Planning and Development tools for Automation System Design

Master's Thesis

MARCUS DAHLMARK
MIKAEL SEGERLUND

Department of Signals and Systems
CHALMERS UNIVERSITY OF TECHNOLOGY
Göteborg, Sweden, 2011

Abstract

In the development process of manufacturing systems late modifications of the systems are costly and time consuming. These modifications can be caused by human errors while planning the manufacturing. To find the errors at an early stage virtual models can be used. A virtual model can be a static three dimensional description of the manufacturing system or a dynamic model that also describes the movements of robots and other equipment. The movement of robots and other equipment is typically controlled by programmable logic controllers. In order to be able to evaluate the behavior of a manufacturing system in a virtual model it is thus necessary to also include models that describe the behavior of the programmable logic controllers.

The manufacturing industry demands flexible systems because of the short product life cycles of today. Flexibility can be achieved by working with operations and relations between the operations in the manufacturing process. Each operation is processing the product in some way, for example a new part can be assembled. The relations defines how the operations are related to each other, for example in which order the operations must be performed. For large scale production systems it can be unmanageable to determine in which sequence the operations should be performed and with the available methods today the generation of the sequences are made manually.

To manage these problems three tools have been evaluated; Google SketchUp for static models, V-REP for dynamic models and Sequence Planner for operation modeling. These software tools have been applied to a real automation project in order to evaluate the tools and improve the development process of automation systems. The automation project consist of an industry cell where solar panels are assembled together. The cell is a semi-automatic cell where an operator manages the input and output of materials and a robot assembles the panel.

Sequence Planner is in an early stage of development and do not include several important functionalities needed to automatically generate the sequence of operations. However, it can be used to manually create the sequences. Sequence Planner is developed to describe manufacturing systems where one product at a time is processed. In our cell several products were processed at the same time which caused problems modelling the manufacturing system. Suggestions on how to further develop Sequence Planner is presented in the report.

V-REP can in combination with models from Google SketchUp be used to simulate dynamic behavior. This can be used to demonstrate and verify prototype solutions. This was done with the industry cell in the automation project. Physics engines in a simulation tool can be used to achieve a realistic behavior, simulate physical properties and decrease the computational load. Different types of conveyors where evaluated to find a good solution. The best result were obtained by a static conveyor to which a script where added to simulate the behavior of a dynamic conveyor with moving parts but without the high computational load.

Keywords: Sequence Planner, sequence planning, PLC, simulation, V-REP, SketchUp, physics engine

Preface

This report describes a master thesis which has been performed during spring term 2011 at the master's programme Systems, Control and Mechatronics at Chalmers University of Technology in Gothenburg, Sweden. The master thesis includes 30 high educational credits and has been performed at Teamster AB and at the Department of Signals and Systems.

Chalmers University of Technology, Gothenburg 2011

Marcus Dahlmark
Mikael Segerlund

Thanks to

We would here like to say a big thank you to the persons and companies that have contributed with knowledge and thus making this project possible.

- **Thomas Lydhig**
  Manager for Robot Automation at Teamster AB. Thomas has been our supervisor and has served as a sounding board for all types of ideas and problems.

- **Knut Åkesson**
  Researcher in industrial automation at Chalmers University of Technology. Knut has been our supervisor at Chalmers and contributed with ideas for the project and has been a great help in producing this report.

- **Marc Freese**
  The founder of the software V-REP. Marc has contributed with knowledge, support and solutions for the software in an outstanding way.

- **Kristofer Bengtsson**
  Industrial PhD at the Department of Signals and Systems and at Teamster AB. Kristofer has contributed with important knowledge about Sequence Planner.

- **Patrik Magnusson**
  PhD at the Department of Signals and Systems at Chalmers. Patrik has contributed with support and ideas about Sequence Planner.

- **Nina Sundström**
  PhD at the Department of Signals and Systems at Chalmers. Nina gave us feedback on the report and information about Sequence Planner.

- **Sajed Miremadi**
  PhD at the Department of Signals and Systems at Chalmers. Sajed has helped us with both Sequence Planner and Supremica and gave us an introduction to the softwares.

- **Mohammad Reza Shoaei**
  PhD at the Department of Signals and Systems at Chalmers.
  Mohammad introduced us for Sequence Planner and has helped us with information about different simulation tools.

- **Teamster AB**
  We would also like to thank all the employees at Teamster AB that has treated us in a very nice way and made us feel welcome.

- **Laurence Schneider**
  Laurence Schneider at K-Team has contributed with information about how they work with V-REP.

Contents

Abbreviations **2**

1 Introduction **3**
1.1 Background 3
1.2 Purpose 4
1.3 Aim 5
1.4 Limitations 5
1.5 Method 6
1.6 Report structure 6

2 Description of the S*Solar industry cell **7**
2.1 The S*Solar project 7
2.2 The S*Solar cell 8

3 Evaluation of Sequence Planner **12**
3.1 Sequence Planner 12
3.1.1 Resource or product focused SOP 13
3.2 Implementation of the S*Solar project in Sequence Planner . 14
3.3 Missing features and problems in Sequence Planner 19
3.4 Future usage of Sequence Planner 21
3.5 Alternatives to the SP method 22

4 Evaluation of V-REP **27**
4.1 V-REP 27
4.1.1 Features of V-REP 27
4.1.2 Companies using V-REP 29
4.1.3 Development of V-REP 30
4.2 Implementation of the S*Solar project in V-REP 30
4.2.1 Evaluation of different conveyor solutions 30
4.3 Effective use of V-REP 32
4.3.1 Pure shapes 33
4.3.2 Module based use of V-REP 34

4.3.3 Targets 35
4.4 Advantages with a physics engine 36
4.5 V-REP at Teamster AB 37

5 Results 39

References i

A Sequence Planner overview A-1

B V-REP A-5
B.1 Importing Google SketchUp models to V-REP A-5
B.1.1 Colors and textures A-5
B.1.2 Grouping of objects A-6
B.2 How to make pure shapes A-6
B.3 Video recorder A-6

Abbreviations

API *Application Programming Interface*, set of rules and specifications that a software can follow to access and make use of the services and resources provided by another software.

CAD *Computer Aided Design*, design and modelling made with help of a personal computer, often in three dimensions.

EFA *Extended Finite Automata*, an ordinary automata (theoretical model of a computer hardware or software system) extended with variables, guard formulas and action functions.

FSC *Full Synchronous Compositions*, all the transitions guards that enables a specific event must be true for the transition to be enabled.

IEC *International Electrotechnical Commission*, an international commission that develops standards within electronics and electrotechnology.

ODE *Open Dynamics Engine*, a high performance library for simulating rigid body dynamics.

PERT *Program Evaluation and Review Technique*, a tool for project planning.

PLC *Programmable Logic Controller*, computer which is used to control an industrial process. The computer is often designed to connect to a large number of I/O units.

SFC *Sequential Function Chart*, a standard for visualization of PLC-code.

SME *Small and Medium Enterprises*, companies whose headcount or turnover falls below certain limits.

SOP *Sequence of Operations*, a graphical programming language within Sequence Planner. Similar to SFC.

V-REP *Virtual Robot Experimentation Platform*, a 3D robot simulator based on a distributed control architecture.

XML *eXtensible Markup Language*, Open and expandable file format.

Chapter 1

Introduction

This is a introduction for the master thesis work Planning and Development Tools for Automation System Design performed at Teamster AB and Chalmers University of Technology in Gothenburg.

