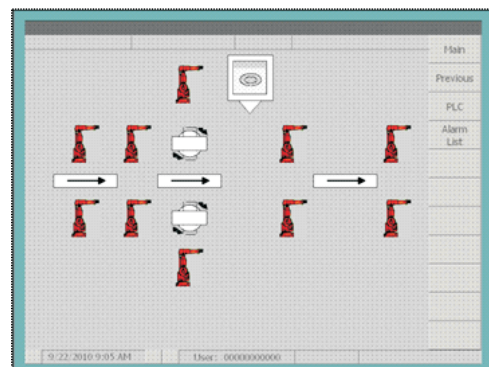

Figure 4. Generated HMI screen (top level).

2.6 Verification and validation of the results

The PLC code has been generated according to the standards regarding naming, structure, numbering of blocks and alarm handling. The generated project does not contain any function blocks or data blocks other than the ones needed and used for the case study project.

The screens generated for the HMI have the right information, but the names do not correspond to the Volvo Car standard. It is the same problem with the tags, however

the tags have the right connection to the PLC code in the form of a data block address. Automation Designer is not capable to do the connection between HMI screens and therefore a manual connection needs to be done inside the HMI software before a validation can be performed according to [2].

3 Discussion

In this case study the results of the generated PLC code is according to the Volvo Car standards. However, the generated code does not include the overall sequence which makes the code hard to validate. The generated PLC project can work as a better base project than the one used by the company today. Advantages with the generated PLC project is that the generated program only includes the function and data blocks used for the current PLC project. No unnecessary information is stored in the project. Another advantage is that the majority of the needed variables in the data blocks are defined and every data and function block has been set with the right name and number. The alarms handling have also been defined inside the PLC project which is very important, because this is something that often is forgotten and therefore missed in some of the function blocks.

The most important advantage is that the correct PLC structure for any manufacturer can be generated. This will decrease the risk of getting an inaccurate PLC program structure in the finished project.

The function block library available from the company was successfully integrated into Automation Designer. The library function blocks used in the generated PLC project is collected from the library during the generation of the PLC code to STEP 7. This makes it possible to have only one function block library in one place. Automation Designer can collect the required blocks each time it generates a PLC project. As Güttel et al [3] describes this is an important feature when generating PLC code.

Another main goal after generating the PLC code was to generate HMI screens for the case study stations. The generated screens show that it is possible to get the layout desired by the company; however the names of the tags connected to the screens are not named according to the standards of the car manufacturer. The same goes for the names of the screens which have got a search path from Automation Designer as a name. It is possible to change the names manually in the HMI software and still maintain the address to the data blocks. This case study is also about how to reuse information and one of the tasks was to investigate how information from earlier simulation studies, in this case from Process Simulate, could be reused. As seen in the results it is possible to import an XML file from Process Simulate into Automation Designer and get the tree structure with the same names on the objects as they had in the simulation study. The other advantage of the given 2D layout from the import is also a help for the user of Automation Designer to avoid reading a lot of documents to set up the layout.

It is important for the companies that are going to use Automation Designer to have cooperation between the PLC and simulation department to develop a tree structure which works for both departments. One of the benefits of working together and using a software as Automation Designer is the naming of objects. By setting a name of e.g. a robot in the simulation study, the name will be the same through the whole commissioning phase. This will simplify the documentation of the work when one object always has got the same name everywhere in the company.

To be able to generate information with Automation Designer, templates must be created for the equipment used by the company. This will take some time, but once this is done the company only needs to build up templates when new equipment is purchased.

4 Conclusion

Generating PLC code and HMI screens with Automation Designer will get the same structure and naming standard in every project. Following a corporate standard will be a quality assurance of the PLC code and HMI screens. The case study shows that it is possible to; import an XML file from Process Simulate to get the tree structure, naming of objects and 2D layout, generate PLC code according to a company standard; however logic calculations are not included, generate HMI screens according to a company standard, correctly set the alarm handling of the PLC code within the PLC project.

5 Acknowledgement

References

[1] M. Fabian, K. kesson, and D. Spensieri. Automatic model generation and plc-code implementation for interlocking policies in industrial robot cells. *Proc. of Control Engineering Practice*, 11, pp. 1416-1426(1), 2007.

[2] G. Frey and L. Litz. Formal methods in plc programming. In *Proc. of IEEE Conference on System Man and Cybernetics, Nashville*, pp. 2431-2436, oct 2000.

[3] K. Güttel, P. Weber, and A. Fay. Automatic generation of plc code beyond the nominal sequence. In *Proc. of Emerging Technologies and Factory Automation, Hamburg.*, pp. 1277 - 1284, oct 2008.

[4] IEC. *IEC International Standard 61131-3: Part 3: programmable controllers*. IEC, Geneva, Swizerland, 2002.

[5] B. Mulman, T. Devinder, and W. Gi-Nam. Generation of plc ladder diagram using modular structure. In *Proc. of International Conference on Computational Intelligence for Modelling, Vienna*, pp. 1194-1198, oct 2008.

[6] S. A. I. Sector. *Manual SIMATIC Automation Designer based on Comos*. Siemens, Nürnberg, Germany., 2010.

[7] Siemens. http://www.plm.automation.siemens.com/en_us/products/tecnomatix/assembly_planning/process_simulate/index.shtml, 2010-04-12.

[8] Siemens. http://www.automation.siemens.com/mcms/automation-software/en/digital-engineering/automation-designer/Pages/Default.aspx, 2010-04-12.

[9] Siemens. http://www.automation.siemens.com/mcms/simatic-controller-software/en/step7/Pages/Default.aspx, 2010-04-12.

[10] Siemens. http://www.automation.siemens.com/mcms/human-machine-interface/en/visualization-software/wincc-flexible/Pages/Default.aspx, 2010-04-12.