Master Thesis CODEN:LUTEDX/(TEIE-5208)/1-114/(2005)

Industrial Electrical Engineering and Automation

Interactive Control of a Virtual Machine

Joakim Davidson, Electrical Engineering

Tobias Sennö, Mechanical Engineering

Department of Industrial Electrical Engineering and Automation
Lund University

2005

Master Thesis in Industrial Automation

Interactive control of a virtual machine

Joakim Davidson and Tobias Sennö

2005

LUND UNIVERSITY
Department of Industrial Electrical Engineering and Automation

In co-operation with

Tetra Pak
protects what's good™

Supervisors:

Gustaf Olsson, Professor, Industrial Electrical Engineering and Automation, Lund University, Lund

Istvan Ulvros, Senior Technology Specialist, Tetra Pak Research and Development, Lund

Department of Industrial Electrical Engineering and Automation (IEA) LTH
Lund University
Box 118
SE-221 00 Lund
Sweden

http://www.iea.lth.se

Printed by Tryckeriet i E-huset
Lund, Sweden 2005

Summary

The purpose of this Master thesis is to examine the market for simulation software that can control a model of CAD-geometry by PLC-code, and then evaluate selected software.

The main idea is to use existing CAD-geometry, add kinematics and logics, and then control this by PLC-code. The purpose of the simulation is to discover problems with the geometry, i.e. collisions, early in the development process. When the geometry is determined, the verified PLC-code can be downloaded to the real PLC and connected to the real model. The real machine will then behave exactly as the virtual machine.

The project is done at Tetra Pak Research and Development in Lund, Sweden. The thesis starts with defining requirements for a simulation tool to be used in the developing process at Tetra Pak. The simulation software at the market that best fulfills the requirements will be evaluated using realistic hardware designs.

The software selected is Delmia Automation and is delivered by Delmia. The model to be used for the simulations was created by importing existing geometry from Tetra Pak´s CAD-software, while logics and kinematics were added in Delmia Automation. The model was controlled by a virtual PLC. To handle this a virtual Human Machine Interface was created.

The result of the project is that Delmia Automation can perform a simulation of a model controlled by a virtual PLC, but there are some significant limitations.

Because of a delay in the delivery of the simulation software and the number of problems and lack of documentation, the last step in a complete simulation was not accomplished. This step involves control of the virtual model and the real model by a real PLC.

The recommendation to Tetra Pak is to do further works in this area, because simulations can save a lot of time and money in the development process. There are some major requirements that Delmia Automation today not fulfilled and for that reason it can be hard to entirely implement it in Tetra Pak’s development process at this stage.

Preface

This master thesis was done at Lund Institute of Technology (LTH) at Lund University for a Master of Science degree. The project treats a mechatronic problem and the authors' background is one from electrical engineering and one from mechanical engineering. The study was initiated by the department of Industrial Electrical Engineering and Automation (IEA) and was carried out in co-operation with Tetra Pak Research and Development.

First of all, we would like to thank our supervisor Professor Gustaf Olsson at IEA for his arouse enthusiasm in share of the work.

We will thank all employees at Tetra Pak that has been involved in this project and especially our supervisor at Tetra Pak, Istvan Ulvros.

We would also like to thank Delima and Eric Retraint for their interest in our project and for supplying us with necessary software. Lise Upjohn at Delia Automation's support has been an indispensable access for help with technical questions.

Lund, October 2005

Joakim Davidson   Tobias Sennö

Nomenclature

CAD	Computer Aided Design
CPU	Central Processing Unit
DA	DELMIA Automation
DOF	Degree of Freedom
FBD	Function Block Diagram
HMI	Human-machine interface
I/O	Input Output
IC	Integrated Circuit
IEC	International Electrotechnical Commission
IEEE	Institution of Electrical and Electronic Engineers
IL	Instruction List
LD	Ladder Diagram
MSE	Mechanical Simulation Engine
OLE	Object Linking and Embedding
OPC	OLE for Process Control
PLC	Programmable Logic Controller
Pro/E	Pro/Engineer, 3D CAD-software from PTC
R&D	Research and Development
SFC	Sequential Flow Chart
SRS	Software Requirement Specification
ST	Structured Text
STEP	Standard for Exchange of Product model data, neutral file format in CAD-software
VRML	Virtual Reality Modeling Language, neutral file format in CAD-software

Contents

1 Introduction

This chapter is an introduction to the master thesis and explains why Tetra Pak is interested in this topic. In the end of the chapter there is a short description of how the master thesis project has proceeded and an outline of the report.

1.1 Background

Tetra Pak R&D develops packages and filling machines for the food sector. The new filling machines are developed at the R&D Department of Machine Technologies, where also the capacities of the existing machines are increased. They work with the whole process from the in-feed of material to the package outlet via the filling process.

When a new machine is developed it is necessary to build up a real machine at least one time. This machine is often built in stainless steel, which is an expensive material and hard to work up. The prototype is not perfect in the mechanisms, so changes are necessary to do. In this way the tests demand a lot of time and money. The software for the control of the prototype starts to take shape first after the prototype is built, which causes further time delay in the development.

Tetra Pak R&D wants to find a method that improves the development process, this can be done in many ways. If the number of errors is reduced before the prototype is built, both money and time will be saved. By developing the software in parallel with the mechanics also the process will be improved. If it is possible to reject a non-working concept at an early stage of the process much effort will be saved.

By doing simulations it is possibly to fulfill these ideas. With simulation it is also possible to make as many tests as wanted, and a failing test will not cause any real parts to break. It is also easier and cheaper to change the mechanics in a virtual machine than in a real one.

Tetra Pak wants to investigate the possibility to control a virtual machine with a real PLC.

The project treats a mechatronical problem, which includes and combines techniques from electrical and mechanical engineering. To take care of the opportunities and challenges in this topic the authors’ background is from one of each faculty.

1.2 The task

This master thesis focuses on the possibilities to simulate a virtual machine.

The first part of the master thesis was to make an investigation of the market for simulation software. The software should be able to simulate a virtual machine together with a real PLC and the main reason to do a simulation is to verify the mechanisms of a machine.

After the investigation, in the second part, one or more software platforms should be chosen for practical evaluation. Tetra Pak uses the CAD-software Pro/E for design of all their machines. For that reason the simulator must have the ability to import a CAD-model from Pro/E. The simulator must also handle signals between the model and a PLC. The simulated model should behave as a real machine, and the reaction of the signals from the PLC-system should also be as in reality.

This thesis will deliver a survey of the simulation market and an evaluation of selected software product. With the software a simulation of a virtual machine will be made, controlled by a PLC, and the results will be compared with a real machine. This will also result in a demonstration arrangement at Tetra Pak R&D.

1.3 Delimitation

In this master thesis there were several delimitations made. The delimitations discussed in this chapter concern the thesis in general. The limitations of the software will be discussed later on in a market survey of the software. The limitations of the chosen software will be discussed in Chapter 5.3.

Today Pro/Engineer from PTC is used as CAD-software at Tetra Pak R&D. At the moment it is not planned to change CAD-software so the simulation software must be able to import geometry from Pro/E. A survey of CAD suppliers will not be made even if there are suppliers who have CAD-software that is more integrated with the simulation software than Pro/E is.

A model of a simple machine has been used to be simulated. Complex models have been avoided, since the time for a master thesis would not allow evaluating and doing an exhaustive research of all processes in creating a complete simulation.

This project is limited in time. As a master thesis it is planned to take approximately 20 weeks to complete, including writing of this report and presentation of the thesis.

The economy and the price of the software were of subordinate significance, the primary property is the functionality. Because of none of the software products that are surveyed in this thesis are used at Tetra Pak today, a loan arrangement during this master thesis was preferred but not a limiting factor.

The house language at Tetra Pak is English and therefore this thesis is written in English. Another reason is that some of the people at Tetra Pak R&D who are interested in this project do not speak Swedish.

1.4 Target group

The target groups for this master thesis are persons at Tetra Pak, and other companies, who work with machine development and programming of control systems with PLC. Traditionally this is done by different persons but this thesis will show the possibilities to incorporate it in a mechatronic way. Another target group is electrical and mechanical engineer students at the end of their education. Because the targets are from two different disciplines, parts of the report can feel to elementary for their own discipline.

1.5 Company presentation

This master thesis was done by order of a company, Tetra Pak. In this chapter a presentation of the group of companies that forms the Tetra Laval group will be done.

Tetra Laval

Tetra Laval is the name of three industry groups that are focused on systems for production and distribution of foods. The three industries are Tetra Pak, DeLaval and Sidel. They are all autonomous. The name Tetra Laval was founded in 1993 after Tetra Pak acquired Alfa Laval and its activities were integrated[1]. Alfa Laval was issued in 2002 at the stock exchange and no longer belongs to the Tetra Laval group.

Today Tetra Laval is organized as in Figure 1.1.

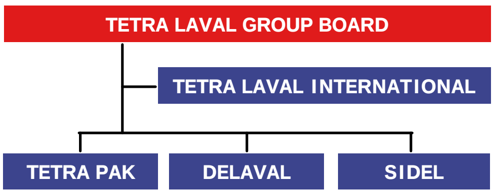


Figure 1.1: Tetra Laval organization scheme[2].

Tetra Laval International

The Tetra Laval International is responsible for the financing in the Tetra Laval Group. The company assures that the standards and operations from Tetra Laval Group Board are fulfilled in the three industry groups[3].

[1] Nationalencyklopedins Internettjänst, Tetra Laval (Internet), 2005-04-08
[2] Tetra Laval, About Tetra Laval (Internet), 2005-04-08
[3] Tetra Laval, About Tetra Laval (Internet), 2005-04-08

Tetra Pak

The founder of Tetra Pak was Dr. Ruben Rausing. He started his career together with Erik Åkerlund by founding the packaging company Åkerlund & Rausing, Å&R, in 1929[4]. In this company the development of a package for flour and sugar began. After some years of success a development of a package for milk started. The packaging system that Å&R developed was not only a revolution by packaging technique but also the package material, paper with a layer of plastic. As a result of this package the new company Tetra Pak was founded in 1950.

In 1951 AB Tetra Pak was established in Lund, Sweden. Later this year, 18th of May, a new packaging system was presented. This was a system that created a center of attention from the press. In September one year later the first system was installed at Lundaortens Mejeriförening[5]. From now on the system was spread to dairy companies all over Sweden. Some years later the installation of Tetra Pak machines in dairy companies in Europe began.

The development result of a new package was launched in 1963, the rectangular shaped carton, Tetra Brik. This invention became a major product for Swedish dairies[6]. This was also one of the reasons why Tetra Pak made their breakthrough worldwide in the sixties.

From this point Tetra Pak expanded worldwide and introduced new products every year. In 1985 Tetra Pak acquired Alfa-Laval Logistics AB, a company involved in the development and marketing of distribution equipment for the dairy and food industries[7]. In 1991 this was finalized and the Tetra Pak Alfa-Laval Group was formed.

One of Tetra Pak's latest products is the new packaging system Tetra Recart. The material in Tetra Recart is resistant to fruit acid, which makes it to a competitor to the traditional tin, but still it can be recycled as cardboard. The Tetra Recart package can be retorted, a high-pressure cooking in an autoclave, to make it aseptic. This is done when the package is filled and sealed. Retorting was not possible earlier with other packages

[4] Region Siljan, De som lyfte landet (Internet), 2005-04-08
[5] Tetra Pak, The Company History (Internet), 2005-04-08
[6] Tetra Pak, The Company History (Internet), 2005-04-08
[7] Tetra Pak, The Company History (Internet), 2005-04-08

made of paper and makes the Tetra Recart usable in many different applications.

Today Tetra Pak is one of the largest packaging companies in the world with over 20 000 employees in 165 countries around the world, and net sales of 7.5 billion EUR[8]. In 2003 Tetra Pak broke the magical mark of 100-billion-packages delivered in a year[9]. Tetra Pak is a fast growing company, especially in China.

DeLaval

In 2000 Alfa Laval Agri changed name to DeLaval. DeLaval supplies complete system for the dairy industry. They manufacture and market everything between barn floor and cow brushes to milking systems[10].

Sidel

In 2003 the French company Sidel was purchased[11]. Sidel is one of the world leading companies for plastic bottles. They manufacture blowing and filling machines, and they also design bottles. The bottles are used for liquid food such as water, oil, milk, tea, fruit juices and beer[12].

1.6 Overview of the project

This project was started in March 2005 by defining the main purpose of the master thesis. Interviews with people at Tetra Pak were done to get foundation to form the definitions.

A survey of the players on the software market was the next step in the project. Simulations where a virtual machine and a real PLC are connected to each other are a young business area and therefore it was necessary to

---

[8] Tetra Pak, In figures (Internet), 2005-04-08
[9] Tetra Laval, Tetra Laval 2004 (Internet), 2005-04-08
[10] Tetra Laval, About Tetra Laval (Internet), 2005-04-08
[11] Halogen, Tetra Laval (Internet), 2005-04-08
[12] Tetra Laval, About Tetra Laval (Internet), 2004-04-08

make a software requirement specification (SRS) that clearly describes what the software must handle. Out of the SRS an investigation was made to get knowledge of which players there are on the market and what their products manage. From the investigation the software was selected for further evaluation. This part of the project was made during most part of April and the first part of May.

The software to use for the simulations was selected and a copy from the manufacturer was demanded. Although the manufacturer promised to send it in a short time, it was not received until three weeks later. The theoretical parts of the report were written during these weeks, while the contacts were established with the vendor.

A copy of the selected software, Delmia Automation, was received at the turn of June. It started with a self-learning process of the software, where a knowledge base was built up to be able to do a complete simulation. A model of a simple machine was developed and simulated. A result of the test showed that it is possible to control a virtual machine with a virtual PLC.

After the tests the model of the simulated object was built up in DA. This took place during the second part of June and July.

From August the work was concentrated to write this report and do a presentation of the project.

1.7 Outline of the report

This report can be divided into three parts. The first part is an introduction to the master thesis and virtual simulations, consisting of Chapters **Error! Reference source not found.**, 2 and 3. In the first chapter there is an overview of the task, the client and the master thesis. In the next chapter are some discussions about why doing virtual simulations. Chapter 3 is about the working methodology of the master thesis.

Next part is a market survey, found in Chapter 4.

The last part is the practical result of the master thesis, Chapters 5, 6 and 7. In Chapter 5 there is an overview of the used environment. Chapter 6 describes how a simulation can be implemented. In Chapter 7 the conclusions are found as well as the recommendations to Tetra Pak and for future works.

In Appendix A the Software requirements specification for the assessment is found. In Appendix B a list of signals that was used in the simulation is presented and in Appendix C the code are described. The comments sent to Delmia as a reflection of the simulation project are recorded in Appendix D.

2 Controlling the virtual system

To take both the control system and the mechanical system into consideration under simulation is quite an unexplored area. Therefore there will be an introduction in this chapter to the fundamental ideas and what kind of research that has been done. There are also some general aims why to do simulations.

2.1 The idea of control and machine simulation

To simulate means to mimic or imitate the behavior of a system[13]. By using a simulator the behavior of a model can be studied and how it will react, depending on which control signals that are influencing the model. The behavior can also be studied depending on different parameter values, model complexities and initial conditions.

There are two types of systems to simulate, dynamical systems and discrete event systems. Both of them can be simulated, but the result of a simulation will never be better than the models and the data that are used. This must always be remembered when the results are presented.

Discrete event and dynamical simulation

A discrete event simulation is for example simulation of queues in manufacturing systems. This type of simulation works with events. Events can for example be arrival of a job, completion of a job or a machine that breaks. Everything that changes the state of the system is an event. Therefore a discrete event simulator is said to be event driven, because the system only changes at an event time.

A dynamical system is different from a discrete event system. In a dynamical system the model is given in differential or difference equations instead and they are dependent of time. Both differential and difference

[13] Olsson, Gustaf – Rosen, Christian; (2004); *Industrial automation*, Ch. 8

equations can be used to define the model and these equations must be solved to run the simulation. The initial conditions for the model must be known. The equations can be used together with conditional expressions, i.e. if-then-else. The equations of the dynamical model can be used in two ways in a dynamical system, depending on the software. To describe the system either a conventional text editor, an equation oriented description, or a block oriented description can be used. An example of block oriented description is Simulink, which is a part of Matlab, from The MathWorks. In block oriented description the model and the different parts of the system will be defined in different blocks, such as input generating functions and integrators. Then these symbols and blocks are connected in a flow diagram to describe the dynamics of the process.

A dynamical system uses continuous time or discrete time-steps and the system will change its states more or less continuously. The difference between the systems is that the discrete event systems are run in steps, while the dynamical systems are solved. Therefore a dynamical system simulator is said to be time-driven.

The results of a simulation

Often the most essential part of the simulation is the results of the simulation. It is therefore important how the results are presented, how they can be saved, and interact with other software systems to analyze the output data. If it is possible to connect the simulator to a database with real data, or to a real-time input, it is possible to incorporate real data in the simulation. With a real-time input it is possible to do on-line analysis and diagnosis.

For the discrete event simulators there are other parameters that are important in the simulation results, for example in a manufacturing system, such as utilization rate, queue lengths and production rates - both for the whole system and for each machine.

Connection between traditional simulators and PLCs

In traditional design processes the machines are simulated in one system and the PLC's control software in another system. These systems are not connected to each other and they could not be verified as one system before the prototypes are built up. The idea with the simulations made in this thesis is the connection between the systems. Both the parts will be integrated in the same system. The system consists of four parts, a virtual machine, a virtual PLC, a real machine and a real PLC. In the first part of a project only the virtual machine and the PLC are used. Many ideas and codes can be tested on the virtual machine. When the geometry of the machine is fixed, the machine can be build up and the real PLC can be ordered. Then the real systems can be verified against each other. The real PLC can also be run against the virtual machine during the development process. The idea is visualized in Figure 2.1.

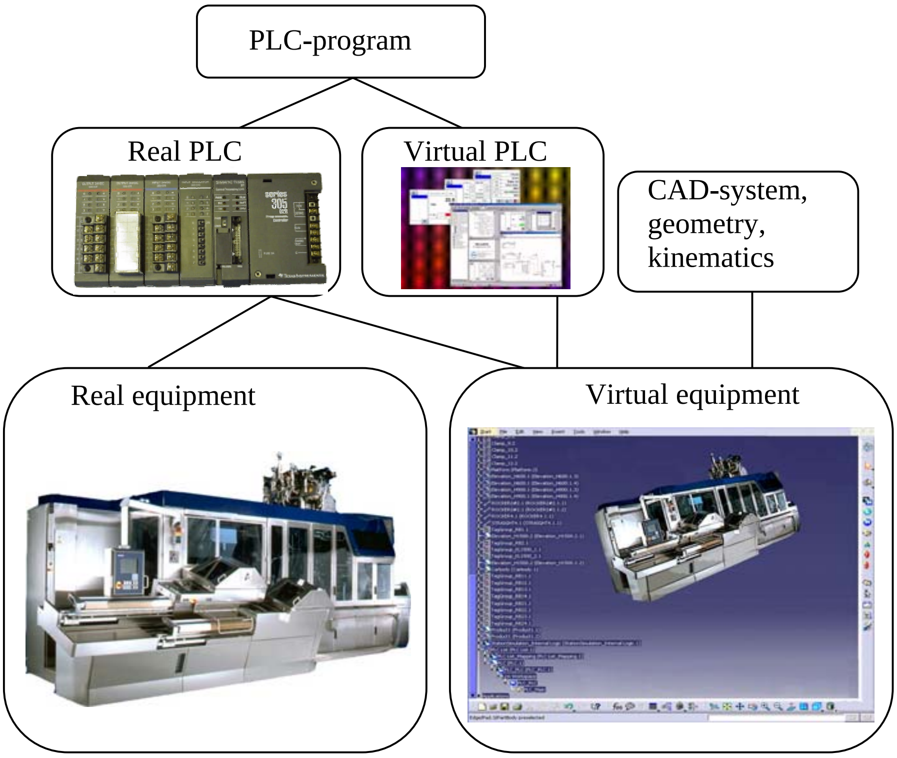


Figure 2.1: The idea of mechatronic simulation.