1.1 Background

When designing an automation system the aim is to make the development process as easy and accurate as possible. Defining in what sequence different tasks should be executed is a problem when projects get comprehensive. Another problem is that late modifications in the development phase of a system could cause major changes in the sequence of operations. Together with the demand on flexible manufacturing, due to the fast development of product generations, makes the complexity of the control functions to increase. This is important issues to address for the industry of today (Lennartson et al., 2010).

The company Teamster AB in Gothenburg, Sweden, has been using Google's software SketchUp in order to build 3D models of industry units. These 3D models are made in an early stage of a project. The models are mainly used in order to develop and demonstrating prototype solutions for costumers in a manner that both engineers and operators can understand. By using Google SketchUp instead of conventional solid body *Computer Aided Design* (CAD) softwares Teamster AB is able to develop prototype 3D models faster and easier.

Since the usage of Google SketchUp has proven good results the aim is to expand this concept. Next step is to expand this concept with the ability to simulate automation units and to generate *Programmable Logic Controller*

(PLC) structure early in a project.

The software *Virtual Robot Experimentation Platform* (V-REP) which has been developed by Dr. Marc Freese, can be used as a simulation tool. V-REP makes it possible to create 3D simulations quickly. This property makes it a good candidate to use in order to simulate automation units with 3D models built in Google SketchUp. Other simulation softwares often demands a solid body CAD model as a base, which is time consuming to produce.

In order to generate the main PLC structure for an industry unit, the software Sequence Planner could possibly be an efficient tool to use (Lennartson et al., 2010). When the main PLC code structure is generated some control function would be preferred to guarantee that the generated code do not contain any errors. One possible alternative for this task is the formal verification and synthesis tool Supremica (Åkesson et al., 2006).

The overall goal with this thesis is to evaluate the possibility to use Google SketchUp for building a static 3D-model of the factory, then using V-REP for extending the static model with dynamic movements and control logic, where the control logic is generated semi-automatically generated from Sequence Planner. If this tool chain of low-cost tools can be used to develop solutions quickly then it will have an positive impact on the development procedure for Small and Medium size Enterprises (SME).

1.2 Purpose

This master thesis purpose is to evaluate how the above mentioned software tools can be used in an automation project. More precisely in the planning and development work of an automation project. The main goal is to produce a simulation in V-REP with a Google SketchUp model as the base and evaluate if Sequence Planner could be used to generate PLC structures in an early stage of an automation project.

In the investigation of the softwares an investigation on different conveyor solution in V-REP will be done in order to establish pros and cons with different solutions. This is done in order to understand what solutions works with different simulation setups. V-REP have a physical engine implemented which can simulate physical properties as for example gravity and collision forces. The benefits of having a fully implemented physical engine in a simulation software will be investigated.

1.3 Aim

Initially the aims in this master thesis were the following:

- Evaluate how Google SketchUp and V-REP could be used together in order to simulate automation systems in an efficient way.
- Investigate possibilities to import PLC code from Sequence Planner to V-REP.
- Investigate if PLC structure generated in Sequence Planner can be verified using Supremica.
- Evaluate if and how Teamster AB could use the mentioned software tools in a real automation project.

When evaluating Sequence Planner it was clear that Sequence Planner was far from a finished product.. Because of this early stage of development the master thesis aims were changed.

The final aims of this master thesis are:

- Evaluate how Google SketchUp and V-REP could be used together in order to simulate automation systems in an efficient way.
- Compare and evaluate different conveyor solutions in V-REP.
- Evaluate the benefits of having a physical engine when simulating.
- Determine what is missing in Sequence Planner.
- Compare functionality of the SP method (Sequence Planner method) with other methods.
- Evaluate if and how Teamster AB could use the mentioned software tools in a real automation project.

1.4 Limitations

- Softwares that will be studied and used are Google SketchUp, V-REP and Sequence Planner.

1.5 Method

At first a study of several different software tools will be performed in order to gain knowledge in how they work and how they can be combined to work as a unit. In this study; Google SketchUp, V-REP and Sequence Planner will be included.

When testing and investigating different softwares, a real automation project at Teamster works as a test ground. Having a real project as a base for evaluating the softwares, will result in an understanding of how they can be used in a real project in the best manner.

When evaluating these softwares' problems and difficulties will appear. Problems and difficulties that appear throughout the project are to be solved in the best fashion with the tools available.

After studying all the softwares that are used in this project, focus will be on using Google SketchUp and V-REP. This is done in order to see how they can be used together and could be applied on a real project in an efficient way.

1.6 Report structure

In this introduction chapter the thesis is presented and the problems are described. In chapter a real automation project is presented. this project have been used to evaluate different software tools. Chapter three and four contains the evaluation of the softwares. In chapter five the results are presented.

Chapter 2

Description of the S*Solar industry cell

*This chapter describes the S*Solar automation project at Teamster AB. First the product that is being produced is presented. After the product have been presented the S*Solar cell and its functionality is described.*

2.1 The S*Solar project

S*Solar is a real automation project at Teamster AB. The project includes designing, building and implementing an industry cell that produces a new kind of solar panels. The solar panels are designed to convert heat from sunbeam to energy by leading water through the solar panels. They are designed as architectural building units in order to like windows when mounted on the facade. This makes it possible to cover whole facades of a building with the S*Solar panels and at the same time achieve an appealing appearance.

The S*Solar panels are composed by several different parts. In the S*Solar project some assembling of the panels have been done in advance so the automation cell designed at Teamster only have to glue the main two parts together and add some minor parts (bushes). The minor parts are assembled by the operator of the industry cell and is not considered in the automated part of the cell.

The two main parts of the solar panels that are glued together are the bottom part with the water system and the top part which is a glass pane. These S*Solar panels are to be produced in two different sizes.

2.2 The S*Solar cell

The automation cell is built up by several different actuators and sensors, see Table 2.1 and Figure 2.1. These actuators have to be coordinated and controlled in order for the cell to work and behave in a proper way.

Table 2.1: Actuators and sensors in the S*Solar cell

Name:	**Description:**	**Remark:**	**ID:**
Conveyor belt 1	Transporting the body to position for merging with top part.		1
Conveyor belt 2	Transporting finished product to position for buffer.		2
Conveyor	Transporting glass to position to be lift by robot.	Top part is standing almost vertical leaning against the upper section of the conveyor.	3
Robot	- Lifting glas - Movement under glue unit - Pressing glas on body.		4
Glue unit	Apply glue to glas.		5
Fixture 1	Fixate the body.		6
Fixture 2	Make sure the robot lifts the glass in the exact right way.	Non moving fixture. Make sure the top part is the exact right position by gravity.	7
Lift arm 1	Used by operator to put glass on conveyor.		8
Lift arm 2	Used by operator to remove finished product and place them in buffer.		9
Sensor body	Sensing when body is in position to be glued with top part.		10
Sensor glass	Sensing when glass is in position to be lift by robot.		11
Sensor panel	Sensing when finished product is on the end of Conveyor belt 2.		12
Sensor size 1	Determine the size of the body.		13
Sensor size 2	Determine the size of the glass.		14

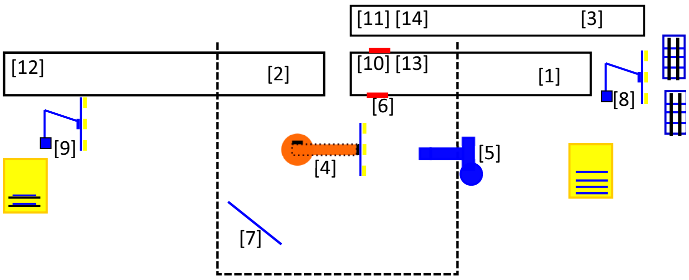

Figure 2.1: Description of the S*Solar cell.

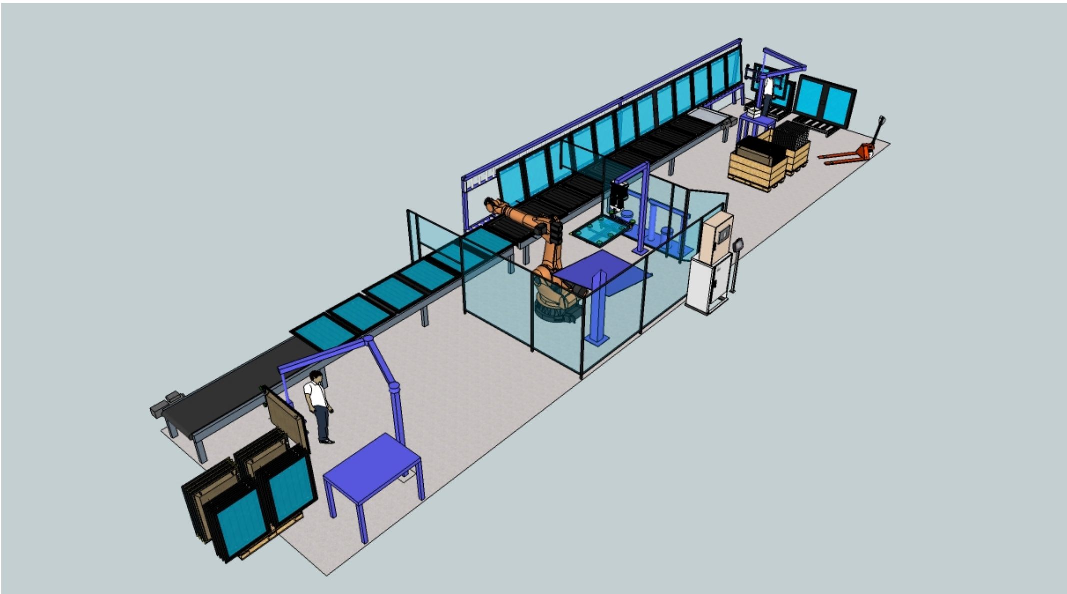

Figure 2.2: The S*Solar cell model built in Google SketchUp by Teamster AB.

The cell is a multi product assembly cell, this means that several products have to be considered by the logical control. In the cell an operator have to coordinate with the automatic part which makes the cell semi automatic. The layout of the cell can be seen in Figure 2.1 and Figure 2.2.

The operator puts the bottom part of the solar panel on the *Conveyor belt 1* and assemble the bushings to the bottom part. Then the operator uses the *Lift arm 1* to lift and place the top part (glass pane) in an almost vertical position on the *Conveyor* which transports the glass into the cell. These two procedures can be done in the opposite order. The operator can also fill the *Conveyor belt 1* with bottom parts and then fill the *Conveyor* with top parts.

The automation part of the cell runs the *Conveyor belt 1* until the *Sensor body* detects that a bottom part is in position. When the bottom part is in the right position the *Conveyor belt 1* stops and the *Fixture 1* fixates the bottom part.

The *Conveyor* runs until a *Sensor glass* detects that there is a top part in the right position which make the conveyor stop.

When a top part is in position the robot lifts the top part with vacuum suction. The robot moves the part to the opposite side of the cell and place the top part in the *Fixture 2*. After the robot have placed the top part in *Fixture 2* it once again lifts the part. The top part is placed in the *Fixture 2* in order to be sure that the robot lifts the part in a precise way.

After the robot have lifted the top part from *Fixture 2* it moves the part in a square motion under the glue gun which applies a precise amount of glue around the edges of the top part. Before this is done the system controls that there is a bottom part fixated on *Conveyor belt 1*. This is done in order to control that no glue is applied on the top part before there is bottom part to be glued on top of.

When glue have been applied on the edges of the top part the robot moves the top part to an exact position over the bottom part and presses the top part toward the fixated bottom part. After the two parts have been merged to a single product, the *Fixture 1* releases the product and *Conveyor belt 1* and *Conveyor belt 2* starts running if the *Sensor panel* does not detect any finished product lying on the end of *Conveyor belt 2*.

When the operator have loaded the cell with top parts and bottom parts he or she goes to the other end of the automation cell and uses *Lift arm 2* in order to remove finished products and place them in a buffer. A Gantt representation of the cell can be seen in Figure 2.3.

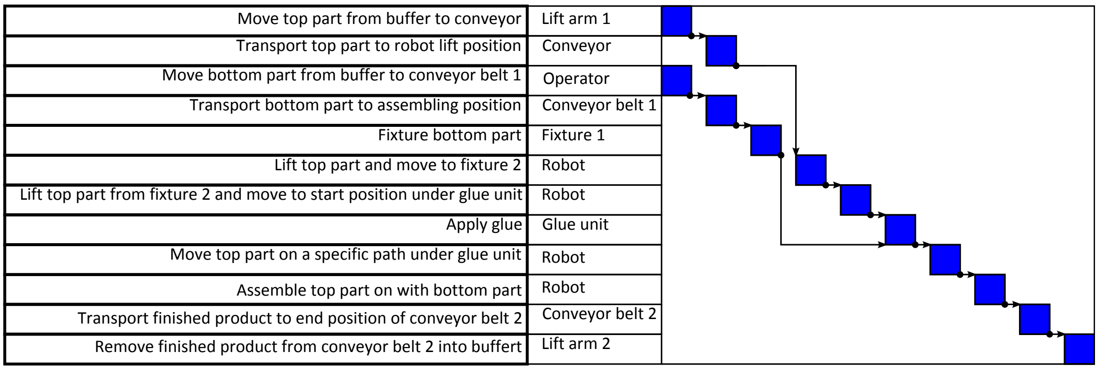

Figure 2.3: Gantt chart description of the industry cell.

Chapter 3

Evaluation of Sequence Planner

*This chapter starts by describing how Sequence Planner works and how it might be used to model using two different approaches; Product and Resource focused Sequence of Operations (SOP). After that an description of how the S*Solar project have been built up in Sequence Planner. This chapter then describes what is missing and needed in Sequence Planner in order to be able to represent the S*Solar project in a good way. A comparison between the SP method and alternative methods for sequence planning follows after that. Finally a chapter on how Sequence Planner can be used at Teamster is presented.*

3.1 Sequence Planner

Sequence Planner is a software that is currently being developed at different universities, mainly by the Automation research group at Chalmers University of Technology. The purpose of Sequence Planner is to facilitate the work with developing production systems. The expectation is to combine the information management with sequence planning while at the same time decrease the gap between production systems and logic control that exists today (Bengtsson et al., 2009). The link between the sequence planning and the logic control is the main advantage with Sequence Planner and the SP method aims to complement for example Gantt, described in Section 3.5.

Sequence Planner links the development of a product with the development of a production system. The aim is to develop a method where a product and a process are developed close together, and in this way achieve products of high quality to a low cost. The key is according to the research group; the information management. Due to this a large part of the method is about packing the information connected to a product in a structured way. This is hoped to facilitate the information transfer to other parts of the project. The aim is that the information shall be used in the development of the logic control.

The idea with Sequence Planner is to generate SOPs. By collecting information about process operations and product properties, such as demands and preconditions for operations, a set of SOPs is generated.