The connection between the virtual machine and the virtual PLC can be done in several ways. The easiest case is if the software comes from the same manufacturer and already has got a communication link between the different parts. This is the case with Delmia Automation. Another case is if the software communicates with a bus, i.e. Profibus. In this case it can sometimes be necessary to write a handshaking-protocol. This can be difficult to do and if the equipment is changed the handshaking-protocol must be rewritten. The third way to communicate is via Object Linking and Embedding for Process Control (OPC). OPC is open connectivity via open standards. To use open standards is an advantage because many software systems on the market can use OPC, but it is also a big drawback because the communication over OPC is slow. Therefore many of the software products on the market can handle several types of communication. This usually includes both OPC for the compatibility and an own developed system with much higher performance.

2.2 Previous work

There have been some other master theses done at Tetra Pak with the purpose to simulate a virtual machine.

In a master thesis by Jan-Olof Sivtoft a communication core between FrameworX from GE-Fanuc and Working Model 3D Motion from MSC Working knowledge was implemented. FrameworX is a soft PLC and Working Model 3D Motion is a mechanical motion simulation software. The conclusion of the master thesis was that it is possible to control a virtual machine with a soft PLC. In the conclusion he states that there are some drawbacks with this solution. Both software systems are running on the same computer, which makes the simulation far from real-time. The interface in Working Model 3D Motion becomes hard to work with when the model is complex[14].

Later there were two more theses at Tetra Pak, and they have used software from the Finish company Lumeo. The software was an add-on module in Pro/Engineer.

In one of these, a master thesis by Ola Hammarlund and Dan Persson a communication interface with Profibus between Lumeo's software and a PLC was designed. The conclusion of this thesis was that it is possible to

[14] Sivtoft, Jan-Olof (2002); *The design of a simulator for the automation industry*

verify a machine and control a machine simulated with Lumeo's software by a commonly used fieldbus[15].

A bachelor thesis by Annika Olofsson and Marie Lundqvist that connects to the thesis by Hammarlund and Persson was done. In this bachelor thesis they have done a PLC-program to a prototype of a part of a machine. The aim was to use the same PLC to control a virtual and a real machine but according to problems during the project the PLC-program was not totally completed. But the part of their program that was completed, works fine. It communicated with both the real machine and the virtual model and they both responded identically to the PLC-program[16].

Unfortunately Lumeo was dispersed from the market and their products are no longer available for use.

There are some other theses done at the topic of simulation and offline verification of virtual machines. Most of them handle robots and off-line programming of robot cells. It is mainly the automotive industry that urges this development[17].

At Chalmers University of Technology a master thesis in verification of auto-generated PLC-programs has been presented. In this thesis they use eM-PLC from Tecnomatix and Control Builder from ABB. The aim was to know how much the PLC-code must be changed if additional equipment is added, i.e. one more clamp, or if the sequences of operations of the cell are changed. The robot cell that was used got two industrial robots and two clamps. The verification was done against a virtual model that was reproduced from a CAD model[18].

---

[15] Hammarlund, Ola - Persson, Dan (2003); *Real Simulation Control, Design of Communication for a Simulator Device*

[16] Olofsson, Annika – Lindqvist, Marie (2003); *Simulation and programming of a prototype machine as a simulator*

[17] CIMdata, The Benefits of Digital Manufacturing (Internet), 2005-05-13

[18] Gore, Ganesh – Sharique, Farid Syed (2004); *Virtual Verification of Auto-generated PLC programs*

2.3 Benefits of simulations

What is the purpose of doing simulations and off-line programming instead of doing real things and on-line programming? The short answer is time and money. Today there are less margins for mistakes, and with simulation it is possible to discover errors early in the development process. If the right things are done directly with the real equipment a lot of money can be saved, see Figure 2.2. This will also give an increasing productivity and decreasing development time. This gives a shorter time to market.

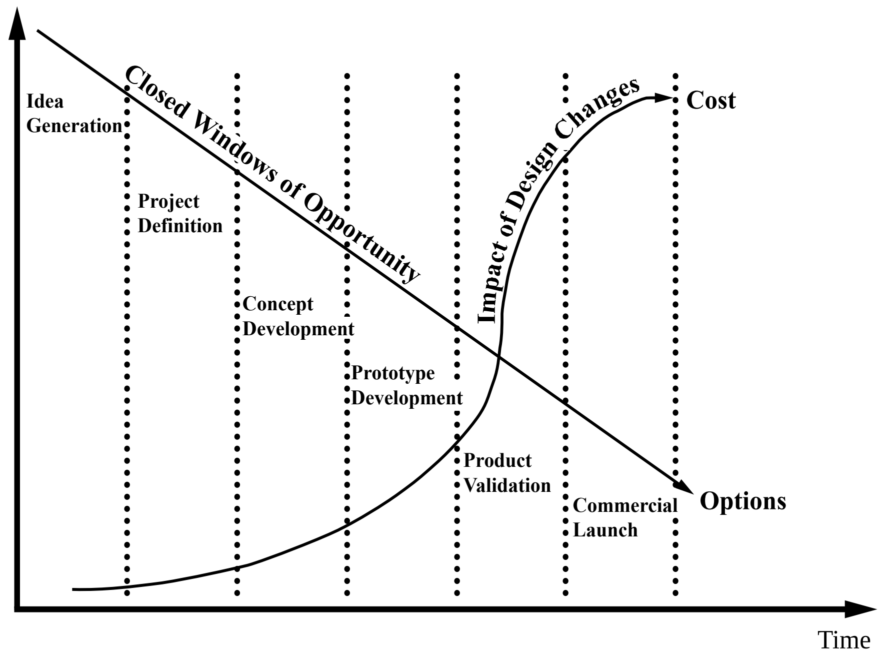


Figure 2.2: Cost of doing a change in the development process[19, 20].

In the last decade the interest in simulators has increased significantly. Two reasons to this are that computing calculation capacity has increased

[19] Tech-Clarity, Digital Manufacturing (Internet), 2005-05-24
[20] Tetra Pak, Tetra Pak Innovation Network (Internet), 2005-09-19

enormously and that the competition between the actors on the market has increased. By doing 3D-simulations the results can also be visualized for a non-technical person.

There are great benefits to do off-line programming instead of on-line programming because the production does not have to be stopped. If the code also can be verified off-line against a virtual model and correct what is wrong before the production phase, a lot of time and money can be saved. In a gantry crane project within a Ph. D thesis the downtime was reduced from approximately 10 weeks to 2 hours[21].

By doing simulations it is possible to start the development of the control system earlier. Today the developer of the control system needs to wait until a prototype is built, to make it possible to run their program on it. By using simulators and virtual machines the code can be run and tested on the virtual machine before the prototype is built. This way of work can also discover problems and incorrectness in the construction and geometry, which can be corrected before the real prototype is built. It is also possible to reject concepts and ideas that not work, before the real prototype is built. To reject them as early as possible in the development process will save a lot of time and money, that can be used to develop concepts that work even better. By using simulators and virtual machines there will not be a queue to the real prototype machine for testing the code. Different developers can instead simultaneously run their own code on different virtual machines. If many people are involved in the development process and only one real prototype is available there can be a lot of waiting before the prototype is unoccupied.

Another important purpose for simulations is the training of operators. This has been used for many years in some critical industries, i.e. pilots for airplanes are trained in flight simulators and operators of nuclear plants are trained in full scale simulators of the control room. By doing this training, the operators can train on unusual and dangerous situations, without hurting any people and damage of material and machines. Simulation makes it possible to experiment with reality.

---

[21] Danielsson, Fredrik (2002); *Off-line programming, verification and optimisation of industrial control systems*, p93

One example of using simulators in the industry is the development and production of Volvo's new SUV XC90. This was developed, tested, corrected and reconstructed in a virtual factory. This gave the opportunity to start the production in 20 % shorter time that had been possible in a traditional way. Volvo also trained their operators and assembly workers at the production line. These digital development processes gave the result that instead of manufacturing three test runs of cars they only needed to manufacture one test run. The production started one year earlier and the cost savings were, conservatively calculated, hundreds of millions of SEK to develop and adjust the car to the production[22].

[22] Ny Teknik, Volvo lär Ford att bygga bilar i datorn (Internet), 2005-04-14

3 Methodology

How to organize the work is an important part of the master thesis. If there is a good profile of the work from the beginning it becomes much easier to finish the project in right time and with a good result. Another important part is the information, how it is collected and critically inspected.

3.1 Work method

The first part of this master thesis was to do a software requirements specification. This software requirements specification was done in an iterative method. The more that was learned by the way, the more specified and correctly were the requirements. To get the information of the different software different kinds of sources were used.

Parallel with the development of a SRS a literature study was initiated. This study has its main part in the first period of the project, but it has continued during the whole project when new questions have come up. A real advantage with a literature study is the possibility to obtain much information in a short time with small economical resources[23]. The literature study was done to acquire knowledge from earlier investigations in the topic of our thesis and also knowledge of theories, models and methodologies[24]. Together with the information from Tetra Pak R&D this formed what was essential for the project. It was also in help when the delimitations of the project were made.

3.2 Information sources

Primary sources used for information was the Internet, University Libraries, staff at Tetra Pak and the support at Delmia. The main purpose was to find software able to fulfill the task for our thesis. An important source to get names of possible software to use was the supervisor at Tetra Pak, Mr. Istvan Ulvros. On the Internet general search engines, such as

[23] Björklund, Maria – Paulsson, Ulf; (2003); *Seminarieboken*, Ch. 4
[24] Patel, Runa – Davidson, Bo; (1994); *Forskningsmetodikens grunder*, Ch. 3

Google, were used and also different forums and portals for automation. The manufacturers' home pages were used to get detailed information of the software features. The vendors were also contacted to get a clearer answer to some questions. To get an own conception of the software features and limitations a trial run was done of some of the software products, where this was possible. Most part of the information comes from the manufacturers of the software but other sources were searched, for example where the software was used by independent professionals and specialists.

In April 11-15, 2005 the Hannover Industry Fair took place and at this fair the four vendors in the second part of the assessment were represented. One of the possible vendors introduced their new product on the market at this fair. The supervisor for this master thesis at Tetra Pak went to Hannover and this gave the opportunity to meet people from the vendors and discuss the features and possibilities of their new product.

At Lund University Library (LUB) several searches were made for material of research methodology and earlier theses of simulations. From the department of Industrial Electrical Engineering and Automation (IEA) some theses were borrowed and via the Studiecentrum at Lund Institute of Technology (LTH) several theses were ordered from Chalmers University of Technology, and also one Ph. D thesis from De Montfort University in the UK.

3.3 Criticism of source

This master thesis considers a new work method to verify a machine, both the mechanics and the PLC-software. Therefore most of the information, to evaluate the simulation software, came from the vendors. The vendors often present the software in a bright way, since they have an economic profit in the product. The vendors try to get the reader to think in a way that fits exactly their product, and if they are succeeded the customer will probably buy their product. Projects where the product has been used has often been done by the company itself or a collaborator to it and therefore also this information is slanted in a favor for the vendor. Another experience is that the information from the vendors is not always correct, some functions in the software are used for sales argument and they will be implemented in a future version. The survey has been done with these things in mind. A simulation of a product from Tetra Pak has been accomplished to give an own opinion of the software and to do a correct evaluation of its advantages and disadvantages. As they work in practice and not how they are described by vendors.

4 Market survey

To do good simulations it was necessary to do a market survey to get knowledge of which actors there are on the market. In this chapter there will be a discussion of the requirements of the software and how the vendors meet these requirements.

4.1 Requirement of the software

In the beginning of the project a software requirements specification (SRS) was made. At first a proposal was made and then this was discussed with the staff at Tetra Pak. The process was iterative and the proposals were changed several times to get the final result. The requirements were graded in three steps depending on how important they were. Totally there were 31 requirements at the list. The full list of SRS is in Appendix A.

One of the most important requirements is that the software can import geometry from Pro/E, the CAD-software they use at Tetra Pak. It is also required that the software can handle control signals from a PLC and to get a visual display of the model during the simulation. Another requirement is that the software and documentation must be available in English, the house language at Tetra Pak. There were also requirements that the software will run on a PC with MS Windows, that internal logics can be defined and that a review of the simulation will be in real-time. Because of demanding calculations during the simulation, it is most probably that the simulation will not be in real-time and therefore it must be an automatic synchronization of emulated real-time between the PLC and the virtual machine in the PC.

4.2 Vendors

At the beginning there were proposals of vendors that got software to handle the type of simulations that will be done. It was only proposals and several extensive searches on the Internet were made to find even more vendors. The searches gave some more candidates and they are included in the list of vendors. The assessment of the vendors was made in two steps. In the first step the software was appraised with some simpler requirements together with the general impression of the software. In this step ten vendors were available who had software that possibly could do the simulations. They are listed in Table 4.1.

Table 4.1: The ten vendors and simulation software in the first assessment.

Vendor	**Software product**
Cimetrix	CODE
Cosimir	Cosimir PLC
Delmia	Delmia Automation
Fiebach-Datentechnik	E-control
Mewes & Partner	WinMOD
Mitsubishi	GX Developer
Rockwell	RSTestStand + RSLogix
Tarakos	taraVRcontrol
Tecnomatix	eM-PLC
Woodhead	SST Pics

Because of different reasons six of the possible vendors and their software did not qualify to the second part of the assessment. Cosimir PLC and Woodhead SST Pics aimed at training of operators. SST Pics is mostly aimed for the process industry and Cosimir is developed for the simulation of robots. In Mitsubishi GX Developer it is possible to test the software but there is no feedback from the model. In the software CODE from Cimetrix it was not possible to import geometry from a CAD-system and the software does not seem to have been developed during the last three years. Except for this the software had potential to be good software for simulation purposes. There was not sufficient information about E-Control from Fiebach-Datentechnik, but it has the possibility to import CAD geometry and to define sensors. A request for more information was sent to them and an answer came that they should send it, but despite reminders they didn’t send the information. The eM-PLC from Tecnomatix has the direction of auto-generating PLC-code. The sequences will be defined and then the PLC-code will be auto-generated. This code can then be verified against a virtual model.

After the first step in the assessment had been completed there were four vendors left: Delmia, Rockwell, Mewes & Partner and Tarakos.

After the survey was done and a vendor was selected a new software, MSC.visualNastran 4D, was found. Because it was found so late there was no complete evaluation done. It has some good potential, and can easily be integrated with the CAD-software and changes can easily be transferred into MSC.visualNastran 4D. It uses friction and gravity so the simulation can be realistic. There are different kinds of possible analysis, e.g. collision and velocity. The system can be controlled by Matlab and Simulink or via Visual Basic. If Visual Basic is used it can be connected to a PLC. This type of control, with a second program and a specialized communication link feels like going backward in the development. Communication by special written communication links and protocols has been used in earlier master theses at Tetra Pak and was problematic and caused many problems.

In the second step a more extended assessment was made. A presentation of the qualified vendors and their products will now follow.

Delmia – Delmia Automation

Delmia Automation (DA) is a new software product on the market delivered by Delmia. The availability was announced at the Hannover Fair in April 2005. DA enables the users to digitally define, control and monitor automated systems. PLCs can be programmed and the logics can be validated against a virtual machine, cell or an entire line. The PLC-code from DA must pass a special PLC Setup to convert the code to a specific PLC, like a postprocessor. With DA it is also possible to incorporate the Human Machine Interface (HMI), real or virtual. In Figure 4.1 a screenshot of the interface of Delmia Automation is shown.

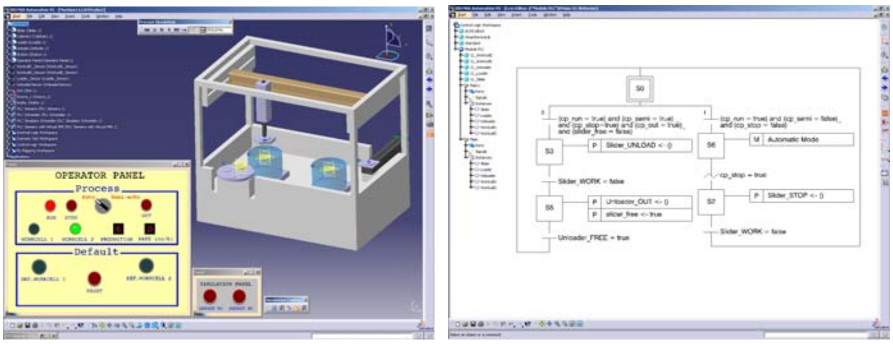


Figure 4.1: The interface of Delmia Automation[25].

[25] Screenshot from an example file made by Delmia, Machine14

Rockwell – RSTestStand

Rockwell offers the software RSTestStand, which is designed to test control systems off-line. RSTestStand got a software based PLC emulator for the PLC logic. RSTestStand is available in three versions, light, standard and enterprise. The enterprise version was used for the assessment. This version can integrate with Logix 5000, SolidWorks/Catia and Simulink for example. Another benefit is the possibility to make connection diagrams. In Figure 4.2 a screenshot of the interface of RSTestStand is shown.

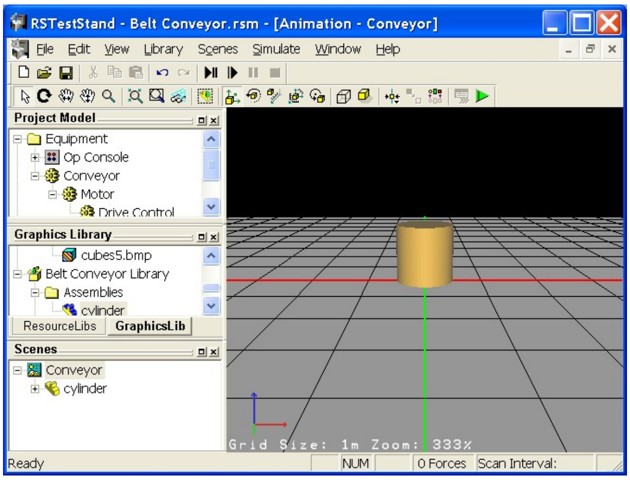


Figure 4.2: The interface of RSTestStand[26].

[26] Rockwell Software, RSTestStand Enterprise, Getting Result Guide (Internet), 2005-05-30

Mewes & Partner – WinMOD

The company Mewes & Partner offers the software WinMOD. This software is for real-time simulation, signal-management and for serving and observing functions. Devices and component-assemblies will form the virtual machine that is implemented as WinMOD-macros. WinMOD can be connected to a real PLC via Profibus or to another PC via Ethernet. With WinMOD it is possible to incorporate the real HMI in the simulation system. In Figure 4.3 a screenshot of the interface of WinMOD is shown.

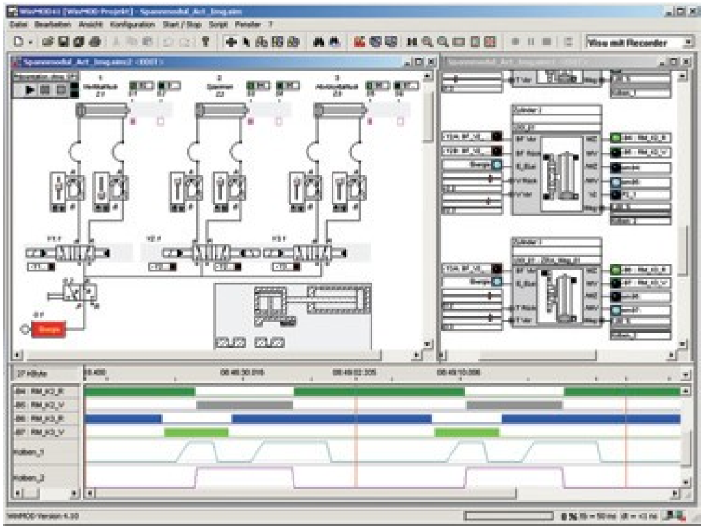


Figure 4.3: The interface of WinMOD[27].

[27] WinMod, Virtual machine for mechanical engineering (Internet), 2005-05-30

Tarakos – taraVRcontrol

Tarakos offers the software taraVRcontrol. This software is designed to visualize 3D-process scenarios. The taraVRcontrol offers easy-to-handle collision checks and easy set-up of sensors. There is no PLC emulator in taraVRcontrol but it can handle process variables from other software products via OPC. In Figure 4.4 a screenshot of the interface of taraVRcontrol is shown.