Information from product, process and automation are interconnected by operations and SOPs. Based on liaisons and precedence relations a formal relation between product properties and process operations can be achieved. A liaison is a way of describing merging of two objects or describing a new position or orientation of an object (Lennartson et al., 2010). From the precedence relations, preconditions can be generated for the related process operations. Together with other manufacturing information, demands and resource booking for example, the preconditions are used to auto generate the SOPs.

Generally the SOPs are on a high level at start. To achieve control functions and a complete view of the system, lower level SOPs can be implemented in the higher level operations. Sensors, actuators, alarm functions, manual control and human-machine interface can then be added.

In Sequence Planner liaisons is a way of describing the relationship for how one product is mounted on another product or to describe the position and orientation of a product. “A liaison defines an interface between different entities in a product. The interface may include connection, location and orientation” (Bengtsson, 2009).

The generated SOPs then needs to be tested for deadlocks, collisions and correct logical behavior. Optimization result in the final SOPs. In order to test and verify the SOPs and avoid possible problems the SOPs are converted into *Extended Finite Automata* (EFA) (Skoldstam et al., 2007).

3.1.1 Resource or product focused SOP

When modelling the system two different views are possible, product focused SOP or resource focused SOP. Both describes the same system behavior but has different benefits. In the product view the SOPs is represented as a sequence for each main product part in the assemble and is independent of resources. This view shows the flow of the parts in the cell and gives information about which operations that are needed. Information about how the resources should be configured to be able to perform the operations can also be achieved. This information can be used as the base when constructing a resource focused SOP. The product view is often more intuitive to model then the resource view and is therefore recommended to start with.

In a resource focused SOP there is a sequence for each resource in the cell. The operations in the resource view is the same as in the product view except for non-product related operations that may need to be added to complete the model. The view shows the behavior of the individual resources in the cell (Lennartson et al., 2010).

3.2 Implementation of the S*Solar project in Sequence Planner

Figure 3.1-3.3 describes the S*Solar manufacturing cell as a product focused SOP model built in Sequence Planner, the whole model can be seen in Appendix A. The S*Solar cell is described in Chapter 2. The model consist of two SOPs, one for the glass part of the product and one for the body. This model do not consider the fact that the product is produced in two different sizes. If size variation of the product were to be considered it would just add two more SOPs that describes body and glass with another size. These SOPs would be almost identical to the existing SOPs since there is no difference in manufacturing of the different sized products. The operations would be the same but they would differ in positions and movements, for example where the robot grips the glass. Another possibility is to add some operations to the existing SOPs, and with variables decide which path in the SOP that should be executed.

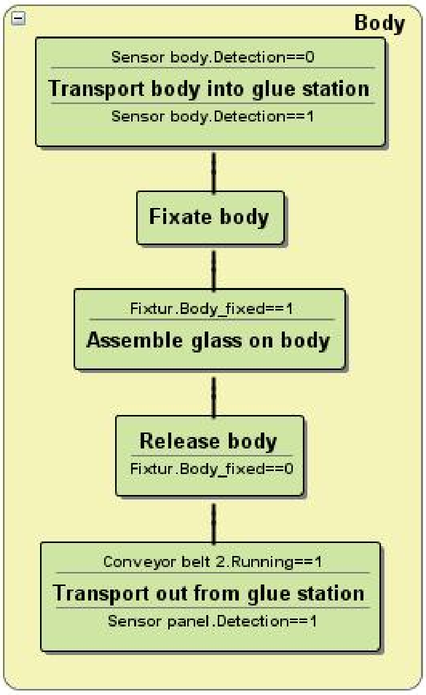

Figure 3.1: SOP for the body part.

In Figure 3.1 the SOP of the body part in the S*Solar cell can be seen. In order for the operation Transport body into glue station to be executed the *Sensor body* can not detect any body in the *Glue station*. The body part will be in this operation until *Sensor body* detects the body in the *Glue station*.

After the transport to the glue station the body is fixed and stay that way until the a glass is mounted on top on it and the finished product is transported to the end position of *Conveyor belt 2*. In order to transport the finished product from the *Glue station* the *Conveyor belt 2* have to be running. This is because the finished product is transported from *Conveyor belt 1* over to *Conveyor belt 2*. This operation ends when the finished product reaches the end position of *Conveyor belt 2* and *Sensor panel* detects the product.

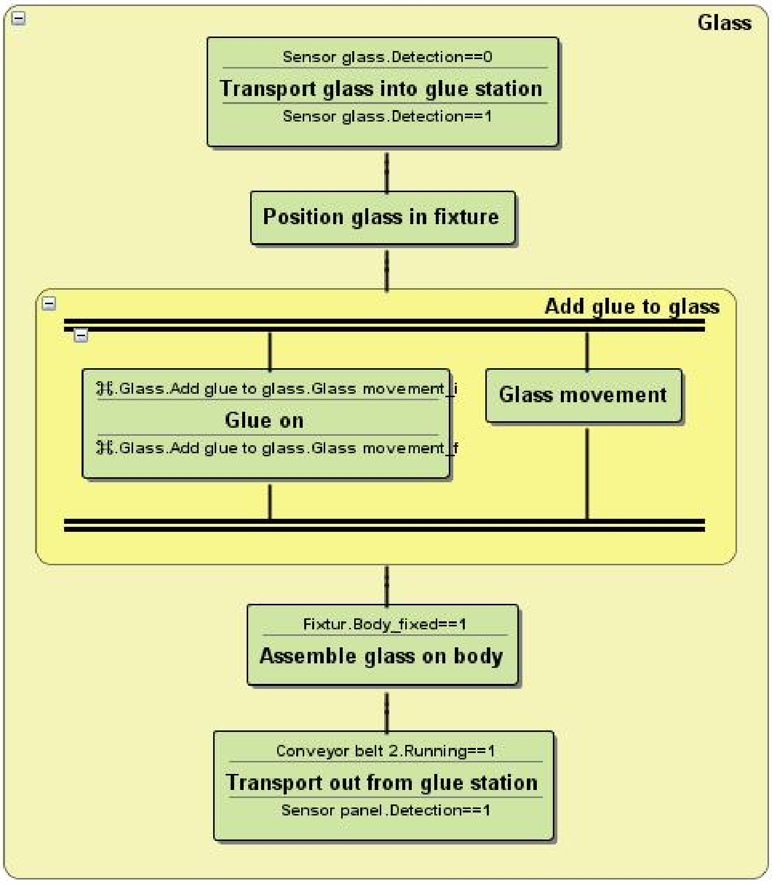

Figure 3.2: SOP for the glass part.

The glass part of the solar panel is transported into the *Glue station* if *Sensor glass* do not detect any glass in the glue position. If *Sensor glass* detects a glass the *Conveyor* is not running and the glass panel will not be transported, since this will cause the glass part in the *Glue station* to fall of the *Conveyor*.

In Figure 3.3 the conditions to start the SOP Add glue to glass can be seen. These preconditions is not included in the Figure 3.2 because in Sequence Planner the SOP have to be closed in order for conditions to show on the screen. The precondition ensures that there is a body part fixed on *Conveyor belt 1* and that the *Glue unit* is ready.

In the SOP, Add glue to glass in Figure 3.2 the glass is moved along a given path under the *glue unit* while glue is applied to the glass.

Figure 3.3: SOP for Add glue to glass in closed appearance.

The model in Figure 3.4-3.9 is a resource focused SOP over the S*Solar manufacturing cell, the whole model can be seen in Appendix A. The cell is divided into six resources; three conveyors, one fixture, one robot and one glue station. The lifting arms are not considered because of the fact that these are just helping tools for the operator to use.

In the cell two different types of panels are mounted, normal sized and large sized panels. This affects mainly the robot with respect to pick up and drop down positions together with glue movement. Therefore the robot has two different sequences to step through depending on the size of the panel.

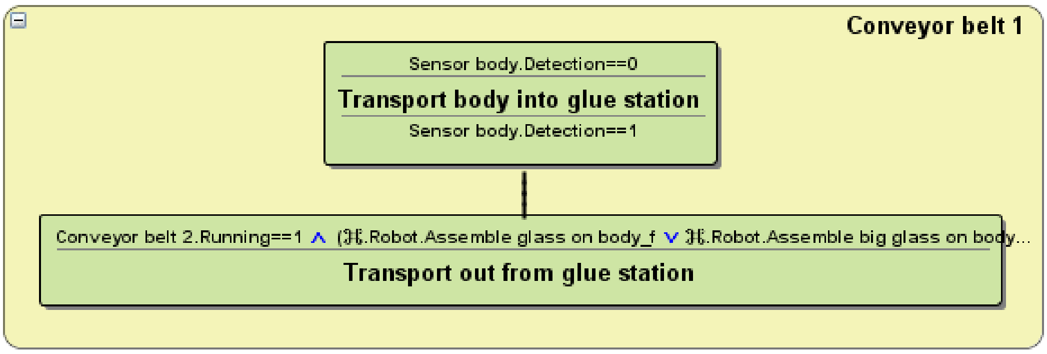

Figure 3.4: SOP for Conveyor belt 1.

Conveyor belt 1 transports body parts into the glue station continuously except when a body is in position to be assembled with the glass. This is signaled by *Sensor body*. When the gluing is performed *Conveyor belt 1* transports the panel out from the station over to *Conveyor belt 2* if *Conveyor belt 2* is running.

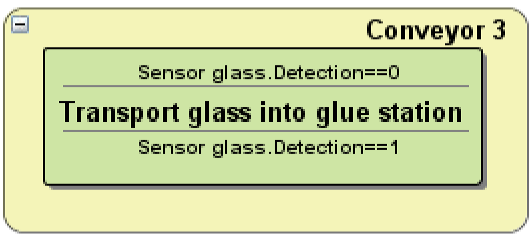

Figure 3.5: SOP for Conveyor 3.

Conveyor 3 transports glasses into the pick up position continuously except when a glass is waiting to be picked up by the *Robot*. *Sensor glass* signals detection when the glass is in position.

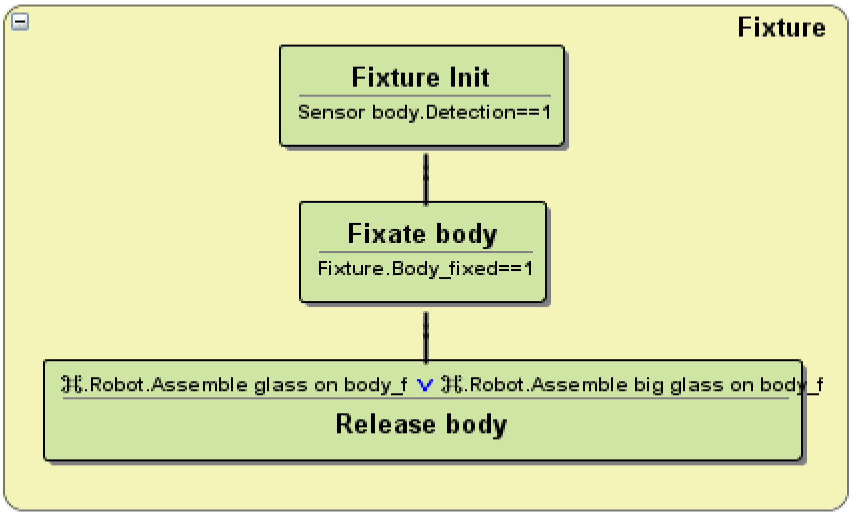

Figure 3.6: SOP for the fixture.

When the body is in position for the assemblage, controlled by *Sensor body*, the *Fixture* can start the fixation operation. Then when the *Robot* has pressed the glass on top of the body the *Fixture* releases the finished panel.

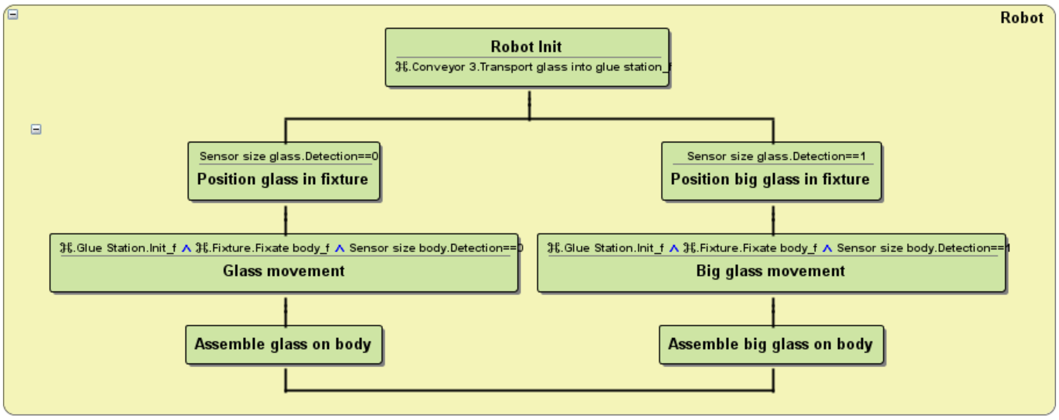

Figure 3.7: SOP for the robot.

In the operation Robot Init the *Robot* waits for a glass to be in position for pick up. Depending on the size of the glass the *Robot* runs through two different sequences, which is decided by the information from *Sensor size glass*. During the Glass movement operations the *Glue station* applies the glue on the glass. Therefore the *Glue station* must be ready before this operation can start. Both glass and body must have the same size and the body must be fixed in the *Fixture* for the SOP to continue. The body size is controlled by *Sensor size body*.

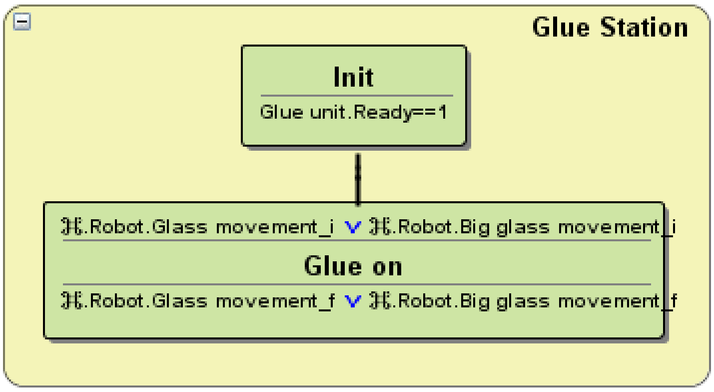

Figure 3.8: SOP for the glue station.

During the operation Init of the *Glue station* the glue is warmed up to the right temperature and the gluing is prepared. When one of the Glass movement operations starts the *Glue station* starts to apply glue on the glass until the Glass movement is completed.

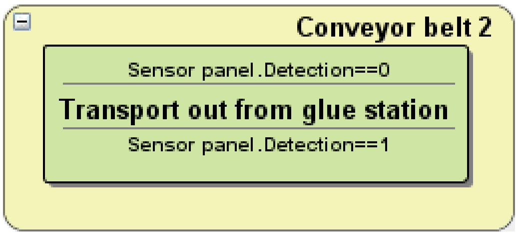

Figure 3.9: SOP for conveyor belt 2.