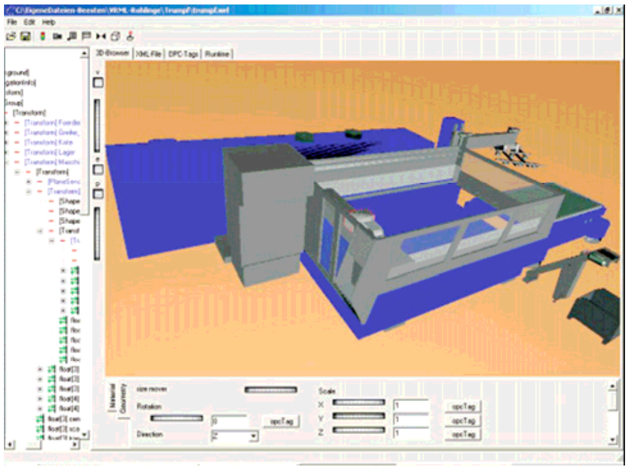


Figure 4.4: The interface of taraVRcontrol[28].

4.3 Satisfying the requirements

In the second step, as said earlier, a more extended assessment was made. To get an answer what the different software products are able to do many different sources were used. The vendor's homepages were used to get information and searches on the Internet were made. Other sources were reports, papers and articles in different magazines. Representatives of the vendors were contacted to provide answers on various questions that had not been clearly answered. Unfortunately some of them did not answer our

[28] Tarakos, IEE 11-2004, Fertige Programme für fictive Maschinen (Internet), 2005-05-30

questions. Some of them said they would return with answers and some did not answer at all. The representative from Delmia answered fast and was very positive and interested in our project. This has influenced the choice of vendor in the way that there were answers to many of the questions in the SRS. Also the representative for WinMOD has answered our questions in a good way, but their answer took a long time to arrive and when their answer arrived the vendor was already chosen.

Two of the vendors, Rockwell and Tarakos, got evaluation versions of their software available. From Tarakos it was possible to download it free and from Rockwell to order a CD with the software. A CD was ordered from Rockwell, but it was never delivered. Several mail were also sent to Rockwell with questions of their software RSTestStand, but they did not answer it. Instead the information of their software came from their homepage and what other people, not from Rockwell, had written about the software.

The demo-version of the taraVRcontrol from Tarakos was downloaded and tested. The software was in English, but the help and tutorial were in German and this made it harder to understand. A finished model was included with the software and this has been studied and simulated. This gave some answers to questions in the SRS. Some articles from magazines were found on the Internet, where projects using taraVRcontrol were described. In one project both taraVRcontrol and WinMOD were used[29]. The taraVRcontrol was used for the 3D-visualization and WinMOD was used for the behavior of the model and the real-time simulation.

There was no demo-version available of WinMOD. But at the manufacturer’s homepage there were some reports that describe some projects where WinMOD had been used. It was quite hard to find information about WinMOD and therefore most of the information comes from their homepage. Mewes & Partner pointed out especially that WinMOD is a real-time simulator. When a vendor already was selected WinMOD sent us manuals to some of the functions in the software. They also gave access to downloadable demonstration videos from their server.

---

[29] Tarakos, IEE 11-2004, Fertige Programme für fictive Maschinen (Internet), 2005-05-13

From Delmia there was no demo-version available of DA at the time for the assessment. The reason was that the software was not on the market at that time; it was released on April 12. From our supervisor at Tetra Pak, Mr. Istvan Ulvros, there was access to confidential information about DA. At Delmia’s homepage there were also some press material, both written and videos, available. Eric Retraint, Vice-President Sales and Channel Delmia Automation, gave access to more information and answered the questions of the software’s possibilities.

All of the four candidates can import geometry from Pro/E. Some of them use VRML or other standard formats, like STEP. If Delmia’s CAD solution Catia is used, the geometries in DA can be redefined and the original CAD file will be changed. All of the software can handle control signals from a PLC, both external and internal PLC. All software can handle the signals via OPC except WinMOD that uses Profibus. RSTestStand, Delmia Automation and WinMOD can be connected to an external HMI, while taraVRcontrol can be used in combination with WinMOD to get this possibility. The only software that offers real-time simulation, according to the vendors’ specification, is WinMOD and for the others it depends on the complexity of the simulation and on the available calculation capacity. Real-time simulations are very demanding in computing power. It is always an adjustment how accurate the simulation will be and how much time the simulation will take.

All of the software will run on a PC with MS Windows. Delmia Automation and RSTestStand support IEC 61131-3 but DA today only supports SFC. WinMOD and taraVRcontrol got their own solutions for the control of the behavior. DA and taraVRcontrol offer collision analysis and also the possibility to set sensors at critical positions to only calculate collisions at these places. This is to save calculations and simulation time. It is quite easy to set up sensors, and make use of graphical configuration. WinMOD offers a tool to plot time-sequence results.

Conclusion

The conclusion of the assessment is that Delmia Automation meets the requirements best and has the most complete solution, for these reasons Delmia was the selected vendor. An alternative choice could be WinMOD together with taraVRcontrol. The drawback with this choice is that this solution can give rise to problems. A communication problem can come up between the software modules, since they are not tested together by the vendors, and two new software products must be learnt. If trouble appears the vendors can refer to each other and no one will take responsibility for the problem. The taraVRcontrol documentation is not completely in English, since the help functions are written in German.

The drawback with DA is that this solution is new on the market and has never been tested in real projects. Another drawback is that DA today does not offer a setup tool for PLCs from B&R or for the GE Fanuc 90-30 family, which Tetra Pak R&D normally uses.

The results of the market survey are very depending of the manufacturer of the software. Their answers, if any was received, have influenced the results in a great way. Also other information and presentation material sent and their treatment and establish of contacts has influenced the results. The best evaluation method had been to do a short test run of all the software before the choice was done.

No one of the software fulfill all requirements, not even all requirements of highest degree. The entire SRS and the result of how all the vendors fulfilled it are described in Appendix A.

4.4 Presentation of selected vendors

Delmia is a part of the big French group Dassault Systèmes. Delima has a close cooperation with other companies to develop and sale the software, namely Dextus and Schneider Electric.

Delmia

Delmia was founded in June 2000 as a high-end brand for digital manufacturing[30]. Today Delmia provides solutions to plan, create, validate

[30] MCADCafe, Delmia V5R13 – Detailing and Validating Digital Manufacturing Process (Internet), 2005-04-11

and control manufacturing processes[31]. Examples of products that Delmia offers are Quest, Igrip, Human and Automation[32]. Quest is a 3D digital factory environment for process flow simulation and analysis, accuracy and profitability. Igrip is a robot simulation solution for modeling and off-line programming of multi-device robotic work cells. Human is a solution to create and simulate workers in the digital manufacturing environment and to introduce human factors into the product lifecycle earlier. DA is a next-generation collaborative environment for the automation industry. It allows a control engineer to describe a PLC program in a logic modelling language and validate this against a 3D model of the cell, machine or an entire line[33]. A closer presentation of Delmia Automation will be done in Chapter 5.3. Delmia has 600 employees worldwide and sales and marketing teams in 12 countries[34].

Dassault Systèmes

The company Dassault Systèmes was founded in 1981 and is a global software developer for the CAD/CAM/CAE/PLM market[35]. Dassault Systèmes is the world leader for Product Lifecycle Management (PLM) software solutions using 3D representation. They have six global brands, Catia, Delmia, Enovia, Smarteam, Spatial and SolidWorks. Dassault Systèmes has over 4000 employees in 20 countries[36].

Dextus

Dextus was founded in June 2004 as a subsidiary of Schneider Electric as an outgrowth of a partnership with Dassault Systèmes. Dextus was founded to market and sell Delmia Automation, and Delmia PLM. They will also promote the development of the digital factory. Dextus themselves says, "Dextus is dedicated to production engineering and automation design benefiting from Dassault Systèmes experience in 3D CAD and PLM

[31] Delmia, Our History (Internet), 2005-04-11
[32] Delmia, Automotive (Internet), 2005-04-11
[33] Delmia, Automation (Internet), 2005-04-11
[34] Dextus, Delmia (Internet), 2005-04-12
[35] Dassault Systèmes, Press release (Internet), 2005-04-13
[36] Dextus, Dassault Systèmes (Internet), 2005-04-12

solutions, and Schneider Electric's vision of automation and digital factories introducing a breakthrough in productivity"[37].

Schneider Electric

Schneider was founded by two brothers in France 1836. Their main markets were steel, heavy industry, railroads and shipbuilding. During the 1930s Schneider got problem because of Germany's renewed power. After the World War II Schneider began to focus on civilian needs and were a success during the past 1950s. However, they were in problem again in 1960s and 1970s. Therefore they began in 1981 to rationalize the company by divesting non-strategic or unprofitable businesses. During the last part of the 1980s and during the 1990s Schneider acquired companies to refocus on electricity. In May 1999, Schneider changed name to Schneider Electric and they are today a world class manufacturer of equipment for electrical distribution, automation and control[38]. Today Schneider Electric got over 80 000 employees and is present in 130 countries[39].

---

[37] Dextus, Dextus (Internet), 2005-04-12
[38] Scheider Electric, The year in review (Internet), 2005-04-12
[39] Schneider Electric, Local operation (Internet), 2005-04-12

5 Simulation environment

To perform simulations different kinds of equipment and systems are required, in this master thesis they can be split into three categories. The first one includes standards to program a PLC defined in IEC 61131-3. The second one is software, and in this master thesis Delmia Automation and Pro/Engineer were used. The last category is physical equipment, where a part of a filling machine and a PLC were used. There will be a closer description of these categories in this chapter.

The computer used in this project was equipped with a Pentium III processor at 850 MHz and 256 MB internal memory. In the end of the project a more powerful computer was provided. This could run the simulations faster and with a higher accuracy. This computer was equipped with a Pentium 4 processor at 1.7 GHz and 1 GB internal memory.

5.1 Programmable Logic Controller

Programmable Logic Controller, also known as PLC, is a microcomputer for logical calculations. In this section the PLC will be described, the history and how it works.

History of the PLC

The first PLCs were introduced in the late sixties to reduce the high cost of the controllers being in use. When the production changed, the controller needed to be change too. The controllers then consisted of lots of relays. The relay was expensive to change and they were also mechanical so the lifetime was limited and they needed maintenance. There were also other drawbacks with the relay, and when a complex controller was built it was hard to get an overview of it. Modular Digital Controller (MODICON) was introduced by Bedford Associates to General Motors in 1968, and it could solve lots of these problems. The modular system was soon abandoned because Bedford thought an open architecture was not good[40]. But still it was the MODICON that brought the PLC into a commercial market. With

[40] R. Morley Inc., The History of PLC (Internet), 2005-05-26

this system it was possible to do changes in the program without changing the hardware.

PLCs of today

To be used in industrial automation systems a PLC must operate in real-time. This can be done by two different techniques, polling or with interrupt signals. The central processing unit (CPU) in a computer checks for incoming data over and over, and when it finds data it processes it. This method is called polling since the processor polls the input devices continuously to see if they have any input data to report[41]. The other way is that the input devices send a signal to the CPU when new data has to be processed. This signal is called an interrupt signal and is more effective to use since the processor only has to work when there are data to process.

Polling is easier to program and cheaper but if the PLC is not sufficiently fast it can miss some external event. To use interrupt signals is more difficult to program, and more expensive, but the risk of missing an external event is much smaller. Therefore polling is used in simpler systems and interrupts in more complex systems.

The programming of a PLC can be done in several different languages, but the IEC has managed to merge the languages in one international standard. The latest standard is IEC 61131-3, more about this in Section 5.2. The PLC-code is, in contrast to many other programs, executed in an infinite loop. The *read-execute-write* cycle is called a scan cycle and in a small PLC every full scan may take about 15-30 ms[42].

5.2 IEC 61131-3

The worldwide standard defined by the International Electrotechnical Commission, IEC, for industrial control programming is IEC 61131. The standard was formerly known as 1131. In 1993 it changed name to 61131

---

[41] Internet FAQ Archives, Asynchronous Events: Polling Loops and Interrupts (Internet), 2005-09-09

[42] Olsson, Gustaf – Rosen, Christian (2004); *Industrial Automation*, Ch. 14

when IEC standardized the names of the standards[43]. The standard consist of five parts and part three describes the language itself. It is this part that is used and it is named 61131-3. By implementing this standard into different environments for development, it is easy to change vendor with minimum effort.

The standard consists of five language definitions. The languages are Sequential Function Chart (SFC), Function Block Diagram (FBD), Ladder Diagram (LD), Structured text (ST) and Instruction List (IL). FBD and LD are two graphical languages while ST and IL are two textual languages. SFC has been defined to structure the PLC program and function blocks[44].

Sequential Function Chart

Sequential Function Chart, SFC, is a graphical language and is a good method to organize a program, sense it uses top-down approach. SFC consists of the main components; step, action and transition. Each component consists of a function that is described in one of the other languages.

To a step there are none or more actions connected. A step can either be active or inactive. In any given moment every active step is executed, and its actions will be accomplished. If there is no action connected, the step will act like a wait function.

An action can be represented in many different ways and can be written in any of the languages described below. An action is where the individual task is performed.

There is a transition between every step. To a transition there is a condition connected. The condition is a logical expression and when it is fulfilled the program proceeds to the next step, the new step becomes active and the previous becomes inactive.

---

[43] PLCopen, 1131 or 61131: status of the Standard (Internet), 2005-04-14
[44] Olsson, Gustaf – Rosen, Christian (2004), *Industrial Automation*

There are several ways to execute the code. The first and simplest way is a simple sequence, where there is only one way to execute, see Figure 5.1.

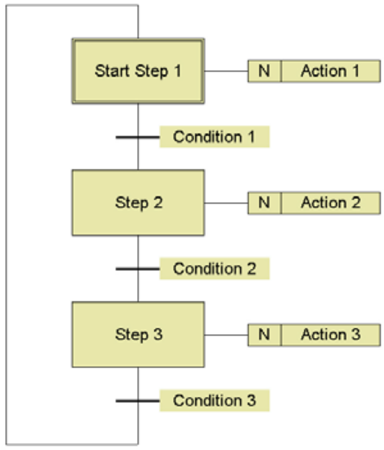


Figure 5.1: A simple sequence in SFC.

The next is alternative branch, showed in Figure 5.2.

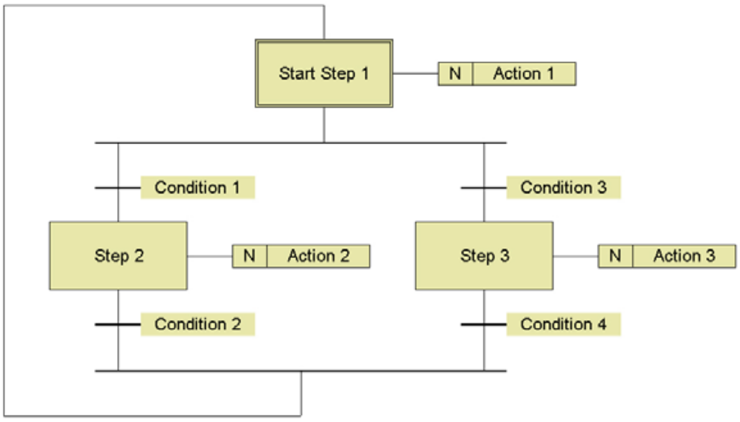


Figure 5.2: An alternative branch in SFC.

After a step in an alternative branch there are two or more transitions and a branch connected to each transition. Here only one of the transitions can be fulfilled, and the program proceeds into that branch.

The last way is simultaneous branch, showed in Figure 5.3.

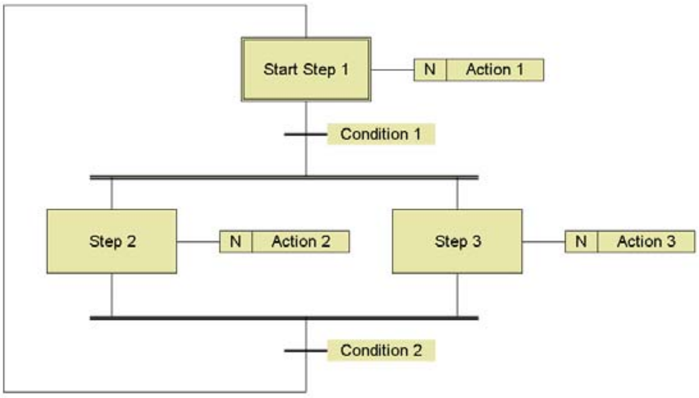


Figure 5.3: A simultaneous branch in SFC.

Here the entry transition will be fulfilled and all branches below will be executed simultaneously. Not until all branches are finished the program will proceed.

Ladder Diagram

Ladder diagram, LD, is the first graphical language in IEC 61131-3. Ladder diagram was the traditional way to program a PLC. The language has a lot of similarities with the relay ladder logic that was used formerly to control industrial systems, but the IEC 61131-3 ladder diagram has been further developed and can use predefined function blocks[45]. The great advantage with ladder is for logical expressions. It is also possible to use it for sequential programs. A simple sequence in ladder can be seen in

[45] EC&M, What to know about PLC ladder diagram programming (Internet), 2005-04-19

Figure 5.4, where hold circuits are used for the current step and stop conditions for the other steps.

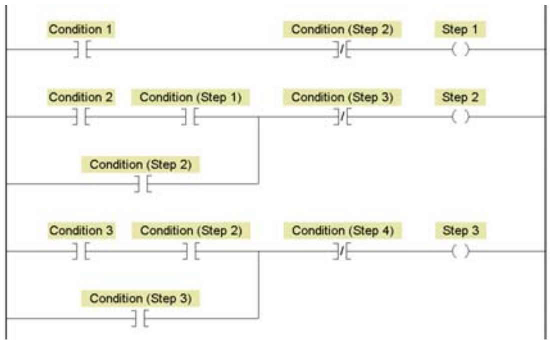


Figure 5.4: A simple sequence in ladder diagram.

Function Block Diagram

Function block diagram, FBD, is the other graphical language in IEC 61131-3. FBD resembles of an integrated circuit, IC, and it has a specific function. The block has a well-defined interface with inputs and outputs, and the logic and algorithms are hidden inside, see Figure 5.5. The FBD can be written in any of the IEC 61131-3 languages but in most cases it is written in C[46]. It is helpful when a complex algorithm will be used over and over again in the same project or even in another project. In this way a FBD can be written by someone else, e.g. vendor, and used by the programmer.

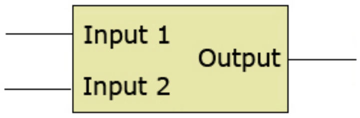


Figure 5.5: A function block diagram.

[46] PLCopen, IEC 61131-3: a standard programming resource (Internet), 2005-04-19

Structured Text

Structured text, ST, is a high-level language; it has its origin in languages like ADA, Pascal and C. It supports most of the standard functions and operations, e.g. if-else, for-loop, while-loop, see Figure 5.6. This permits an easy modeling of nestled loops and complex algorithms, which is useful when creating a FBD. ST is often used for the description of the actions connected to a step in a SFC.

```
WHILE condition1 THEN
    action1
END_WHILE
IF condition2 THEN
    action2
ELSE
    action3
END_IF
```

Figure 5.6: Structured text with a while-loop and if-else.

Instruction List

Instruction list, IL, is a low-level textural language; it has a similarity with assembler. The language is close to the machine instruction code, and therefore it is the most efficient language in IEC 61131-3. A simple sequence with the same function as in Figure 5.4 can be seen in Figure 5.7. Every IEC 61131-3 language can be converted into IL.

```
ld      condition1
ani     condition(step2)
out     step1
ld      condition2
and(    condition(step1)
or(     condition(step2)
ani     condition(step3)
)
)
out     step2
ld      condition3
and(    condition(step2)
or(     condition(step3)
ani     condition(step4)
)
)
out     step3
```

Figure 5.7: A simple sequence in IL.

5.3 Delmia Automation

Delmia Automation is a tool for the development process in industrial automation. It gives ability to do everything from implementing a PLC-code to simulating the mechanics and verify the PLC-code against the virtual machine. Delmia Automation is a software solution that consists of many different modules and it is scalable so that each module can be used independently. In DA it is possible to implement the PLC-code and do simulations on the virtual machine before the real machine is built up. In this way it is possible to verify that both the code and the mechanics behave in a desirable way[47]. The module that was used in this project was Device Builder, HMI Control Panel Design, LCM Studio, Smart Device Builder and Controlled System Simulator. The version used in this project was Delmia Automation V5R15 and during the project service pack 1, 2, and 3 was received.