Conveyor belt 2 finally transports the finished panels out from the *Glue station*. It only stops when a panel is waiting to be unloaded by the operator at the end of the conveyor.

3.3 Missing features and problems in Sequence Planner

In Sequence Planner models of manufacturing systems can be made in an convenient way. The model gives an overview of the system that is easy to follow and the logic is implemented in a simple way.

In order to be able to use Sequence Planner in an efficient way several missing functions have to be added and developed. Sequence Planner is currently under development and for that reason contains some bugs that needs to be corrected. One of the major bugs that needs to be corrected is the saving ability of the program. During our master thesis saved data have been lost at two occasions.

Variables are available in Sequence Planner. When a variable is used to set up a condition and the project is saved and reopened the condition is manipulated. Because of this variables can not be used as conditions in Sequence Planner.

Liaisons are available in Sequence Planner but can not be used in any way. This function would have been useful in our product focused SOP, Figure 3.1-3.2. Liaisons could have been useful in both describing position and orientation of the glass and body as well as the merging of the parts.

Operations that are carried out on two assembled objects from a product modeling perspective should have the same operation id (Lennartson et al., 2010). An operation that is carried out on two assembled objects is of course carried out on both of the original separate objects. A small example is illustrated in Figure 3.10, where OP[1-5] is operations on two different objects. In OP[3-4] the objects are assembled to one new object, so the final operation OP[5] that is carried out on the assembled object is the same OP on the two different objects.

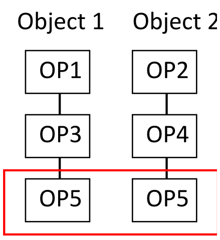

Figure 3.10: Example where two SOPs share a common operation.

This feature of having the same OP carried out on different merged objects is not available in Sequence Planner today. This is a needed feature in order to remake our product focused SOP in a more realistic way. The operation Transport out from glue station, see Figure 3.1-3.2 is a operation that is performed on the assembled panel. In our model this operation appears to be the same since the OP name is the same, but the OP id numbers in the system differs.

By implementing the functionality of classes and subclasses Sequence Planner would gain functionality. For example in our recourses focused SOP, the product could be divided into class and subclasses, see Figure 3.11.

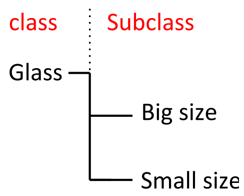

Figure 3.11: Example of classes and subclasses.

If the functionality of classes and subclasses were available the possibility to set up demands and condition for product types would be possible. Some operations are not depending on the product type, for example whether it is a big or a small glass. Instead of having a condition for each product just one condition would be needed for the class glass. If you have several robots in a class it would be possible to call the class and the first available robot in it could execute the requested operation.

3.4 Future usage of Sequence Planner

The main idea with Sequence Planner is that the user can define a set of conditions and demands that needs to be executed before a certain action is taking place and that the SOPs are generated. Only the necessary demands needs to be specified, this creates a flexibility in the sequence in which the operations should be performed.

Industrial units are often complicated which makes it hard to understand in what sequence operations must take place. Instead of defining SOPs by looking at the whole system, the programmer will look at single operations. By doing this the complex system is broken down into smaller pieces that is manageable. For each operation the programmer defines a set of conditions and demands that needs to be fulfilled for the operation to take place (Bengtsson, 2009).

When the conditions and demands have been defined, Sequence Planner will generate SOPs that is later translated into EFA (Lennartson et al., 2010). The EFA may then translated into PLCopen XML code that can be translated to a vendor specific PLC code (Chao, 2009). This translation from demands to vendor specific PLC code is described in Figure 3.12.

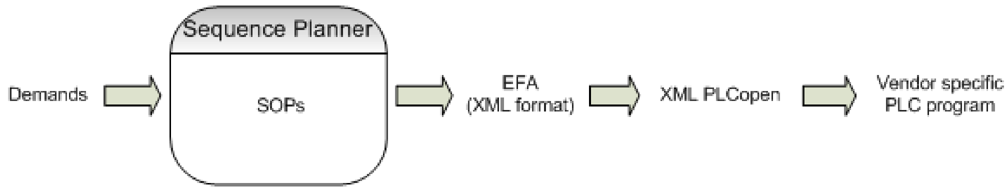

Figure 3.12: Translation from demands to vendor specific PLC code.

XML

The definition of *eXtensible Markup Language* (XML) is; “A markup language is a mechanism to identify structures in a document. The XML specification defines a standard way to add markup to documents” (Walsh, 2011). XML is an extendable language which can be used to exchange data between different information systems. For example it can be used as a database of robots and other actuators in an industry cell (Walsh, 2011).

PLCopen XML

PLCopen is an organization that aims towards making it possible to transfer projects, programs and libraries between different development environments. In order to accomplished this the workgroup named TC6 was founded. This workgroup works with XML to realizing PLCopens aims.

The International standard *International Electrotechnical Commission* (IEC) 61131 is a standard for programmable controllers (PLC) and everything attached to this, such as programming and debugging. This is today the only existing global standard for industrial control programming. The third section of this standard (IEC 61131-3) is the base for PLCopen XML. This section of the standard defines the semantic and syntax of PLC programming (PLCopen, 2011).

3.5 Alternatives to the SP method

Below three alternative methods to the SP method are presented. These can be used for project and sequence planning.

PERT

Program Evaluation and Review Technique (PERT) is a method used for analyzing the parts of a project and to minimize the time needed for the whole project. To be able to use PERT the tasks of the project has to be defined, in which order they have to be executed and estimate how long time each task will take (Malcolm et al., 1960).

In Figure 3.13 an example of a PERT diagram is shown. The circles (10-50) are milestones which indicates when an operation can start or should be finished. They also represent a certain state in the process. The operations are represented by the arrows (A-F) which leads to new states in the process. The diagram gives a clear picture of how long the different tasks of the process takes and, which operation that leads to which milestone and when the moments should be finished.

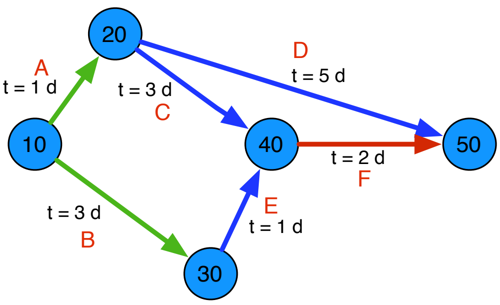

Figure 3.13: Example of a PERT diagram.

PERT is mainly used for project planning, but could also be used for sequence planning. Instead of milestones the circles can symbolize operations in a process and the arrows can represent actions and in which order they shall be executed.

When projects gets big and comprehensive PERT have a ability to be unclear and unwieldy. The same thing should happen when planning a production system. Moreover the method does not include information about which resources a certain operation allocates, just how long time the operation is supposed to take. The resource and zone booking must therefore be implemented in some other way. There is no way of defining a complete logic model in PERT

Gantt

A Gantt chart is in many ways similar to PERT. It is used for structuring a project into smaller tasks, make an assumption of time needed for each task and in which order they need to be performed. What differs from PERT is the way the project is illustrated. In a Gantt chart each operation is visualized by a horizontal pole. Its position and length shows when it should start and how long time it should take (Wilson, 2003). In Figure 3.14 the same process example as in previous section but according to the Gantt model.