[47] Dassault Systèmes, Delmia V5 Automation: Fact Sheet R15 (Internet), 2005-05-11

Device Builder

Device Builder is the module where the functional mechanical objects are built and configured. Properties configured are i.e. the geometry, kinematics and behavior. It is possible to import a model from a 3D CAD system. To use geometry from a CAD-system a neutral file format, i.e. STEP and VRML, can be used. The model can be turned into actuators and sensors so it can be controlled by a PLC. The kinematics can be reused from the CAD-model but it can also be added afterwards. The possibility to import the kinematics from the CAD-model is today only available if the CAD-model is done in Dassault’s CAD-software Catia.

It is in Device Builder that mechanisms are created and every mechanism can consist of many different joints. A joint is a type of motion that a part can perform. Example of types of joints are revolute, prismatic, cylindrical, planar and rigid. Dependent to the type of joint it has different degrees of freedom (DOF) which can be translational and/or rotational. A rigid joint has no DOF and lock a part to another part. To move a part with a mechanism predefined home positions are used. Several motions can be defined in tasklists, and the time for the motion can be defined. It is also possible to define travel limits for the different joints. Figure 5.8 shows the dialog box to define home positions. The pulled-out icons from the toolbar in this figure show different types of joints that can be created in the model. Above these icons is the tool to create frames of interest. This is references used in models imported in neutral file format, i.e STEP and VRML. These references must be used to create mechanisms in files that not are in Catia format.

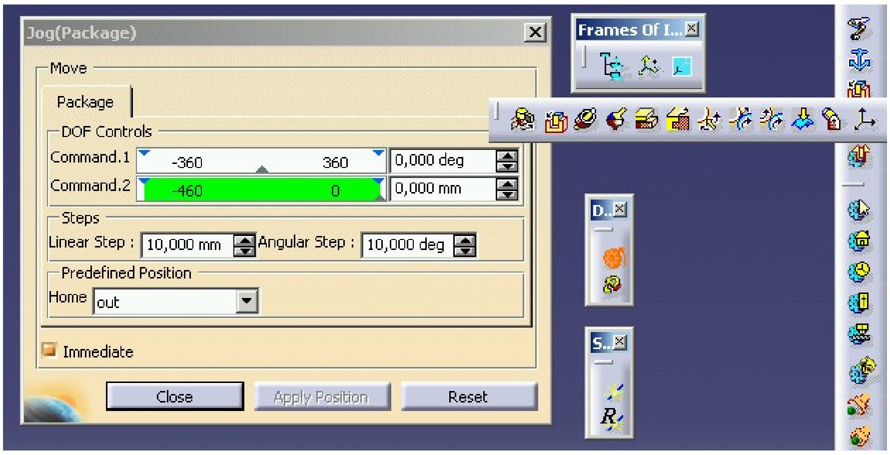


Figure 5.8: The dialog box to define home positions. The pulled-out icons from the toolbar show different types of joints.

Sensors

Sensors, a specific class of object in the model, are created in Device Builder and the sensor's geometry can be loaded as a CAD-file. The behavior is defined as a continuous signal, a rising edge or a falling edge. The difference between them is illustrated in Figure 5.9.

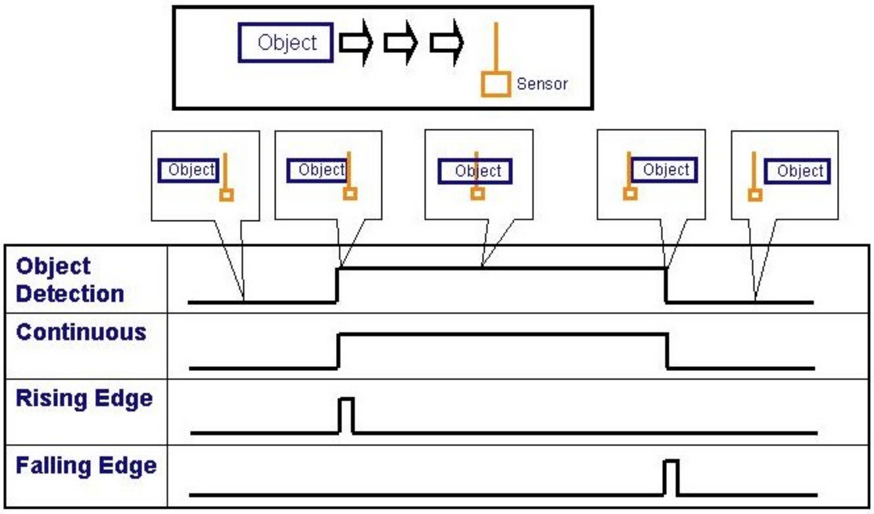


Figure 5.9: The behavior of the sensor is of three different modes[48].

HMI Control Panel Design

HMI Control Panel Design is a product to create virtual human-machine interfaces. Control panels can be used in Controlled System Simulator for interactive use in debugging, validation and simulation. There are several predefined controllers, such as buttons, switches, lights and gauges. But there are no tools to create lines or simple figures, as a rectangle. The tools to design the HMI are showed in Figure 5.10.

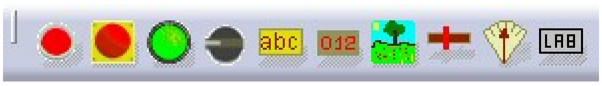


Figure 5.10: The tools those are available to design the virtual HMI.

After designing the HMI the signals will be set. A proposal of signals is auto-generated and it is easy to change them, i.e. the type of signal from

[48] Delfoi Finland, Delmia Automation (Internet), 2005-07-20

Boolean to integer. An overview of the signal management of an HMI is displayed in Figure 5.11.

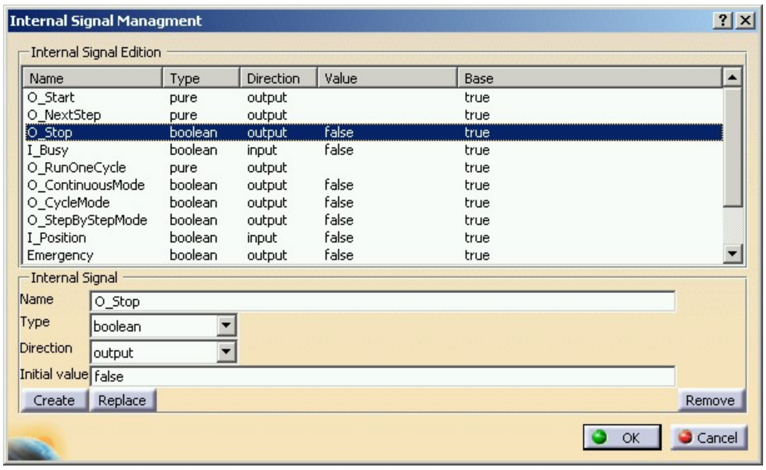


Figure 5.11: The internal signal management tool in the HMI module, where the type of signals and initial value is defined.

LCM Studio

LCM Studio is an environment for implementation of PLC programs. It uses the languages defined in the standard IEC 61131-3, described in Chapter 5.2. In DA V5R15 the only available language is SFC, while the rest of the IEC 61131-3 languages will be added in the following releases of DA[49]. LCM Studio has the possibilities to debug the PLC-code without being connected to an environment, as a real machine. In the debugger the code can be run continuously or cycle by cycle. All the signals are set manually, which makes it possible to test code blocks in many different ways. All the I/O that are set can be saved in a text file to be used for example to understand and correct bugs. In this text file all I/O are saved every cycle of the PLC execution. In the upcoming version, R16, the debugger will work with the 3D-simulation. Then movements of

[49] Phone conference with Eric Retraint, vice president sales and channels Delmia Automation, 2005-05-09

mechanisms can also be analyzed in the debugger. Today these signals are set manually, see also case 47 in Appendix D. In the Smart Device Builder it is possible to create test logic to verify the code together with the virtual machine. See more about this in the section below.

The PLC programs can be built up in a hierarchical model, and then macros will be used. A macro is connected to a step and consists of a SFC that will be executed when the step is active.

There are some differences between IEC 61131-3's SFC and LCM Studio's SFC. The biggest difference is in creation of a simultaneous branch. To create a simultaneous branch macros will be used and a special type of transition, join transition. Join transition works in a way so all macro must be in their last step before the transition can be fired. A screenshot from LCM studio with macros and join transitions is displayed in Figure 5.12.

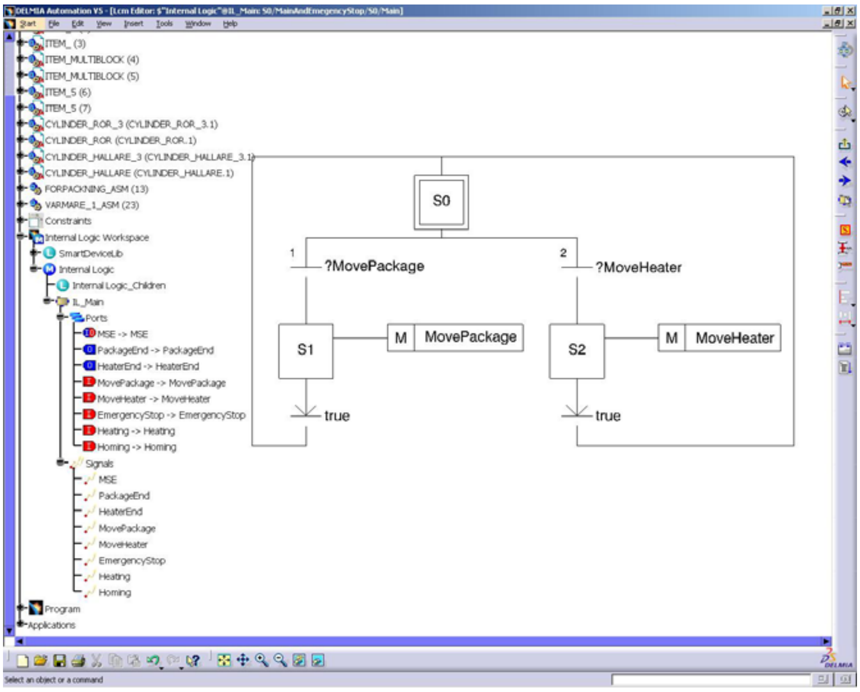


Figure 5.12: A screenshot from LCM Studio. Note that join-transitions are used after S1 and S2.

Smart Device Builder

Smart Device Builder is the module where the behavior for a mechanism is defined. The behavior is built up by using any languages supported by LCM Studio. The behavior uses the predefined tasks done by Device Builder to do the motion for the internal logic.

To make a device controllable it needs to be connected to the environment. This will be done in Smart Device Builder by creating an interface of electrical I/O-signals. The I/O-signals are called ports in DA and a port will be mapped to a signal, which are used in the internal logic. This is displayed in Figure 5.13.

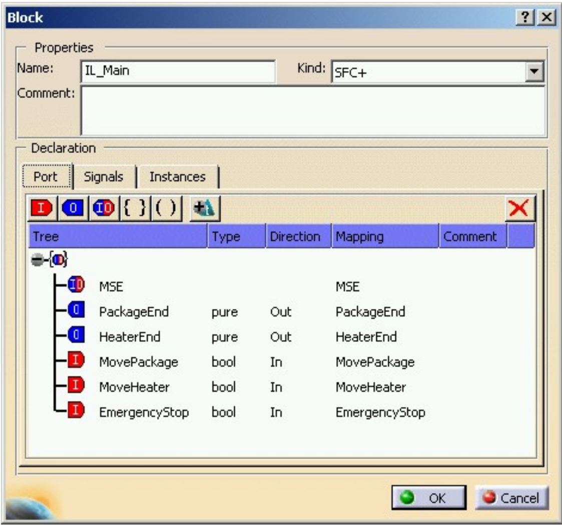


Figure 5.13: Ports are defined in Smart Device Builder, to use them in the internal code they must be mapped to a signal.

In Smart Device Builder it is possible to simulate the internal behavior, to see if the behavior is as expected before it is connected to the external equipment. Without a connection to the environment, the external signals need to be set manually. A screenshot from Smart Device Builder is displayed in Figure 5.14.

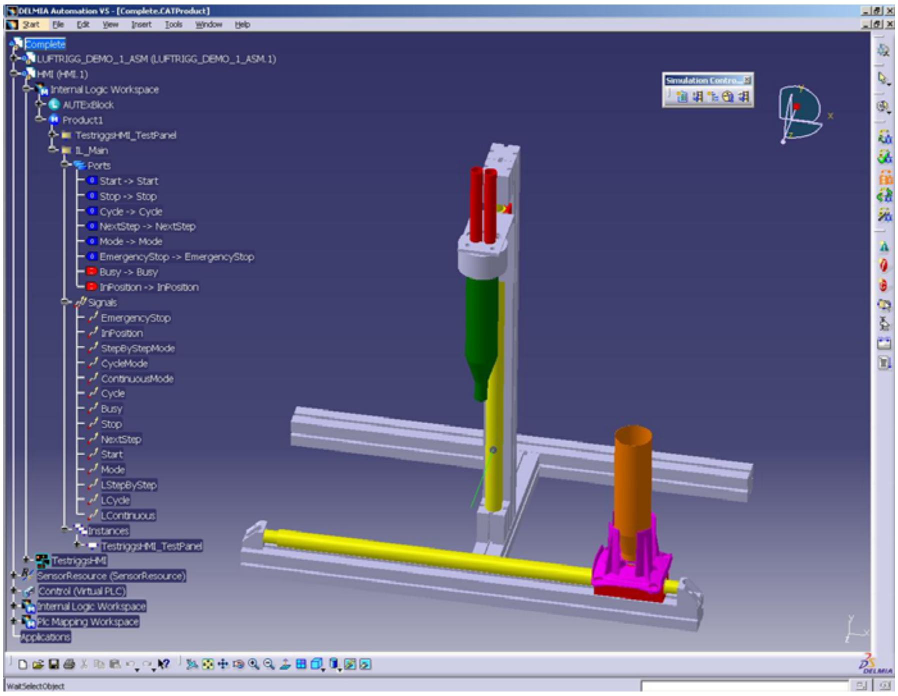


Figure 5.14: A screenshot from Smart Device Builder.

In the Smart Device Builder it is possible to create test logic for a testing scenario. This is useful to automate the simulation of a device and the signals, i.e. from an HMI, can be set in the code. The test logic is created in SFC and the 3D-simulation is connected, but the code is not shown during the simulation. See also case 37 in Appendix D for limitations in the function to create test scenarios.

Controlled System Simulator

Controlled System Simulator acts like a link between the LCM Studio and the Smart Device Builder. The simulations are taking place in this module. It has the opportunity to debug PLC-code and validate the PLC-code against the virtual machine. It is not necessary to only use the virtual PLC, but it is also possible to use a real PLC. If a real PLC is used the PLC Setup module must be used to map the signals.

Controlled System Simulator can do collision analysis to detect if parts in the machine will crash into each other. Some other types of analyses that can be done are distance analysis and sweep of 3D geometry. Sweep is done to create a volume from all positions a machine can have during a run and can be used to determine a working space for a machine.

The module is not only suitable for doing simulations to validate the PLC-code but is also available to connect an HMI. With a full HMI it will be a good machine operator-training simulator, see Figure 5.15 for the working environment during a simulation.

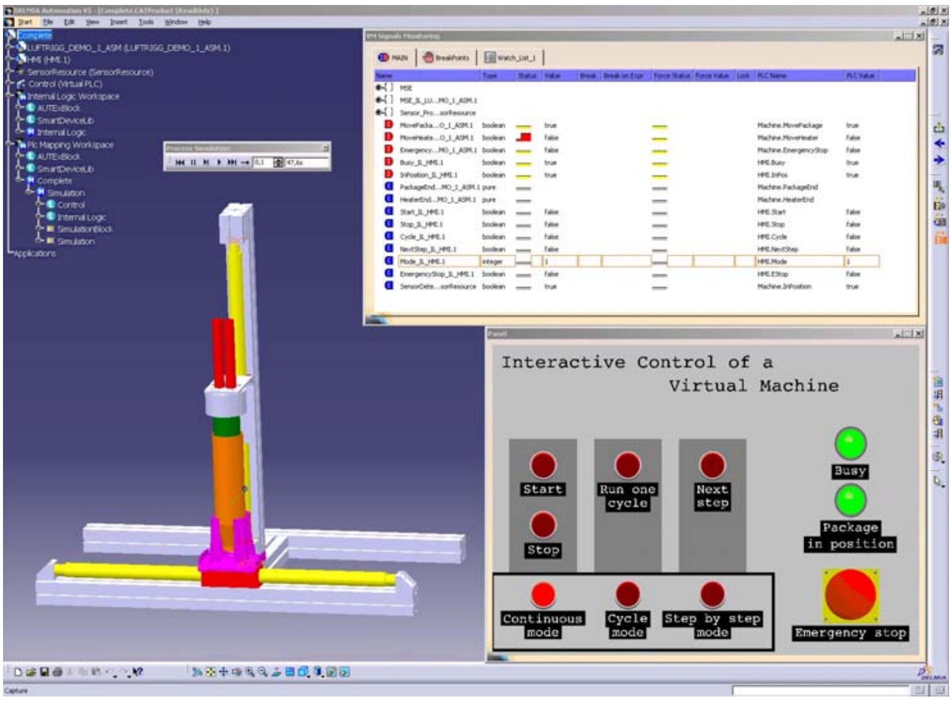


Figure 5.15: A screenshot from Controlled System Simulator. The signals used and their value are seen in the upper right corner and the HMI in the lower right corner.

PLC Setup

To compile the PLC-code to the PLC native code and transfer it to the real PLC the module PLC Setup is used. See example in Figure 5.16 to choose target PLC. The PLC Setup module must be compatible with the PLC provider/type and today the PLC Setup is only available with PLCs from Schneider Electric, Omron and Siemens. This module was not used in this project.

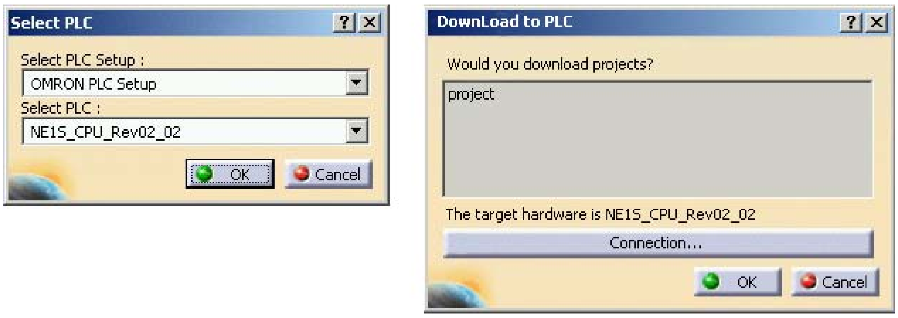


Figure 5.16: A Screenshot from PLC Setup.

5.4 Pro/Engineer

Pro/Engineer is a software for 3D-Computer Aided Design (CAD), and it was made by the company Parametric Technology Corporation (PTC). The company was founded in 1985 and the first version of Pro/E was launched in 1988. Pro/E consists of several modules for different types of tasks and the modules exist in different standards dependent of how advanced possibilities that are needed. There are modules to buy to make NC-code to manufacture the parts that have been designed, and also modules to analyze the parts with the Finite Element Method. These modules enable to get the full CAD/CAM/CAE function. CAM is an abbreviation for Computer Aided Manufacturing and CAE is an abbreviation for Computer Aided Engineering. Today PTC also offers a solution for Product Lifecycle Management (PLM).

With Pro/E new parts and machines can be designed and developed. Each part will be designed separately and then put together in an assembly. Then the kinematics can be added to the model to make the model moving, in 3D. The results can also be saved as a movie or rendered pictures. The properties and dimensions of a part can easily be changed and automatically updated in all other modules in Pro/E that use this part, i.e. assemblies and drawings. This is because Pro/E is fully associative. Several different analyses can be done in Pro/E, for example if the parts intersect when they move and the forces in different cases of load. A great disadvantage is that the associative property is only valid in PTC's products. If the model is used in software from another manufacturer it is necessary to go back to Pro/E in order to change some dimensions and then manually import or update the model again in the software. [50, 51]

The original geometry used in this project was created in Pro/Engineer 2001, but some changes were needed to do and then Pro/Engineer Wildfire 2 was used.