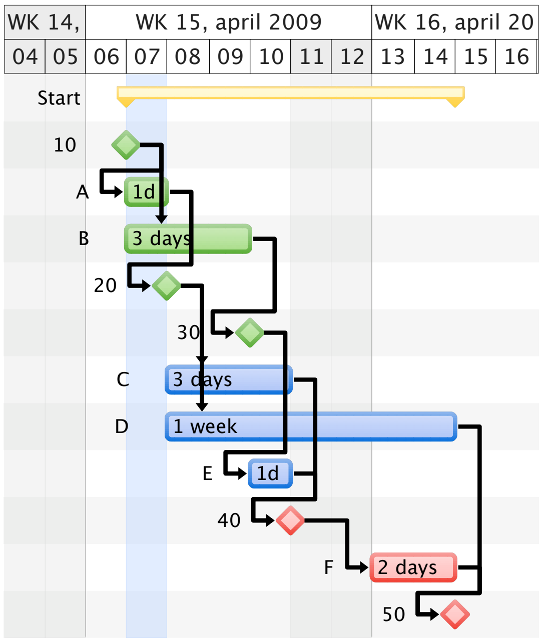

Figure 3.14: Example on a Gantt chart.

As PERT charts Gantt chart often gets unclear when a project increases in size. The fact that the method is developed for projects with humans as resources may lead to non efficient use when planning a production system. Demands and conditions can not be included in a Gantt chart as it can be in the SP method. The only possible conditions available in Gantt is what sequence the operations have to be executed. In neither gantt or PERT charts the possibility to represent alternative or arbitrary relations between operations is available. This is possible in the SP method.

SFC

Sequential Function Charts (SFC) is one of the most important of programming languages for PLC stated in IEC 61131-3, the standard for PLC programming (Fabian, 2006). SFC is a graphical tool for describing sequences for PLC programming, and is used to increase the understanding of the code. SFC is built on the graphical language Grafcet which was developed in France during the 1970's. In 1988 the language was adopted to the IEC standard (Johnsson and Årzén, 1999).

The code in SFC is divided into steps and transitions. The transitions consists of links and conditions. To each step there is an action connected, and a step can be active or inactive. The action in only executed in the active state. Several steps can be active at the same time by building parallel sequences. This differs SFC from other visual programming techniques (Fabian, 2006). In Figure 3.15 an example of a SFC diagram is shown. The steps S11 and S21 can be active at the same time, but S3 and S4 can not. This facilitates the understanding of the code, but it also means that the status of the program in a certain moment can come in many different ways. The rectangles visualizes the steps and the straight lines between the steps visualizes the links. The short horizontal lines is the transitions conditions which acts like barrings that only allows the execution when the condition if fulfilled (Halang and Frigeri, 1998).

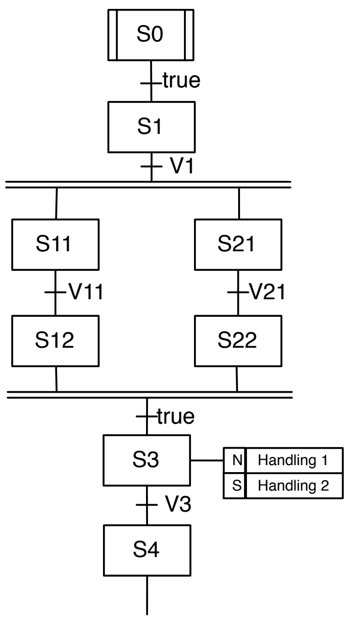

Figure 3.15: Example on a SFC diagram.

SFC is widely used which suggests that the method is a powerful tool for graphical visualization of a process and its PLC code. The disadvantage is that the sequences must be planned manually and that there is no given way of optimizing it. Thus, the method does not help with the work of finding the optimal resource to a certain operation, nor do Gantt and PERT. This is a functionality that is included in the SP method.

Due to that big programs, eventually with both sequential and parallel operations, often gets hard to survey even if SFC is used it is hard to avoid deadlocks in the code. This means that the code gets stuck in a combination of states where it cannot further execute. The reason is often that two entities are waiting for each other. If this is not detected before startup of the production system the system will most likely not function as planned and a lot of time will be wasted when the code must be reworked.

Chapter 4

Evaluation of V-REP

*This chapter starts by describing the software V-REP and its features. Further there is descriptions of how the S*Solar model from SketchUp is implemented and different solutions on how the simulation could be build up. Problems with the implementation is also included together with a part about how V-REP can be used at Teamster.*

4.1 V-REP

V-REP is a 3D robot simulator where whole robotic systems can be simulated. This is because V-REP uses a distributed control architecture where the objects in the scenes can get scripts directly attached to them, which corresponds with how systems are build up in reality. The scripts run simultaneously either in threaded or non threaded fashion which makes V-REP perfect for multi-robot applications (Freese, 2011a).

V-REP offers an amount of functionalities which are very adaptive. This together with the architecture makes V-REP flexible and the sector of application is wide. For example factory automation simulations and product presentations which are relevant for this report, but also for safety double-checking, for fast prototyping and verification, hardware control and for robotics-related education to name a few (Freese, 2011a).

4.1.1 Features of V-REP

V-REP includes an amount of features which can be seen in Table 4.1 where the most important features for this master thesis is compiled. Below some features are described more in detail.

Dynamic engines of V-REP

V-REP uses two different physical engines. These physical engines are Bullet and *Open Dynamics Engine* (ODE) (Freese, 2011a). The user of V-REP have the ability to choose which engine to use for a specific simulation. Both physical engines are powerful but have strengths and weaknesses that differs between them. These strengths and weaknesses make them vary in performance depending on the simulation setup.

ODE is created by Russell Smith and is an open source, high performance library for simulating rigid body dynamics (Smith, 2007). Among others it includes advanced joint types and integrated collision detection with friction. It is used in computer games, 3D authoring tools and simulation tools where it can be used for simulating vehicles, virtual creatures and objects in virtual reality environments.

Bullet is a Collision Detection and Rigid Dynamics Library and is like ODE open source and free for commercial use. It is mainly used for game physics simulations (Bullet, 2011).

Table 4.1: Features in V-REP (Freese, 2011a).

Feature:	**Description:**
Distributed Control	Unlimited number of simultaneously operating scripts, both threaded and non-threaded.
Compact and lightweight	6 MB application, the model is saved into one compressed file.
Powerful Application Programming Interface (API)	Over 300 clean, fine-grained and well-documented API functions.
Inverse/forward kinematics	Calculations for any type of mechanism.
Dynamics/Physics	Possible to simulate real-world physics and object interaction.
Collision detection	Fast interference checking between any meshes.
Cutting simulation	Simulation of surface cutting operations with different customizable tools.
Sensor simulation (Proximity, Visual type, Force/Torque)	Customizable, realistic and exact sensors.
Path planning	Holomonic or non holomonic path planning. Also custom algorithms are supported.
Easy to import/export CAD	Supported formats: DXF, 3DS, OBJ, STL
Fully customizable	Possible to write own plugins and applications
Tutorials	A number of tutorials included in V-REP which gives an introduction to the software and how to work with it

4.1.2 Companies using V-REP

K-Team, is a company that is developing, manufacturing and selling mobile robots for research and education, is a reseller and partner of V-REP (Schneider-Vidi, 2011). In the same way as Teamster uses Google SketchUp to demonstrate their solutions for costumers, K-Team uses V-REP to demonstrate their robots, but also as a test bench. The potential customers can then test the robots in the software before buying to make sure that they get satisfied with their purchase. V-REP is also used as an accurate simulator to use in the development of their products.