5.5 Simulation object

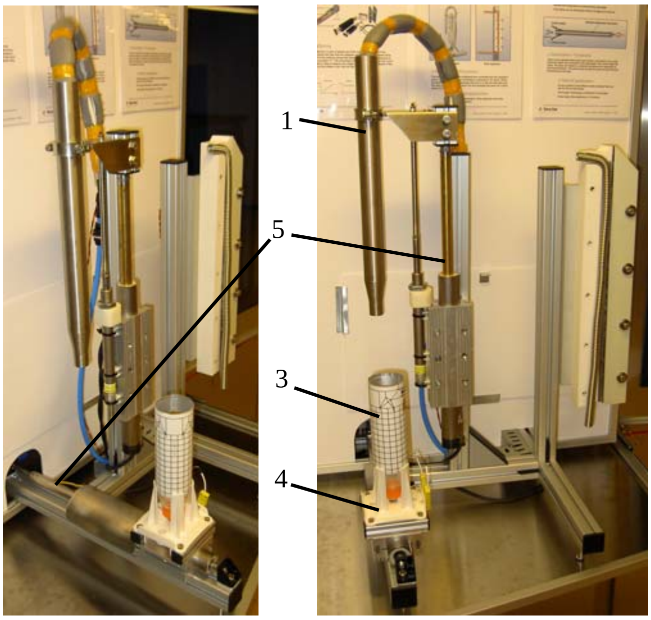
The object that has been simulated in this thesis is a part of a new filling machine developed at Tetra Pak. This part demonstrates an idea of how to warm the package before it will be sterilized in a later step in the machine. This idea is realized in a real prototype machine. This machine has sliders that can move the package, and also a heating mechanism to heat the package. With this real machine the simulation can be evaluated and compared to. The object is shown Figure 5.17.

The idea of this machine is that the package is positioned upside down in a holder and this holder moves along a slider. When the package is in the proper position a pipe will be moved down into the package and heat the inside of the package using hot air. The pipe will then go up and the package will move out. The model is completed with sensors that indicate when the package is in position for the pipe to go down.

The CAD model replicates the real object to highest possible extent, which is a necessary requirement if the result of the simulation will be applied to

[50] PTC, The Product Development Company (Internet), 2005-05-04
[51] PTC, Pro/Engineer Wildfire (Internet), 2005-05-04

the real machine. The idea to simulate the machine in Delmia Automation is that the verified PLC-code can be used in the real PLC. Furthermore virtual sensors can be used in the simulations to do an even more realistic simulation.



1. Heater
2. Sensor (not visible)
3. Package
4. Package holder
5. Slider

Figure 5.17: The simulation object in this master thesis.

6 The simulation project

In this chapter the simulation project is described in detail. How the different parts of the virtual machine were created is described and also how they are connected to each other to a complete model. The complete virtual machine consists of three parts, a mechanical model, an HMI and a virtual PLC.

6.1 Creation of the mechanical model

The mechanical model was created in Device Builder and Smart Device Builder. The geometry was already made at Tetra Pak and it was created in Pro/Engineer. There were only minor adjustments in the assembly that were required to do. The first step was to import the geometry to DA. From Pro/E the geometry was exported in a neutral file format that DA was able to import. Eventually, depending on problems with DA, a bug, add-on modules and license problems the geometry could not be imported. The support at Delmia in France therefore converted our model to Catia-format that was no problem to import in DA. Later in the project the import problem was solved, see more in case 3.1, 3.2 and 40 in Appendix D. Then the mechanisms were added to the model. The mechanisms were defined as for the real machine, with travel limits and time for the motions. The model consists of two mechanisms, one for moving the package and one for moving the heater. They were then used, together with the mechanisms' home positions, to create task lists for the different moves that the machine can do.

In Device Builder a sensor was added to the model. The sensor will detect if there is a package or not in the package holder. If there is not a package present the heater will not go down and the package holder will return to its initial position. The finished model of the machine is depicted in Figure 6.1.

The next step was to create the internal behavior of the model. The internal behavior is how the model will react to different signals that are sent to it. Example of signals can be MovePackage=true or EmergencyStop=true. It is in the internal behavior the mechanisms are activated, thru TaskLists. This was described in Section 5.3. The internal behavior of the sensor must also be defined. In this project a continuous signal was requested and the sensor was defined in that way. The internal behavior of the machine and the sensor was defined in SFC in the LCM Studio. The signals that the internal logic receives and sends was created and they are listed in Appendix B.

The internal logic is in the normal behavior able to receive signals from the control logic and dependent of the signals, the machine will react on it. The logic will react on the signal MovePackage and MoveHeater. Depending on if they are true or false the package and heater will go in or out and up or down. When a movement is finished the internal logic will set a signal to tell the PLC. If the signal EmergencyStop is set, all movements are aborted immediately and the machine cannot be operated until the emergency signal is released. The complete code of the virtual machines' internal logic is in Appendix C.

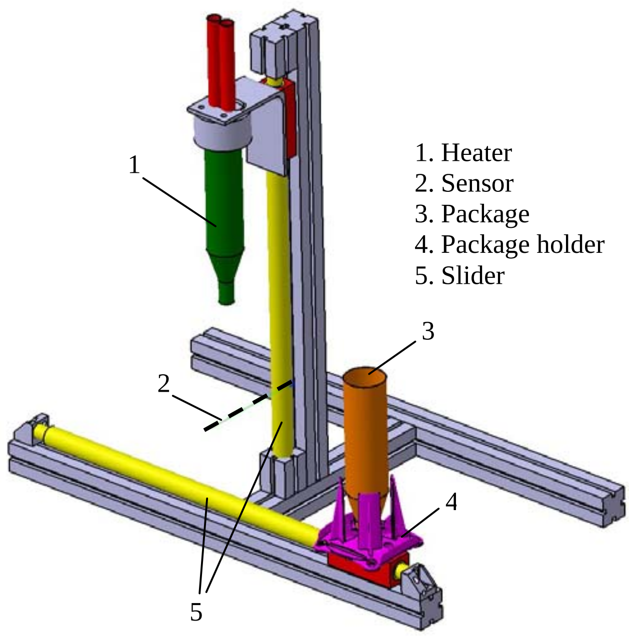


Figure 6.1: The mechanical model of the virtual machine.

6.2 Creation of the HMI

A virtual HMI was created to operate the virtual machine. The HMI was done in two modules, HMI Control Panel Design and Smart Device Builder. At first the design was made. The machine has to be controlled in three different modes so the HMI was designed to handle these three modes. They are Continuous mode, Cycle mode and Step by step mode. To indicate the buttons that belong to the different modes, a darker background was made to group the buttons. Two lights were created to give feedback to the operator, one indicating if the machine is busy and the other indicating if the package is in position for heating. Finally an emergency stop was placed at the panel. This due to the security, it will stop the machine immediately. This is different from the stop button at the continuous mode, which will let the whole cycle run before the machine is stopped. The design of the HMI is in Figure 6.2.

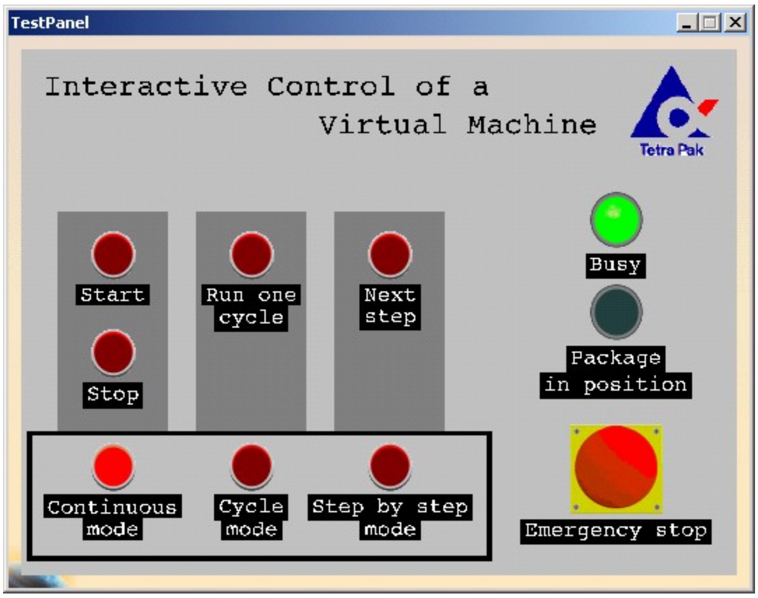


Figure 6.2: The virtual HMI to operate the virtual machine. Note that the light Busy and the button for Continuous mode are activated.

The HMI has an internal logic and this was created in the Smart Device Builder. The internal logic of the HMI controls which of the buttons that will be lighted and set the actual mode. If the emergency button is activated, all the lights will be switched off. The complete code of the HMI’s internal logic is in Appendix C. Some signals are only used internally in the HMI. They are used to light the mode buttons to indicate in which mode the machine is operating. The signals that the HMI uses and can be used by other modules are listed in Appendix B.

6.3 Creation of the virtual PLC

A virtual PLC was created to handle the HMI operation signals and to control the virtual machine. The virtual PLC was done in the LCM Studio and written in SFC.

The program was built hierarchically to get a good supervision of the function’s urgency. The first layer takes care of an emergency stop and stops the machine immediately. Next layer is a reset layer. This runs an initialization loop for the machine and arranges the heater in an upper position and the package holder at the end of the cylinder, see the first step in Figure 6.6 for the start position. Every time the mode is changed the reset is executed.

The last layer is the main layer, which controls the machine in normal operation. The main macro has three alternative branches, one for each mode, see Figure 6.3. Most parts of the behavior for the Continuous and Cycle modes are similar and therefore parts of the code are reused for both modes. When one of these modes is selected the PLC will wait for the signal Start or Cycle, depending of selected mode, see Appendix B for the complete list of signals. After the signal is emitted the cycle starts, beginning with moving the package holder, see part 4 in Figure 6.1, to a position under the heater, see part 1 in Figure 6.1. Then the sensor, see part 2 in Figure 6.1, will detect if there is a package, see part 3 in Figure 6.1, in the package holder. If a package is detected the heater starts moving down and at the same time starts heating. When it has reached the lower position it will move back to its upper position immediately and in the upper position the heating stops. After this the package holder will move back to its original position and a cycle is finished. Here is one difference between the two modes. In the Cycle mode it returns to the beginning of the cycle and waits for the signal Cycle to be emitted. In Continuous mode a new cycle will start immediately, and after the Stop signal is emitted the current cycle will be completed. The program returns to start and waits for the Start signal.

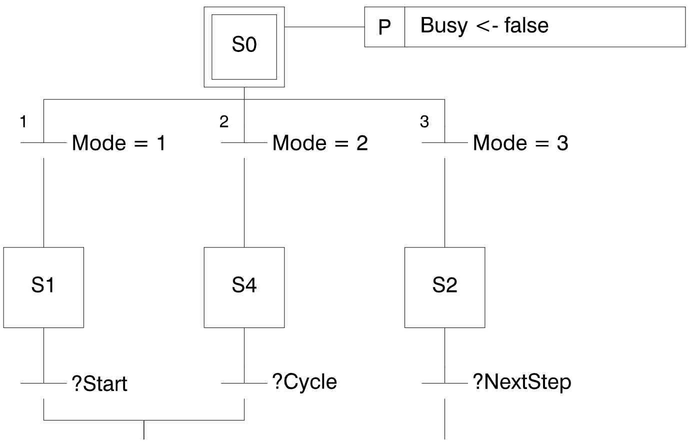


Figure 6.3: The start sequence for the main layer in the virtual PLC.

In Step-by-step mode the signal NextStep needs to be emitted to start the cycle. In the first step the package holder is moving to a position under the heater. NextStep needs to be set again to start moving the heater and start heating if the sensor detects a package, else the package holder moves back to the starting point. When the heater is in the lower position the heating is stopped. This is done to not harm the package while waiting for next movement. Following time NextStep is emitted, the heating will start again and the heater moves to the upper position. NextStep needs to be set a last time to complete the cycle and the package holder is moved to its origin. Now the program is ready to start a new cycle.

6.4 Connect the parts to a complete machine

The last step in creating a complete model is to connect the created parts to each other. In this project there were three different parts created, a mechanical model, a virtual HMI and a virtual PLC. The virtual PLC will control the other parts. To create the complete virtual machine starts with inserting the created parts in a new product, in this project the machine and the HMI. The internal logic is already defined in each product and with a wizard the internal logic is copied from each product to the new product.

Then the control logic will be added to the model, the virtual PLC. After this the internal logic and the control logic will be connected to each other. This is displayed in Figure 6.4. The ports in the control block are the ports of the PLC and the ports in the simulation block are all the ports of the different products in the model, in this project the mechanical model and the HMI. Giving the ports in the products and the PLC good names will make it easier to connect correct signals to each other.

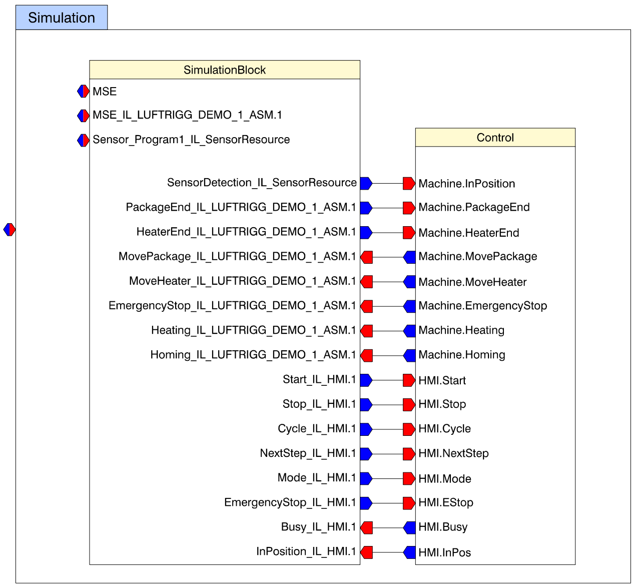


Figure 6.4: The connection between the internal logic and the control logic in a complete machine.

6.5 Running a simulation

When the complete machine is finished and all signals are connected in a correct way it is time to run a simulation. During the simulation all the signals between the PLC and the simulation object can be watched, and they can also be forced to a value to test how the simulation will react on it.

During the simulation the step size of the simulation is set, it can be set manually or the simulation can be in real-time. In real time mode the step size will be configured automatically so the calculations for the simulation are caught. Often the graphical update will slow down a simulation. To speed up the simulation the graphical update can be set to not update the graphic each simulation cycle. See example in Figure 6.5.

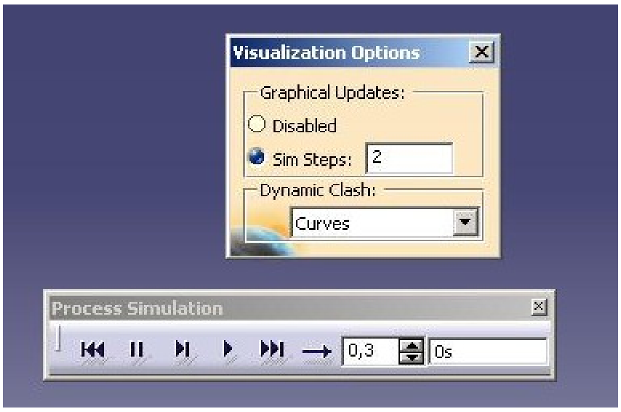


Figure 6.5: Visualization Options for a simulation. In this example, with the step size set to 0.3 seconds, is the graphic updated every second step.

When running a simulation a clash control can be done. If this option is activated the parts that clash to each other are highlighted during the simulation. Another choice is to let the simulation stop when a clash is indicated.

The complete virtual machine in this master thesis is controlled by the virtual PLC via the virtual HMI. In Figure 6.6 one cycle of a run is displayed.

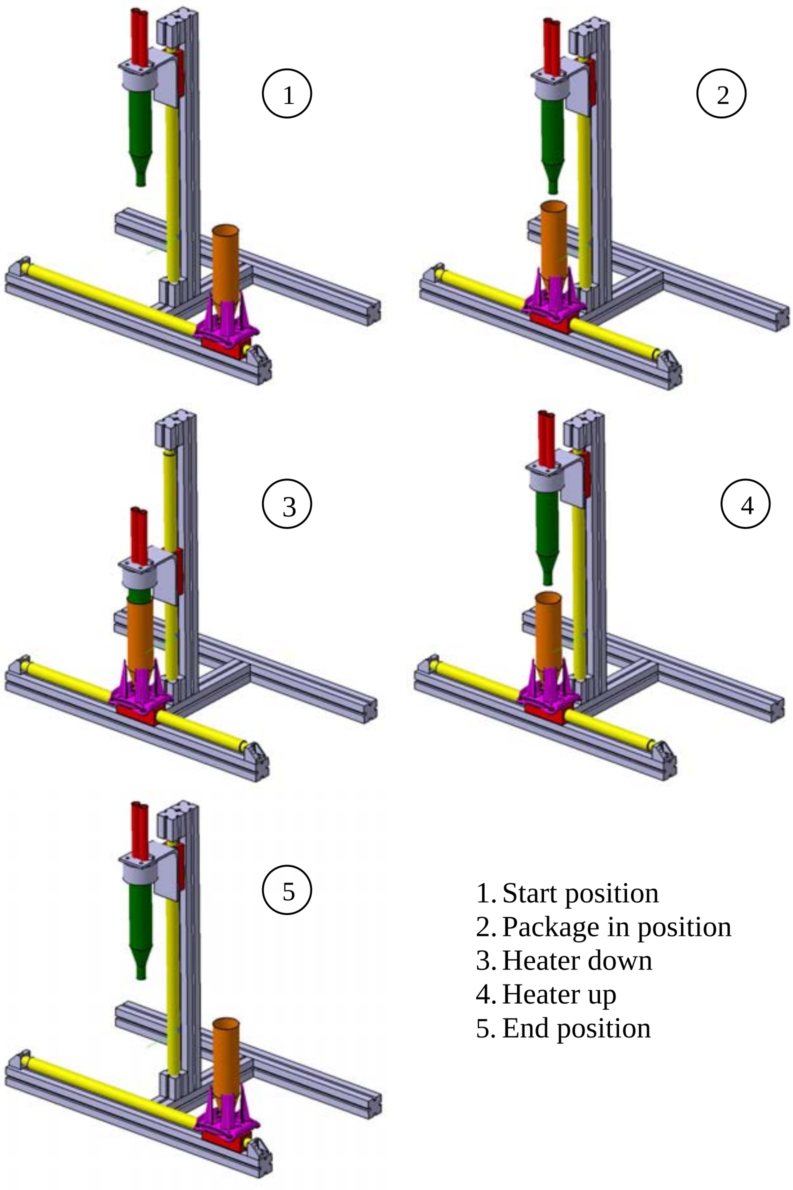


Figure 6.6: One cycle of the machine. The different steps are numbered from one to five.

6.6 Reflections of the simulation project

Delmia Automation is a new software on the market, and during the creation of the virtual machine many more or less serious bugs and problems have appeared. There are also some usable functions that are not yet implemented. During the project a log has been created with our problems and observations, and also the answer from the support at Delmia Automation. This log is in Appendix D.

A great disadvantage is the compatibility with import of models from Pro/E. Only geometry can be imported, no kinematics. The mechanisms must be added in a special way that includes making special references. As described in an earlier section DA is from the same company group that manufactures Catia. If a model is created in Catia there is full compatibility with DA, including kinematics and updates of the geometry.

In the start of the project the system requirements were not distinctly defined. This gave the results that the computers used during the project were not so powerful. In the end of the project a specified system requirements list were available and then a more powerful computer was provided. This lead to a greatly improved performance and some strange errors in the software did not appear, i.e. graphical disturbances and exception errors.

The delay in the beginning of the project, when waiting for software from the selected manufacturer, caused a delay in the project that lead to that all steps in creating a simulation were not evaluated. The steps not evaluated were controlling a virtual machine with a real PLC and controlling a real machine with a real PLC, with the same code used in simulations with virtual machine and virtual PLC. Another contributory factor to that all steps were not evaluated was the lack of documentation of the software and all problems with the current version of the software. See more details in the log in Appendix D.

7 Conclusions

In this section the conclusions from the master thesis are presented. First is a summary of the results and then is a recommendation to pass on, based on our experience. Finally some proposals for future work are presented.

The main conclusion is that simulation of machines behavior together with the PLC code of its control system is very promising. This can save a lot of time and a lot of money and be a carrier of better design methodology.

A concern is that this type of simulations is very demanding on the computer's capacity.

The concept Delmia Automation is correct in principles and in ease of use for the programmer. The user use different modules for different tasks and the interface are common, or similar, in the different modules. It is the same suite used for the whole simulation project, from importing CAD-geometry to test the code on the virtual machine.

However the current version of Delmia Automation, V5R15, did not include some fundamental requirements in programming of the virtual machine, e.g. motion profiles see cases 26 and 27 in Appendix D. But maybe they will be implemented in the upcoming versions.

In the current version of DA there is no support for PLCs from manufacturers commonly used at Tetra Pak R&D, B&R and GE Fanuc.

A progress has been noticed to the development of the simulation-tools as well as an increasing interest for this kind of simulation during the six months of this master thesis. It has started postgraduate studies within this area at University of Trollhättan/Uddevalla[52] in co-operation with the KK-stiftelsen[53]. They have been interested in this master thesis to see the development potential and use it as inspiration for further works[54].

[52] University of Trollhättan/Uddevalla (Internet), 2005-09-09
[53] Stiftelsen för Kunskap och Kompetensutveckling (Internet), 2005-09-09
[54] Meeting with Henrik Carlsson, Fredrik Danielsson, Bo Svensson from HTU, 2005-09-28

7.1 Summary of results

A virtual machine has been controlled by a virtual PLC. A conclusion is that it is possible to control a virtual machine with PLC code. It will react as the real machine would have done with the code written in the simulation program.

In the market survey several different software products are included but many of them are focusing on one part in a simulation. Delmia Automation was the software that offered the most from the different parts to simulate and control a virtual machine. Including CAD-import, use of PLC-language defined in IEC 61131-3 and export of the finished, verified, code. Many of the other software products use their own developed programming language, have poor support for import of CAD-files or were directed for a specified type of industry. Delmia Automation was, at the moment for the market survey, the most all-round software that fulfilled the most of the requirements of the software included in the survey.

Even after the vendor was selected and more information from the vendors was received and more understanding of principles for simulation was experienced, the choice of Delmia Automation as the best alternative for further evaluation was correct.

During the project to create the virtual machine many problems, bugs and limitations in the software DA were discovered. To handle these a log was created, Appendix D. This log was sent to the manufacturer of the software several times during the project to inform them of the project. Delmia have taken the comments into consideration and have reacted to them. Their answers and comments were also recorded in the log.

A great disadvantage was the limited support for importing CAD-geometry from other CAD-systems than Catia. The geometry can be imported in a neutral file-format but no kinematics will be imported. It is impossible to update the geometry, it must be re-imported and all kinematics and logics must be added once again.

Hiding and showing parts in a simulation is not possible to control from the code. This feature is very useful to visualize a simulation, and also in debugging of the code and logic. There is no support for changing color of parts of a machine from the code. This can be used in the same way of hiding/showing parts.

There are no included help-files in the DA. This has lead to many contacts with the support team at the manufacturer, and much time has been spent on this instead of creating and developing the virtual model.

There are no tools to do sequence diagrams of signals and motions. There is a tool to record every I/O in every cycle at the PLC, and this is saved into a text-file.

It is not possible to create a motion profile for a mechanism's motion. When a mechanism is set to move, it will go to a predefined position in a predefined time with the speed needed. The speed is like a square pulse with abrupt start and stop.

To run a simulation is very demanding on the computer hardware, especially on the internal memory. Based on tests on different computers 1 gigabyte, or more, is preferred to run the virtual machine created.

It is possible to use geometry from a CAD-system, not only manufactured inside the Dassault group, and during a simulation the user will get visual feedback of the simulation. A very useful function is to create videos of a simulation. They can be used to visualize and present the results for a non-technician or users of alternative systems.

It is very positive that DA uses a standard programming language and not an own developed language. This gives a shorter learning time for the user. But Delmia had done some minor changes in their implementation of SFC. For example it is not possible to have parallel branches. Instead several macros in one step must be used.

It is possible to create virtual HMIs and control a simulation by it. Even if the editor to create the HMI is very simple and quite limited it is usable in example demonstration purposes. If a more complex machine is created, a simulation with a virtual HMI can be used to train the operators of the real machine.

The step-size of the virtual PLC can be set automatically to get a simulation in real-time. It can also be set manually to get a more or less detailed simulation. A great disadvantage with great step-size is if sensors are used in the model. They can be hard to check and debug if a part passes a sensor position during a step in the PLC, due to the sensor will not set a signal.

During a simulation it is possible to turn the collision check on or off. If a collision between some parts in the model is detected the parts intersecting each other are highlighted, or the simulation is stopped.

Delmia and Dassault Systèmes are large companies with great resources. They can invest much money to the software and have a low risk to be liquidated. Therefore it is most probably that this program will not disappear from the market in the near future, instead there will be continued development of it.

The virtual machine has been compared to a real machine by comparing a video sequence from both machines. The machines had separately control systems and were not connected to each other dependent to incompatibly. These systems had different support for motion control and therefore the real machine used more complex motion profiles. The result shows that both machines have the same behavior, except for the differences in motion profiles.

7.2 Recommendations

As a result of this master thesis there are some recommendations for Tetra Pak. This project is a part of a pre-study if it is interesting for Tetra Pak to implement a simulation-tool to the development process and how to proceed further in the subject.

Recommendations in brief

- Continue the monitoring of development of the Delmia Automation.
- In a doable time make a new pre-study to check the maturity and functionality of the commercial products versus Tetra Pak´s requirements. Recommendations for further decisions.

Comments

There are some major requirements that Delmia Automation today not fulfilled and for that reason it can be hard to entirely implement it in Tetra Pak's development process at this stage. One of the most important requirements was to have the availability to import CAD-geometry from Pro/E and update the geometry in an easy way. Today it is possible to import the CAD-geometry, but still it needs lots of work to make the geometry usable in DA. There is no easy way to update the geometry without the need to redefine the kinematics. There is one possibility to have a good CAD import and it is to use Dassault's own CAD-system Catia, but Tetra Pak uses Pro/Engineer and has no plan to change and therefore this is not an option.

Tetra Pak uses many different ways to control a motion, for one example see Figure 7.1. In the current release of DA it is only possible to control a motion by moving between predefined positions and changing the time it takes. This is not sufficiently good to fulfill Tetra Pak's requirements of controlling a motion.

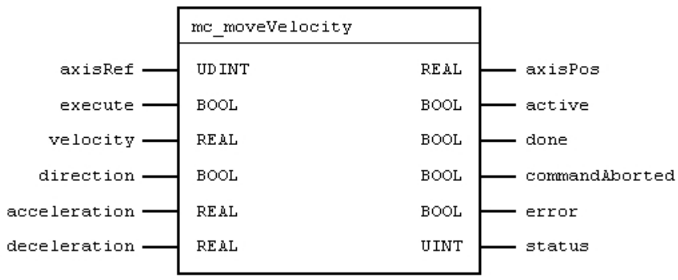


Figure 7.1: A function block diagram Tetra Pak use to control a motion by velocity[55].

In the current release of DA there are support for Schneider Electric, Omron and Siemens' PLC-systems' and Tetra Pak R&D uses B&R's PLC-system, but there are other companies within the Tetra Pak group using Siemens' system. Therefore there are limited resources to use DA completely in a development process. Another common PLC-system used

[55] Documentation to the mc_lib for B&Rs programming tool for their PLCs.

at Tetra Pak is from GE-Fanuc, and today there are no supports for these systems in DA.

Next recommendation is to do further works in this area, read more about the particular details in Chapter 7.3, because this will probably be one of the fastest and cheapest methods to work with.

7.3 Future works

There are several subjects concerning controlling virtual machines by a PLC that are not dealt with. Therefore a recommendation is to do future works within the area in a couple of years. Some interesting aspects on future works are listed below.

A complete evaluation of DA where the last step in a complete simulation, control of the virtual and real model with a real PLC, should be done.

There have been software products with great potential that have not been investigated. WinMOD and MSC.visualNastran 4D should be explored further. It is also interesting to check if PTC is introducing some simulation software for PLC-code, and in that case integrate them in Pro/E. When Delmia has corrected problems discovered during this project and added demanded features it is recommended to make a new evaluation of DA.

The SRS is for a basic simulation tool or a little more complex tool. For further works it is recommended to develop the SRS. A couple of new requirements that can be added, based on the experience from this project, are listed below.

- Better CAD-support where it is possible to update the geometry from Tetra Pak's CAD-system.
- Support for Tetra Pak's PLC-systems.
- More developed and advanced control of motion, where it is possible to control velocity, acceleration and use motion profile.

Another interest aspect that is not connected to the technical subject is a working methodology subject. Investigate how a simulation tool, such as Delmia Automation, can be implemented in the development process at Tetra Pak.

References

Text books
Björklund, Maria – Paulsson, Ulf; (2003); *Seminarieboken*, Chapter 4, Studentlitteratur, Lund

Olsson, Gustaf – Rosen, Christian (2004), *Industrial Automation*, Department of Industrial Electrical Engineering and Automation, Lund

Patel, Runa – Davidson, Bo; (1994); *Forskningsmetodikens grunder*, Chapter 3, Studentlitteratur, Lund

Theses
Danielsson, Fredrik (2002); *Off-line programming, verification and optimisation of industrial control systems*, page 93, Doctoral Dissertation, Högskolan Trollhättan/Uddevalla, Trollhättan

Gore, Ganesh – Sharique, Farid Syed (2004); *Virtual Verification of Auto-generated PLC program*, Master Thesis, Department of Signals and Systems, Chalmers University of Technology, Göteborg

Hammarlund, Ola - Persson, Dan (2003); *Real Simulation Control, Design of Communication for a Simulator Device*, Master Thesis, Department of Industrial Electrical Engineering and Automation, Lund University, Lund

Olofsson, Annika – Lindqvist, Marie (2003); *Simulation and programming of a prototype machine as a simulator*, Bachelor Thesis, LTH School of Engineering at Campus Helsingborg, Lund

Sivtoft, Jan-Olof (2002); *The design of a simulator for the automation industry*, Master Thesis, Department of Industrial Electrical Engineering and Automation, Lund University, Lund

Internet
CIMdata, The Benefits of Digital Manufacturing;
http://www.delmia.com/gallery/pdf/CIMdata_ROI_Study.pdf, 2005-05-13

Dassault Systèmes , Delmia V5 Automation:Fact Sheet R15;
http://www.3ds.com/fileadmin/newsevents/automation/Automtn_FactSheet.pdf, 2005-05-11

Dassault Systèmes , Press release;
http://www.3ds.com/news-events/press-room/release/836/1/, 2005-04-13

Delfoi Finland, Delmia Automation;
http://www.delfoi.com/web/news/fi_FI/Seminaari_16_6/_files/73357998079083371/default/DELMIA%20Automation%20-%20DELFOI%20Workshop.ppt_light.ppt, 2005-07-20

Delmia, Our History;
http://www.delmia.com/discover/html/our-history.htm, 2005-04-11

Delmia, Automation;
http://www.delmia.com/solutions/html/automation.htm, 2005-04-11

Delmia, Automotive;
http://www.delmia.com/solutions/html/automotive.htm, 2005-04-11

Dextus, Dassault Systèmes;
http://www.dextus.net/fo/pages/php/dassaultsystems/dassaultsystems.php, 2005-04-12

Dextus, Delmia;
http://www.dextus.net/fo/pages/php/delmia/delmia.php, 2005-04-12

Dextus, Dextus;
http://www.dextus.net/fo/pages/php/dextus/dextus.php, 2005-04-12

EC&M, What to know about PLC ladder diagram programming;
http://www.ecmweb.com/mag/electric_know_plc_ladder/, 2005-04-19

Halogen, Tetra Laval;
http://www.halogen.se/customer.asp?navid=50&item=15, 2005-04-08

Internet FAQ Archives, Asynchronous Events: Polling Loops and Interrupts;
http://www.faqs.org/docs/javap/c1/s2.html, 2005-09-09

MCADCafe, Delmia V5 R13 – Detailing and Validation Digital Manufacturing Process;
http://www.mcadcafe.com/magazine/index.php?run_date=22-Mar-2004&newsletter=1&no_bottom=1, 2005-04-11

Nationalencyklopedins Internettjänst; Tetra Laval
http://www.ne.se/jsp/search/article.jsp?i_art_id=326368&i_word=tetra%20laval, 2005-04-08

Ny Teknik, Volvo lär Ford att bygga bilar i datorn;
http://www.nyteknik.se/pub/ipsart.asp?art_id=39891, 2005-04-14

PLCopen, 1131 or 61131 : status of the Standard;
http://www.plcopen.org/TC1/1131_or_61131.htm, 2005-04-14

PLCopen, IEC 61131-3: a standard programming resource;
http://www.plcopen.org/TC1/INTRO_IEC_July02.htm, 2005-04-19

PTC, The Product Development Company;
http://www.ptc.com/WCMS/files/26504en_file1.PDF, 2005-05-04

PTC, Pro/Engineer Wildfire;
http://www.ptc.com/WCMS/files/26539en_file1.PDF, 2005-05-04

R. Morley Inc., The History of PLC;
http://www.barn.org/FILES/historyofplc.html/, 2005-05-26

Region Siljan, De lyfte landet;
http://www.regionsiljan.se/_webdoc/Data/6/Files/27/De_lyfte_landet.pdf, 2005-04-08

Rockwell Software, RSTestStand Enterprise, Getting Results Guide;
http://literature.rockwellautomation.com/idc/groups/literature/documents/gr/tstent-gr001_-en-p.pdf, 2005-05-30

Schneider Electric, The year in review (2003);
http://wib-prod.schneider-electric.com/publicprogram/corp/corp_elibrary.nsf/0/84280984FE88F46485256EC500552CA2/$file/Essentiel_2004_en.pdf, 2005-04-12

Schneider Electric, Local operations;
http://www.schneider-electric.com/wps/myportal/!ut/p/.cmd/cs/.ce/7_0_A/.s/7_0_PJ/_s.7_0_A/7_0_PJ, 2005-04-12

Stiftelsen för Kunskap- och Kompetensutveckling
http://www.kks.se/, 2005-09-09

Tarakos, IEE 11-2004, Fertige Programme für fiktive Maschinen;
http://www.tarakos.de/presse_artikel/iee-tarakos.pdf, 2005-05-13

Tech-Clarity, Digital Manufacturing;
http://www.delmia.se/pdf/techclarity.pdf, 2005-05-24

Tetra Laval, About Tetra Laval;
http://www.tetralaval.com/content/page.asp?navid=2

Tetra Laval, Tetra Laval 2004;
http://www.tetrapak.com/docs/TL040426.pdf, 2005-04-08

Tetra Pak, In figures;
http://www.tetrapak.com/index.asp?navid=6, 2005-04-08

Tetra Pak, Tetra Pak Innovation Network
http://innovation.tetrapak.com/

Tetra Pak, The Company History;
http://www.tetrapak.com/index.asp?navid=7, 2005-04-08

University of Trollhättan/Uddevalla
http://www.htu.se, 2005-09-09

WinMOD, Virtual machine for mechanical engineering; http://www.winmod.de/eng/reports/winmod_report_04.18_en.pdf, 2005-05-30

Other

Documentation to the mc_lib for B&Rs programming tool for their PLCs.

Meeting with Henrik Carlsson, Fredrik Danielsson and Bo Svensson, HTU, 2005-09-28

Phone conference with Eric Retraint, vice president sales and channels Delmia Automation, 2005-05-09

Screenshot from an example file made by Delmia, Machine14.

Appendix A
Software requirements specification

Program: DA=Delmia Automation
RST=RSTestStand + RSLogix
WM=WinMod
TC=taraVRcontrol

Requirements: R=Required
O=Optional
D=Desired

Table A.1: The requirements used for assessment of the four software products in the second part of the assessment.

Requirement	DA	RST	WM	TC	
• The software can import geometry, in Pro/E or neutral format like VRML	X	X	X	X	R
o The software can import kinematics from Pro/E	-		X		O
o Can make changes of dimensions of the geometry	P*		P*		O
• The software can handle control signals from PLC	X	X	X	X	R
o Use OPC for communication	X	X	-	X	O
o Can handle external control signals	X	X	X	X	O
▪ Use OPC for external control signals	X	X	-	X	D
▪ Use Profibus for external control signals			X		D
▪ Emulated PLC in the software	X	X	X		D
▪ Import existing PLC-code		X	-		D
▪ Export of PLC-code to existing environment	X	X	-		D
▪ Make changes in the PLC-code	X	X	-		D
▪ Connect to external HMI	X	X	X		D

Requirement	**DA**	**RST**	**WM**	**TC**	
• Visual display of the model during the simulation	X	X	X	X	R
• Automatic synchronization of emulated real-time between the PLC and the virtual machine	X		X		R
• A review of the simulation should be in real-time	-		-		R
▪ The simulation should be in real-time	X		X		D
• Internal logics can be defined	X		X		R
o Standard libraries for internal logics can be defined	X		X		O
o Internal logics are defined in IEC 61131-3 or in C	X		-		O
▪ Possibility to resize standard libraries			X		D
• The software and documentation are available in an English version	X	X	X		R
• The software will run on a PC with MS Windows	X	X	X	X	R
o The PLC-code support IEC 61131-3	X	X	-		O
o Collision analyze	X		X	X	O
▪ Set sensors for collision analyze only at critical places	X		X	X	D
o Plot of time-space results for one signal	-		X		O
▪ Plot of time-space results for many signals	-		X		D
o Plot of time-sequence results for one signal	-		X		O
▪ Plot of time-sequence results for many signals	-		X		D
o Easy handle of sensors, i.e. graphical configuration	X		X	X	O

Table A.2: Summary of how many requirements in the software requirements specification the four different software fulfill.

Summary:

Software	**Σ R (max 8)**	**Σ O (max 11)**	**Σ D (max 12)**	**Σ total (max 31)**
Delmia Automation	7	7	7	21
RSTestStand + RSLogix	5	3	6	14
WinMod	7	7	8	22
TaraVRcontrol	4	4	2	10

* It is partly fulfilled and means it is possible to change the dimensions of the model in the simulation software, dependent to what CAD-software that is used.
X means that the requirement is fulfilled.
- means that the requirement is not fulfilled
An empty box means the answer was not found

Some of the vendors say their product will fulfill some of the requirements in the future. In this case it has been marked as the requirement not is fulfilled in the table because they do not have the function today.

Appendix B
List of signals

Here follows a list of signals that are used as interface between the equipment and the virtual PLC.

Three types of signals have been used, Boolean can only take the value of true or false, integer can be any integer between –2147483648 to 2147483647, pure is a signal that gives a pulse when it is emitted.

Table B.1: List of signals used in the simulation.

Machine

Input

Name	Type	Value	Comment
MovePackage	Boolean	false	Move package to origin
		true	Move package to position for heating
MoveHeater	Boolean	false	Move heater to origin
		true	Move heater to position for heating
EmergencyStop	Boolean	true	No operation is possible at all
Heating	Boolean	true	Set the heater to “On”
Homing	Pure		Starting the homing of the machine

Output

Name	Type	Value	Comment
PackageEnd	Pure		Acknowledge signal for movement of the package is finished
HeaterEnd	Pure		Acknowledge signal for movement of the heater is finished

Sensor

Output

Name	Type	Value	Comment
SensorDetection	Boolean	true	A package is in position for heating

HMI

Input

Name	Type	Value	Comment
Busy	Boolean	true	Light the “Busy”-light
InPosition	Boolean	true	Light the “Package in position”-light

Output

Name	Type	Value	Comment
Start	Pure		Signal for start continuous operation
Stop	Boolean	false	Continuous operation is possible
		true	Continuous operation is not possible
Cycle	Pure		Signal for start operation of one cycle
NextStep	Pure		Signal for move one step forward in the program
Mode	Integer	1	Set mode into “Continuous”
		2	Set mode into ”Cycle”
		3	Set mode into ”Step by step”
EmergencyStop	Boolean	true	No operation is possible at all

Appendix C
Code for the virtual machine

In this section the internal logic of the mechanical model and the HMI are displayed, and also the control logic of the virtual PLC. The internal logic is the behavior of the different parts and the control logic describes how the virtual PLC will react on different input signals and what it outputs.

To clarify where in the structure the different code is, a figure of the code structure is placed above every block of code.

Internal logic of the mechanical model

The internal logic of the mechanical model is divided in one root block and five macros, at three levels.

Root
This is the root of the internal logic for the mechanical model, an overview is showed in Figure C.1 and the code in Figure C.2. This will start up the other activities.

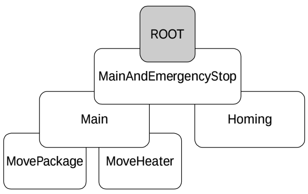


Figure C.1: Overview of the code structure for the machine.

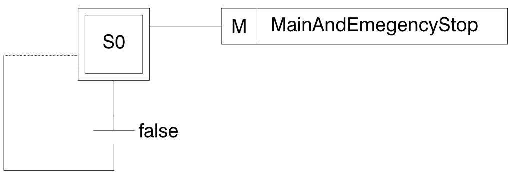


Figure C.2: The code for the root.

Macro: MainAndEmergencyStop
The MainAndEmergencyStop macro is the next layer under Root, see Figure C.3, and takes care of the signals EmergencyStop and Homing, see Figure C.4. If the EmergencyStop becomes true it will stop any active movement of the machine. The macro Homing will be activated when the signal Homing is emitted.

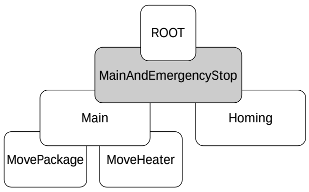


Figure C.3: Overview of the code structure for the machine.

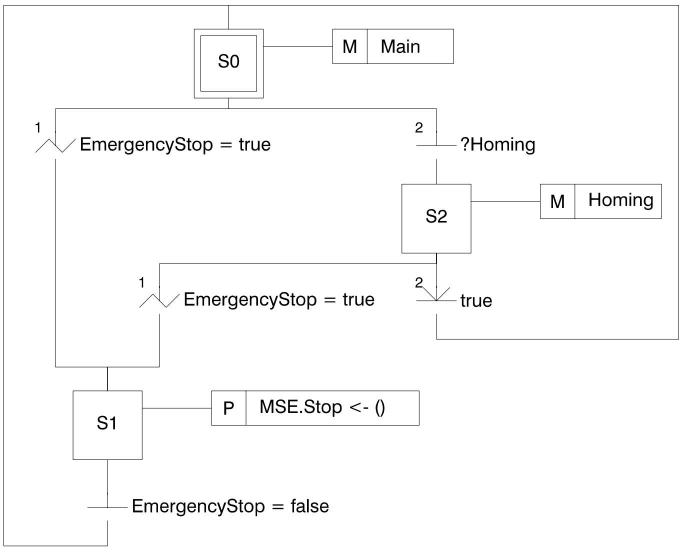


Figure C.4: The code for the macro MainAndEmergencyStop.

Macro: Homing

At layer three the Homing will be found, see Figure C.5. Because of some limits in the possibilities to control a movement a real homing could not be done. Therefore a simulated homing has been written, see Figure C.6, to imitate the real process.

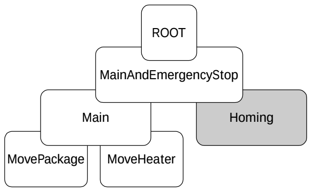


Figure C.5: Overview of the code structure for the machine.

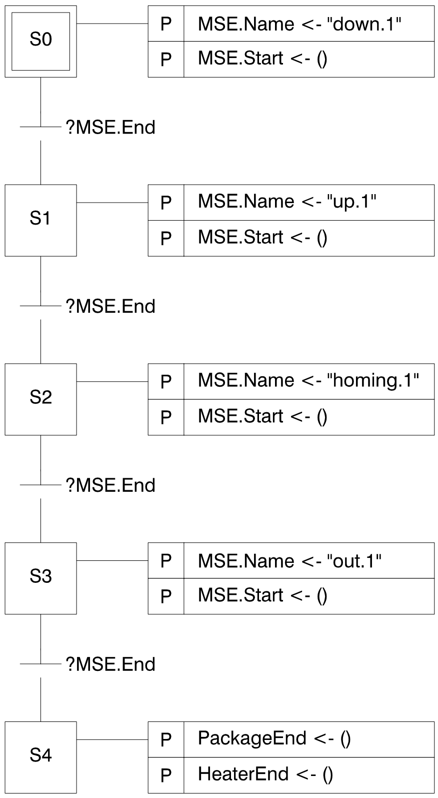


Figure C.6: The code for the Homing macro.

Macro: Main

The Main macro will start one of the macros MovePackage or MoveHeater depending of which of the signals are emitted, see Figure C.8. See Figure C.7 for the overview of the layers.

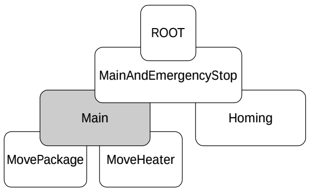


Figure C.7: Overview of the code structure for the machine.

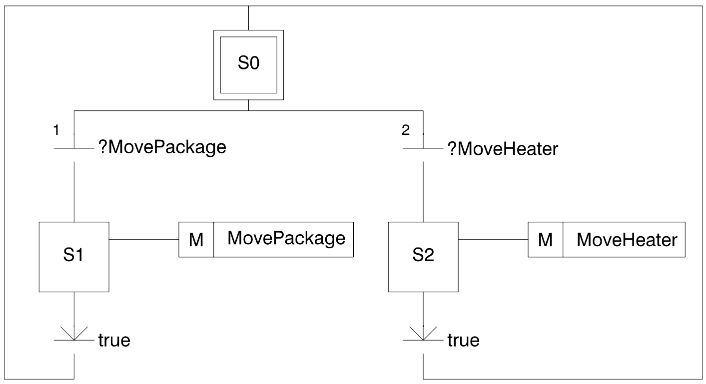


Figure C.8: The code for the Main macro.

Macro: MovePackage

The MovePackage macro is at the lowest layer, see Figure C.9. This macro will move the package holder to its origin or to a position under the heater, see Figure C.10.

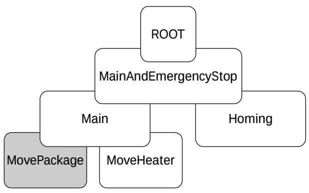


Figure C.9: Overview of the code structure for the machine.

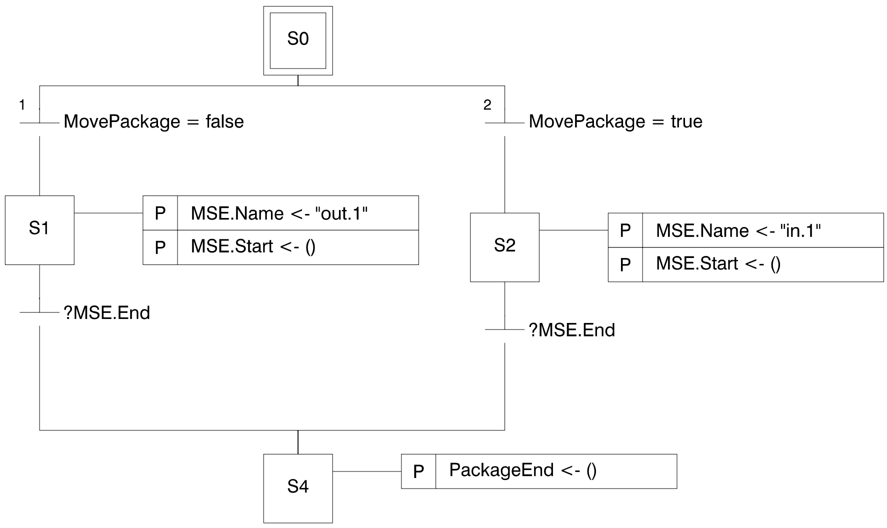


Figure C.10: The code for the MovePackage macro.

Macro: MoveHeater

The macro MoveHeater is at the same layer as MovePackage, see Figure C.11. This macro will move the heater to the origin position or move it to the lower position, see Figure C.12.

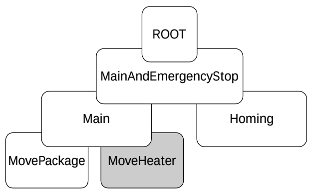


Figure C.11: Overview of the code structure for the machine.

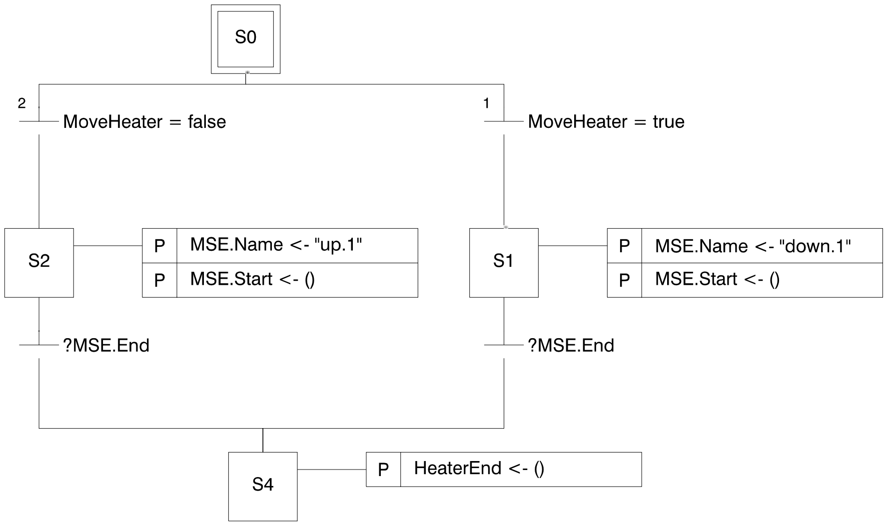


Figure C.12: The code for the MoveHeater macro.

Internal logic of the HMI

The internal logic of the HMI is divided into one root block and three macros, at two levels, see Figure C.13.

Root

The root will start up the main program and TestriggsHMI_TestPanel, see Figure C.14. The TestriggsHMI_TestPanel is an auto-generated program done by DA during the creation of the HMI. For example the program emits signals when a button is pushed.

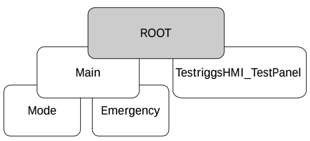


Figure C.13: Overview of the code structure for the HMI.

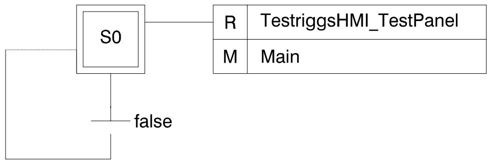


Figure C.14: The code for the root for the HMI.

Macro: Main

The Main macro is at the layer below the root, see Figure C.15. This macro will activate the Mode macro immediately. If the emergency stop button is pushed the Emergency macro will be activated, see Figure C.16.

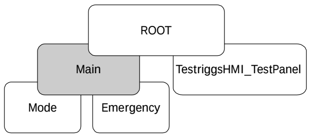


Figure C.15: Overview of the code structure for the HMI.

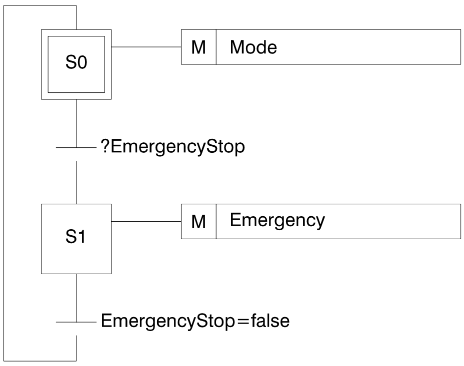


Figure C.16: The code for the Main macro.

Macro: Mode

The Mode macro is at the lowest layer, see Figure C.17. There was no possibility to have buttons that affect each other, therefore it was needed to write this function. The mode macro lights and switches off the buttons depending on which mode is activated. A signal is also set depending on which mode is activated. This signal is an integer that is used by the PLC, see Figure C.18 for the code.

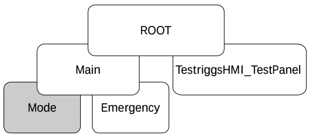


Figure C.17: Overview of the code structure for the HMI.

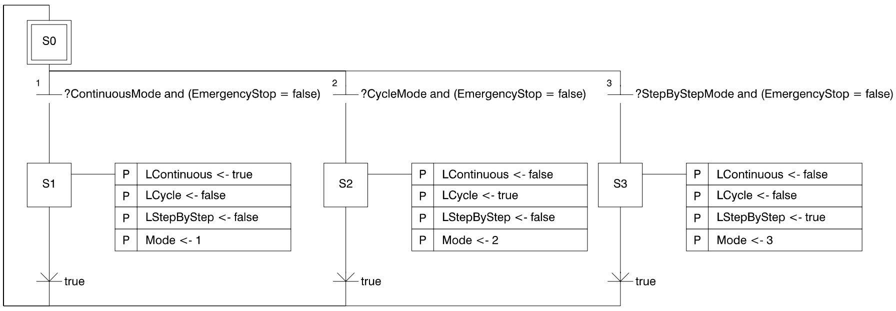


Figure C.18: The code for the macro Mode in the HMI.

Macro: Emergency

The Emergency is in the same layer as the Mode macro, see Figure C.19. The Emergency macro will switch off all lights when the emergency stop button is pushed, see Figure C.20.

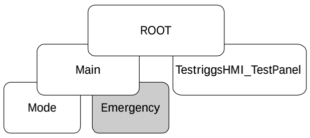


Figure C.19: Overview of the code structure for the HMI.

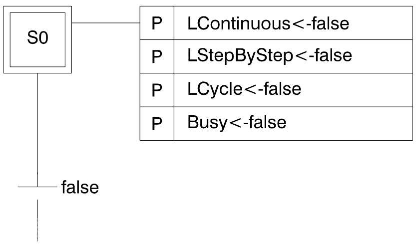


Figure C.20: The code for the Emergency macro.

Control logic of the virtual PLC

The internal logic of the virtual PLC is divided in one root block and three macros, at two levels, see Figure C.21.

Root

The root is the basis for the control logic. It starts with emitting the homing for the machine. When the homing is finished it will activate the macro MainAndReset. This macro will be running until the PLC receives an emergency stop signal, and then stops the machine and switches off the lights at the HMI, see Figure C.22.

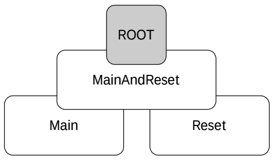


Figure C.21: Overview of the code structure for the PLC.

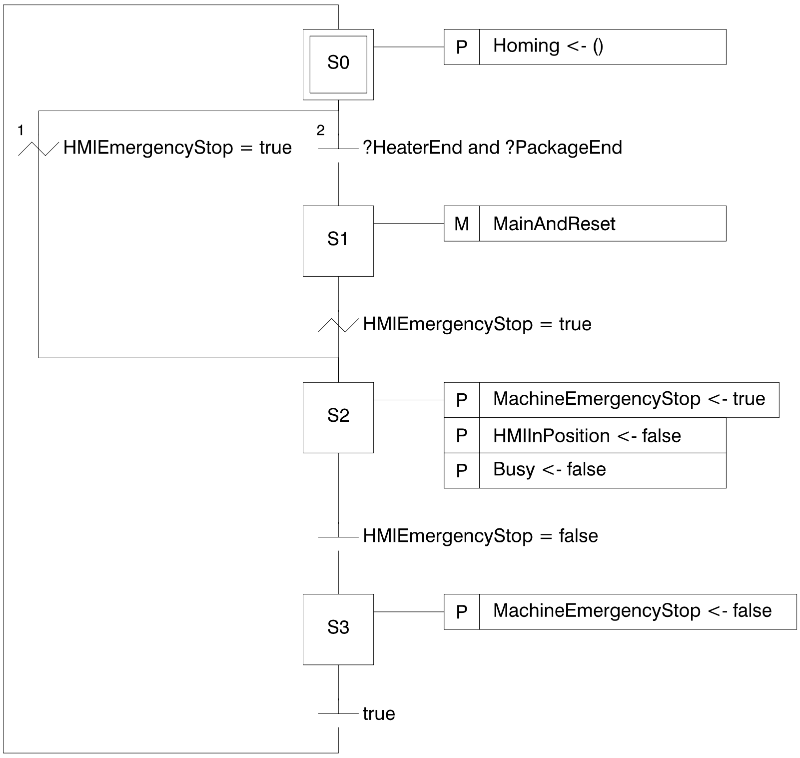


Figure C.22: The code for the root for the PLC.

Macro: MainAndReset

The MainAndReset layer has two macros below, see Figure C.23. The first macro is the Main macro that will be activated until the mode is changed and then activate the Reset macro. When the Reset macro is done the Main macro will be reactivated, see Figure C.24.

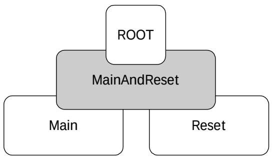


Figure C.23: Overview of the code for the PLC.

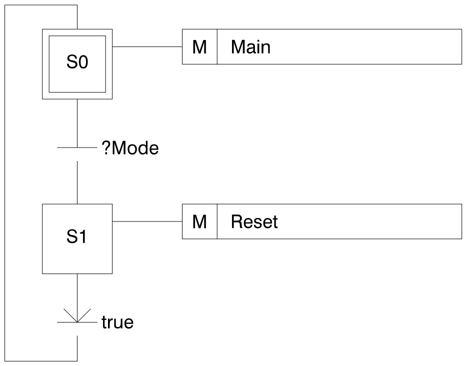


Figure C.24: The code for the macro MainAndReset.

Macro: Main

This macro is where all control logic is placed. In Figure C.25 an overview of the code structure can be seen. The Main macro has three alternative branches, one for each mode, see Figure C.26. In each branch all steps will be executed so a complete cycle is done and then start over again.

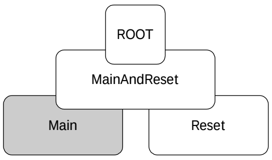


Figure C.25: Overview of the code structure for the PLC.

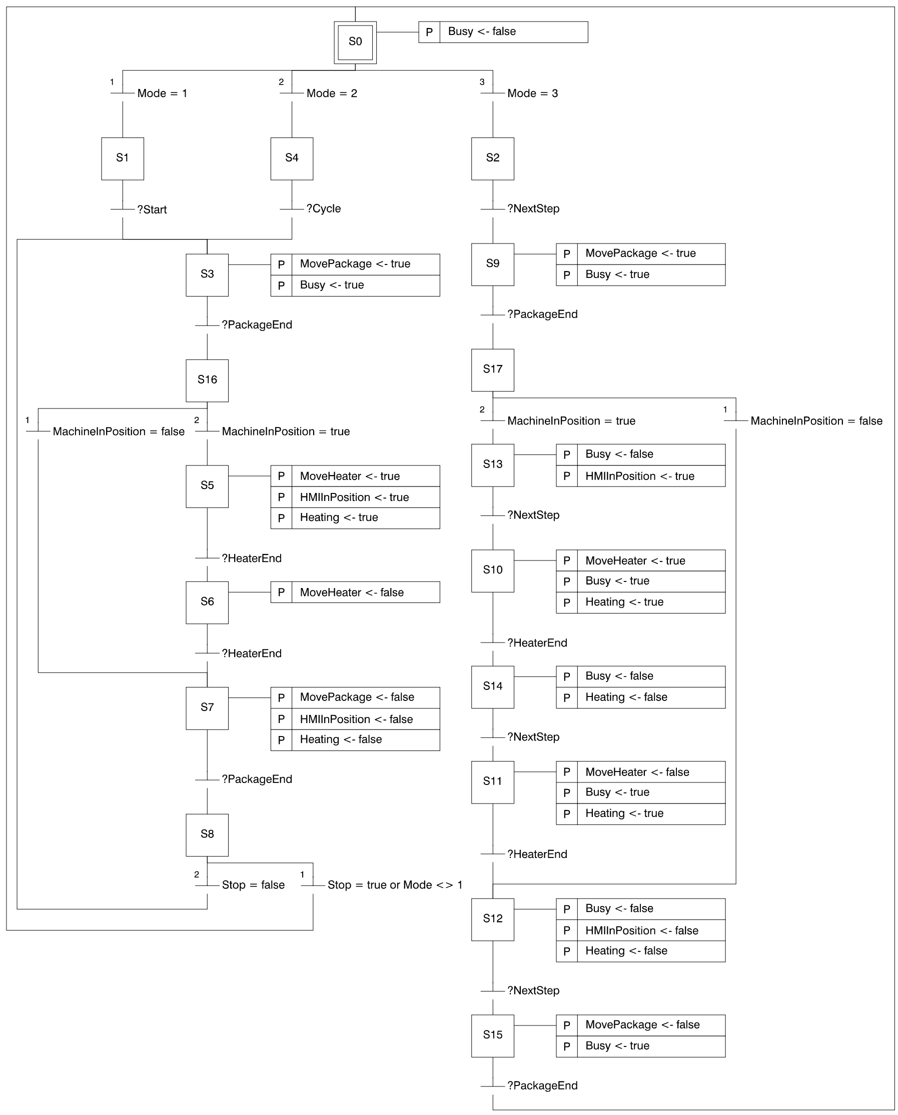


Figure C.26: The code for the Main macro.

Macro: Reset

The reset macro will be run each time the mode is changed and is at the lowest layer in the PLC, see Figure C.27. It first stops the machine and then moves the heater back to the upper position and then moves the package holder back to its origin, see Figure C.28. By moving the heater and the package holder in this order there will be no chance to harm the package.

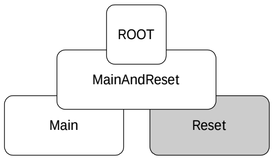


Figure C.27: Overview of the code structure for the PLC.

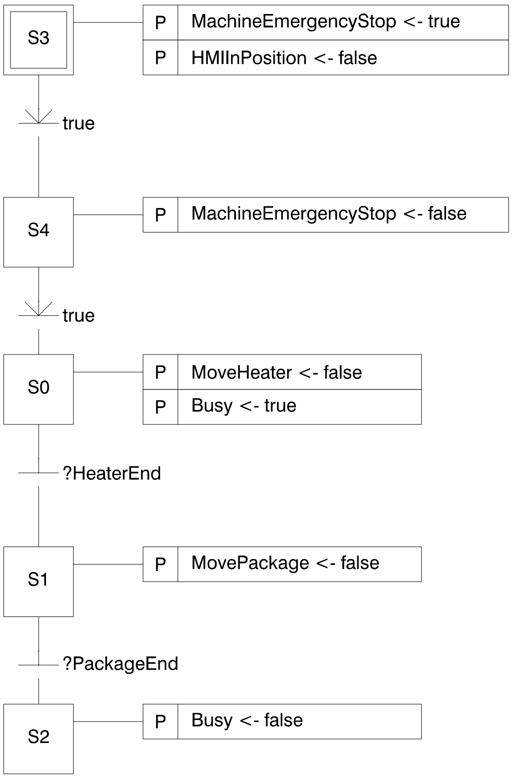


Figure C.28: The code for the Reset macro in the PLC.

Appendix D
Delmia Automation comments from Tetra Pak

Case#	Date of discover	Sent to Delmia	Received answer	Problem solved	Description By Customer	Module	Corr. Planned	Reason / Correction Planned	Version
1	2005-06-01	2005-06-01	2005-06-01	No	There are no documentation included and no help-files available in the software.	All		We did not include the documentation in our Standard Dassault Systèmes format in R15.	V5R15
2	2005-06-07	2005-06-09	2005-06-09	Yes	Tools to create sensors are not visible.	DB		Incorrect installation, you must enter two environment variables manually. DLM_ENABLE_SENSORS value:1 and DLM_WSU_PRODUCT value:1 (enable display of sensor tools and creating robot tasks)	V5R15SP1
3.1	2005-06-07	2005-06-09	2005-06-10	Yes, 2005-07-11	Problems with importing STEP-files.	All		Problems with importing STEP-files is due to our license and that the MULTICAx is needed. See also case #40.	V5R15SP1
3.2	2005-06-07	2005-06-09	2005-06-10	No	Problems with importing VRML-files.	All		Problems with importing VRML-files is a bug (confirmed by Lise Upjohn). But even after correction of the bug assemblies in VRML will not be supported.	V5R15SP1
4	2005-06-20				When TestScenario created, ports are not added to Ports but are in Instances -> Ports. Must be mapped manually to get the TestScenario to work.	SDB			V5R15SP1
5	2005-06-20	2005-09-19	2005-09-22	No	When Build block, the Options "-debug" is added each time so after three times Build, three "-debug" are added after each other.	SDB, LCM		The debug option is used to add debug information in the binary code. the "-debug" option added at each build is a bug. But there is no impact, the option is taken into account only once.	V5R15SP1
6	2005-06-21	2005-09-19	2005-09-22	Yes	Creating reports is not possible, this error message is shown. <script xlink:href = "C:\Program Files\Dassault Systemes\B15\intel_a\startup\Reporting\ToolTip.js"></script> ---^	PSA	To be corrected in R17	I have identified the bug. This bug occurs when there is a space in the path of the customized report (field "XSL StyleSheet/VBA" field). The workaround is to move the "reporting" directory to have no sapce in the path. For the present moment, you have a space in the directories "Program files" and "Dassault	V5R15SP1

Case#	Date of discover	Sent to Delmia	Received answer	Problem solved	Description By Customer	Module	Corr. Planned	Reason / Correction Planned	Version
								Systems". The correction is forecasted for the next release.	
7	2005-06-21	2005-09-19	2005-09-22	No	There are no PLCs in the list to download the logic to a real PLC.	CSS		A 'PLC setup' program has to be installed on your PC and the path of this program has to be added to the installation path of Delmia automation. For the R15 version, the list of the 'PLC setup' is the following one: - OMRON - SIEMENS - SCHNEIDER The 'PLC setup' translates the LCM application to an application compatible with a specific PLC. Schneider and Omron developp their PLC setup. We developp the Siemens PLC setup. So, the installation depends of the PLC you use.	V5R15SP1
8	2005-06-23	2005-09-19	2005-09-22	Partly	Pictures in HMI, both background and Create Picture must be in Temp-directory, otherwise an error message is displayed. When create own pictures to buttons, that not are in …/resource/graphics you get the error message AUTHmiNoGoodDir.	HMI		In the menu "Tool" -> "Options", directory "Logic Control Modelling", "HMI Control Panel Design", you have to set the directory of the images. (nb: Be carrefull when you export your project to another PC, the same directory has to be set)	V5R15SP1
9	2005-06-27	2005-06-29	2005-06-29	Yes	All steps and actions are completely black when printing SFC-code, only conditions are readable.	SDB, LCM		Change printer options, Print ->Options, uncheck "Print White Vectors as Black".	V5R15SP1
10	2005-06-27	2005-07-04	2005-07-04	Yes	When a simulation is run all colors disappear, the model appears in gray and will not get its original colors back when the simulation is done.	SDB, RTD, CSS		Change color to the right one and save the initial states, in Robot Task Definition choose Set Initial State.	V5R15SP1
11	2005-06-29	2005-09-19	2005-09-22	No	Own defined pictures of buttons with different size in Icon on and Icon off resizes the picture to smaller and smaller every time you activate them. After a few pressure its not possible to see the button. This problem occurs when the Icon on is smaller than icon off. Example Icon off AUTHmiPushButtonOff.bmp and Icon on	HMI	To be corrected in R17	This bug should be corrected in the next version (R17)	V5R15SP1

Case#	Date of discover	Sent to Delmia	Received answer	Problem solved	Description By Customer	Module	Corr. Planned	Reason / Correction Planned	Version
					3DExpertLogoP2NT.bmp.				
12	2005-06-29	2005-09-19	2005-09-22	No	In Time Table Editor for Mechanisms, decimal must be entered as a dot but is showed as a comma. If you only change the integer part you get an error message.	DB		I have forwarded this as bug to the development team. Unfortunately, I have no workaround to propose to you. The objective date for the correction of the bug is the next release, so there is no solution available before few months.	V5R15SP1
13	2005-06-29	2005-09-19	2005-09-22	No	Mechanisms are dependent of in which order you click on Angle driven and Length driven. We use cylindrical joint and if we change order to click on angle and length driven the joint began to rotate instead of translate.	DB		I have forwarded this as bug to the development team. Unfortunately, I have no workaround to propose to you. The objective date for the correction of the bug is the next release, so there is no solution available before few months.	V5R15SP1
14	2005-06-29	2005-06-29	2005-06-29	Partly	Motions with mechanisms do not follow the times defined in the Time Table Editor.	DB		Time for a mechanism to move is defined in Tasklists -> MoveHomeActivity, RMB -> Properties, Specified Cycle Time.	V5R15SP1
15	2005-06-29	2005-07-04	2005-09-22	No	When trying to check which home-position that is in a MoveHomeActivity in a TaskList, via Definition, we get the message "This action is not supported".	DB		There is no definition for an activity, this is why you have an error message. Once a task is created, you are not able to find which home position was initially selected.The recommendation is to create the task with the name of the home position.	V5R15SP1

Case#	Date of discover	Sent to Delmia	Received answer	Problem solved	Description By Customer	Module	Corr. Planned	Reason / Correction Planned	Version
16		2005-05-09	2005-05-09, see comments for more dates.	Partly	Asked Eric Retraint at Delmia about documentation for DA. Also asked Lise Upjohn at Delmia Automation Support for documentation continuously.	All		Received some User's Guide and Training Manuals. We also get some Powerpoint presentations. 25/5 Sensor function (DELMIA Automation - Sensor1.ppt) 1/6 LCM Editor User's Guide 1/6 Smart Device Builder User's Guide 1/6 Controlled System Simulator User's Guide 16/6 Controlled System Simulator Training Manual (Complete_Simulation.ppt) 16/6 LCM Editor Training Manual (LCM.ppt) 17/6 Smart Device Builder Training Manual (Smart_Device.ppt) 23/6 Device Builder User's Guide 23/6 Robot Task Definition User's Guide 13/7 Resource Sensor (Sensors.ppt)	V5R15
17	2005-06-08	2005-06-09	2005-06-13	Yes	Problems to create mechanisms with imported CAD-files from Pro/E.	DB		First create Frames of Interest and then use Joint from Axis to create mechanisms.	V5R15SP1
18	2005-06-29	2005-06-29	2005-06-28	No	Debugging HMI with No Fixed buttons and using Next Cycle is not possible.	SDB, LCM		Forwarded to the development team at Delmia by Lise Upjohn.	V5R15SP1
19	2005-06-29	2005-06-29	2005-06-29	Yes	Parallel branches cannot be made in SFC.	SDB, LCM		Use many actions in a block, i.e. macros. According to Delmia this is a quite flexible solution. With Join at the transition it will wait until all actions at the previous step are finished	V5R15SP1
20	2005-06-30	2005-09-19	2005-09-22	No	Simulation with sensors only works first time. After Pause and Jump to Start the sensor signal is true even if the object not is on position.	SDB	To be corrected in R17	The "Jump to start" command resets the graphic and resets the SFC cod (that means the code goes back to the first step) but the port and signal values are not reset, they keep their last value. I agree with you , this behavior is not very coherent. In the next release, it is planned to reset all the ports and signals values. The "Jump to start" will be in fact a complete reset.	V5R15SP1
21	2005-06-30				When changing times in a MoveHomeActivity and then run the simulation, the simulation will stop after a while and will not continue.	SDB, CSS			V5R15SP1

Case#	Date of discover	Sent to Delmia	Received answer	Problem solved	Description By Customer	Module	Corr. Planned	Reason / Correction Planned	Version
					After escape from the simulation and then go in to the simulation again and run it, it will work.				
22	2005-07-01				In Device Task, Task Operations, it's not possible to redefine or delete operation.	DB			V5R15SP1
23	2005-07-04				In an HMI, when grouping two groups of object they move and is placed over each other.	HMI			V5R15SP1
24	2005-07-04				You can only move a label in an HMI when you are over a sign, you can not mark the label only over the background of the label.	HMI			V5R15SP1
25	2005-07-04	2005-07-04	2005-07-04	Partly	You can only have one delay on a mechanism in a tasklist. If there is a delay on the second mechanism it will not be considered. Deleting the delay in the first mechanism lead to the delay in the second mechanism will be considered.	DB, SDB		The delay for a mechanism is only considered the first time a tasklist is running, and the delay starts from the beginning of the tasklist and not from when the mechanisms move-activity starts. Add a Delay in the tasklist instead of specify it in the mechanisms properties.	V5R15SP1
26	2005-07-04	2005-07-04	2005-07-04	No	It's not possible to control the time of a motion for a mechanism in the SFC-code.	LCM		This functionality is not available for the present moment and is not planned for the next release (R16) in September 2005.	V5R15SP1
27	2005-07-04	2005-07-04	2005-07-04	No	It's not possible to control the speed of a motion for a mechanism in the SFC-code.	LCM		This functionality is not available for the present moment and is not planned for the next release (R16) in September 2005.	V5R15SP1
28	2005-07-04	2005-07-04	2005-07-04	No	A simulation cannot be made in full-screen.	SDB		It is not possible to have a full screen simulation.	V5R15SP1
29	2005-07-05	2005-07-12	2005-07-13	No	Opening an HMI, choosing Edit Panels, close the window and then again choose Edit Panel an error message will occur. Header: A check on data will be launched. Message: Command interrupted. After that you cannot change module, the message appear several times after each other. After some times you get a Runtime exception with the message Click OK to terminate. There is only an OK-button and clicking on that will close the program. Sometimes it's possible to close the document but when trying to open it it will message that the document already is open, even if there is none open documents.	HMI		They have tried do recreate the error at the support but it did not appear at their computers.	V5R15SP1

Case#	Date of discover	Sent to Delmia	Received answer	Problem solved	Description By Customer	Module	Corr. Planned	Reason / Correction Planned	Version
30	2005-07-05				Moving several objects at the same time in an HMI will move the objects in a different way. Some moves very much and some just a little bit.	HMI			V5R15SP1
31	2005-07-04	2005-07-04	2005-07-04	No	It's not possible to change the color of a part from the SFC-code.	SDB, LCM		This is not possible.	
32	2005-07-06	2005-07-06	2005-07-07	No	Moving an action with a macro to another step will cause the code in the macro to lay over each other in a mishmash.	SDB, LCM	To be corrected in R16.	This is a bug in the current version. It has been corrected for the R16 release.	V5R15SP1
33	2005-07-06	2005-07-06	2005-07-07	No	Importing a product in another product will add an .1 to the items in a tasklist. This will cause that the code not will work. I. e out becomes out.1.	DB, SDB	Correction forecasted to R16.	This behavior is a bug in the R15 version, confirmed by Lise Upjohn 2005-07-07. The current workaround is to change the taskNames to its original names or add a .1 to the names in the SFC-code.	V5R15SP1
34	2005-07-07				The performance of DA will increase considerably if two processes for Roxio Easy CD creator will be closed. In Task Manager, end the process Directcd.exe and CreateCD50.exe.	All			V5R15SP1
35	2005-07-06	2005-07-06	2005-07-07	No	Deleting a sensor will demand a restart of the DA to let the sensor disappear at the model.	DB	To be corrected in R16.	There is a bug in in the refresh of the screen after deleting a sensor, confirmed by Lise Upjohn 2005-07-07. The sensor is however deleted, its not displayed in the specification tree.	V5R15SP1
36	2005-07-06	2005-07-06	2005-07-07	No	In the SFC-code it's not possible to switch between hiding and showing a part in the model.	SDB, LCM		This functionality is not available and is not forecasted for the next release.	V5R15SP1
37	2005-07-06	2005-06-06	2005-07-07	No	You can only create one testscenario.	SDB		There is only one Testing Scenario by Internal Logic level. A workaround is to got two steps with a False transition between and then create several macros with different testscenarios. Drag the testscenario you will test to the initial step to be executed.	V5R15SP1
38	2005-07-08	2005-07-12	2005-07-13	No	The autogenerated Program1 in a sensor resource must be completed to work properly. Between the step SensorLogicStep0 and the transition 'SensorLoopTrans' an action with the condition Sensor.Initiate=false and a step	SDB		The user has to rewrite it, the default behavior is not very useful.	V5R15SP1

Case#	Date of discover	Sent to Delmia	Received answer	Problem solved	Description By Customer	Module	Corr. Planned	Reason / Correction Planned	Version
					with the same action as to SensorLogicStep0 is inserted.				
39	2005-07-08	2005-07-15	2005-08-01	No	After mapping signals, at an expanded port, between ports in CSS and shrink the port the mapping disappear. Expansion of the port again will not map the signals, it must be redone.	CSS, SDB, LCM		When you map signals of structured ports, all the signals are mapped element by element. If then you use shrink ports, this means that you break the mapping element by element. The mapping has to be done at the top of the structure.	V5R15SP1
40	2005-07-11	2005-07-12	2005-07-12	Yes	Still problem with importing STEP-files though MultiCax and a license is installed.	All	New licenses sent.	New licenses were sent. Then it will work.	V5R15SP2
41	2005-07-12	2005-07-15			A port with the name EmergencyStop is mapped to a signal EmergencyStop. Connection between them and a virtual PLC will not work, but if the port's name is changed to Estop it will work.	LCM		See case #42.	V5R15SP2
42	2005-07-13	2005-07-15			Two ports, under two different levels, cannot have the same name. No error message will be displayed but the simulation will not be displayed. Ex Machine.EmergencyStop and HMI.EmergencyStop.	SDB, LCM			V5R15SP2
43	2005-07-13	2005-07-15	2005-08-01	Yes	Updating the .CATProduct-file with Control logic of a virtual PLC will not have an effect on the complete product (with model and HMI). The virtual PLC must be added again and all ports and signal must be mapped together to get it working. Just to do an update in the control logic. Another way to do a update is to change the file with the complete product. But, in that case, the original file with the control logic will not be affected.	CSS		The changes between the original control and the complete product are not taken into account because the logic between the reference and the instance are not synchronized. Right-click on the module "Control Logic Workspace", select "Propeties". To have the same logic between the reference and the instance, select "Same contain with reference", in this case the changes done in the instance will be duplicated on the reference.	V5R15SP2
44	2005-07-14				Starting a Task Manager during a simulation will let the graphic of the model to be corrupted.	CSS			V5R15SP2

Case#	Date of discover	Sent to Delmia	Received answer	Problem solved	Description By Customer	Module	Corr. Planned	Reason / Correction Planned	Version
45	2005-07-11	2005-07-12	2005-07-13	No	If the port Sensor_Program1_IL_Main (the port out from the Luftrigg product) is deleted the sensor stops work, all signals are left just the port is deleted.	CSS	Will be changed in R16.	This is a feature of the program. All the simulation ports (MSE for the device/robot's tasks, Sensor_program1 for the sensors) have to be moved up to the application's root in order to work with the 3D simulation. This feature will be changed in the R16 version.	V5R15SP2
46	2005-07-11	2005-07-12	2005-07-13	No	When a product with a working sensor is imported into a new product, the sensor signal doesn't work. It will always be false even if the beam is broken.	SDB		The resource sensor has to be at the root to work but for the present moment, I don't know why. I have forwarded your question to the development team.	V5R15SP2
47	2005-07-07	2005-07-12	2005-07-13	Partly	It is possible to record a scenario in the debugger to .eso-files, but there are no tools to analyze the created files.	SDB	Debugger works with .eso-files in R16.	The scenario files allows you to record the Input/output of the simulation.You can open it with "notepad". You will see that I/O are recorded at each cycle. However, in the R15 version, the debugger does not work with the 3D simulation so you have to force the returned values of the MSE port (MSE.END). For the R16 version, the debugger will work with the 3D simulation, so it will be possible to analyse the signals and the movements of the mechanism from the eso files.	V5R15SP2
48	2005-07-18				Problem using the debugger at the complete model, only an action with a ? and "Nothing" is displayed.	CSS			V5R15SP2
49	2005-07-19				Degugging a step with two macro and a joint-transition after will show different behavior in Run-mode or in Step-by-step mode. In Step-by-step mode it will go up to the initial step though the step not was connected.	LCM			V5R15SP2
50	2005-07-13	2005-07-15	2005-08-01	No	Problems with updating a changed CAD-geometry from Pro/E.	DB, SDB	No	There is no update from an external CAD-product. If you change the model, you have to import it and build the mechanism again.	V5R15SP2
51	2005-07-13	2005-07-15	2005-08-01		When running a simulation in CSS and switching between hide and show some home-positions will be redefined. This does not appear if the part is switched before the simulation starts.	CSS		They cannot recreate it at DA support.	V5R15SP2