K-Team found V-REP to be a very good simulating software for their robots and the software is now supporting all of K-Team robots. In the future K-Team believe that V-REP can compete with other more expensive and less performing alternatives. This due to the low price and all of it features, for example path-planning and inverse kinematics, that only can be achieved by buying costly add-ons for other simulators. With V-REP almost any situation can be simulated and it can be used for education, research and industrial purpose. This flexibility, quick adaptivity and Dr. Marc Freese support makes V-REP applicably and for K-Team V-REP is a good sales argument (Schneider-Vidi, 2011).

4.1.3 Development of V-REP

Throughout this master thesis an open dialog with the developer of V-REP Dr. Marc Freese has been conducted. A lot of important information to our simulation has been achieved and Dr. Freese has received suggestions and feedback on V-REP. Some of the features implemented after our discussions can be seen in the list below.

- Creation of a conveyor without moving parts.
- Reworked manual to a web based version with the possibility to search in it.
- Ability to have several code windows opened at the same time.

4.2 Implementation of the S*Solar project in V-REP

In the creation of a simulation in V-REP of the Google SketchUp model of the S*Solar cell different approaches have been tested to find the best solutions. When importing Google SketchUp models into V-REP some problems occurs which are described in Appendix B.

4.2.1 Evaluation of different conveyor solutions

A big problem was to find a solution for how the conveyors should be modeled and function. The program code for the simulation must be adjusted for how the conveyors act, which makes the conveyors to a central role in the simulation. Below is an evaluation of three alternatives described with pros and cons. All these solutions have been tested in V-REP and are working.

Conveyor belt built up by dynamic parts

By connecting several dynamic objects that follows each other on a path a conveyor belt can be constructed, as in Figure 4.1. The individual dynamic objects have physical properties that make it possible for other dynamic objects to follow the conveyor because of the physical engine that exists in V-REP.

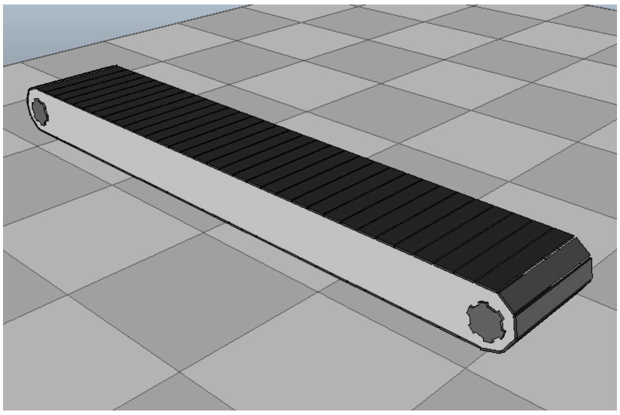

Figure 4.1: Conveyor with dynamic objects.

This solution makes the simulation behavior and the program code realistic. It also looks realistic when the parts of the conveyor actually moves. The problem is that all physical objects of the conveyor demands a lot of calculations and makes the simulation slow, especially when the conveyor are as long as in our model with over 400 objects in each conveyor. This way of doing conveyors is also time consuming due to the fact that there is no effective way to change the conveyors dimensions, each different conveyor must be build almost from scratch.

Conveyor belts by updating the objects position

Instead of making the conveyors move a solution is to make them static and add scripts that are responsible to move the objects that should be transported. Loops can update the position to simulate a movement. With this method the position of the object can be exact and with full control. The problem is that you need to write script code and it takes time to make it work due to the fact that the positions must be continuously controlled. Because of this the main program in this solution differs from the other two described in this section and it does not mirror the reality.

The dynamic properties in V-REP gets unused and a non-real behavior is achieved. This setup cannot be used if a real physical behavior should be simulated. With many objects moving in the simulation a high computer load is obtained since all objects being transported have their own code that runs in its own thread concurrently with the others.

Conveyors belts without dynamic parts

During our master thesis project a new feature was developed in V-REP by Dr. Marc Freese, a dynamic conveyor without dynamic parts which can be seen in Figure 4.2. The body of the conveyor is a rectangular parallelepiped which acts exactly as a *Conveyor belt built up by dynamic parts* which is described in 4.2.1. To the conveyor a script is added that simulates the performance of a real dynamic conveyor. Without the dynamic parts the computational load is reduced and it is also simple to convert the conveyor to the requested size. What can be seen as negative is that the movement of the conveyor can not be seen.

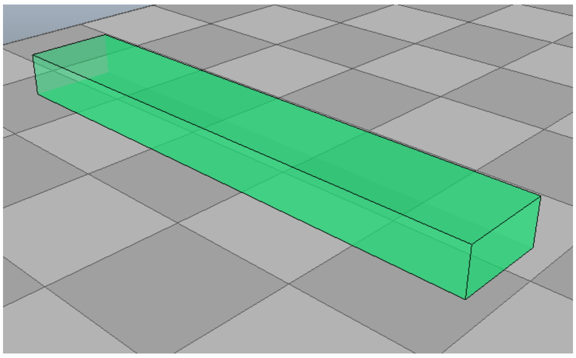

Figure 4.2: Conveyor without dynamic objects.

In V-REP there is a possibility to hide objects and work with different layers. This feature is advantageous especially for this solution where you can place the conveyor at the same position as the realistic looking model of the conveyor. Hide it and make it look like it is the model that transports the objects.

4.3 Effective use of V-REP

This section describes how to use V-REP in an efficient way.

4.3.1 Pure shapes

A polygon mesh is a collection of vertices, edges and faces that defines the shape of a polyhedral object in 3D computer graphics and solid modeling. To simplify the rendering the faces usually consist of triangles or other simple convex polygons (Caponetti and Fanelli, 1993).

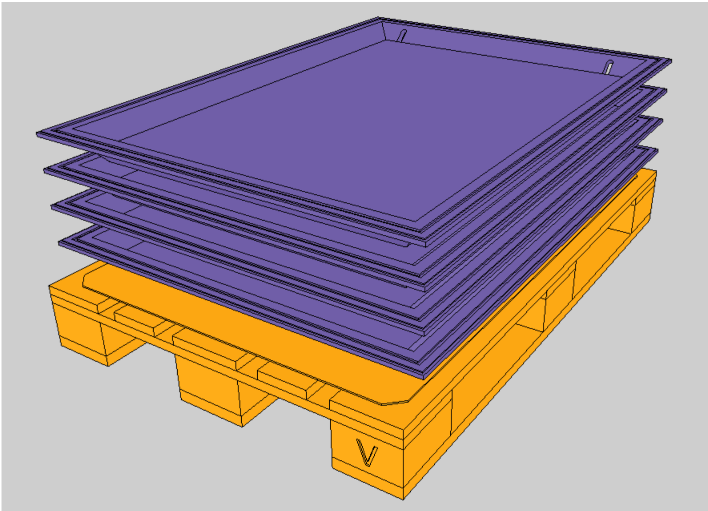

Figure 4.3: The body parts without pure shapes.

When importing a SketchUp model into V-REP the meshes of the objects in the scene are often complex and not stable, see Figure 4.3. This makes the calculations very slow and leads to a non-natural behavior of the dynamic parts in the simulations. This is because the physics engines included in V-REP, Bullet and ODE, do not perform well with these meshes (Freese, 2011b). Instead of using random meshes, which are extremely difficult to handle, pure shapes can be used as in Figure 4.4. Pure shapes are primitive shapes like rectangles, cylinders and spheres. With these shapes it is much simpler and faster to make calculations on contact points and reactions forces. Therefore it is recommended to use a pure shape for dynamic calculation. How to make pure shapes in V-REP is described in Appendix B.

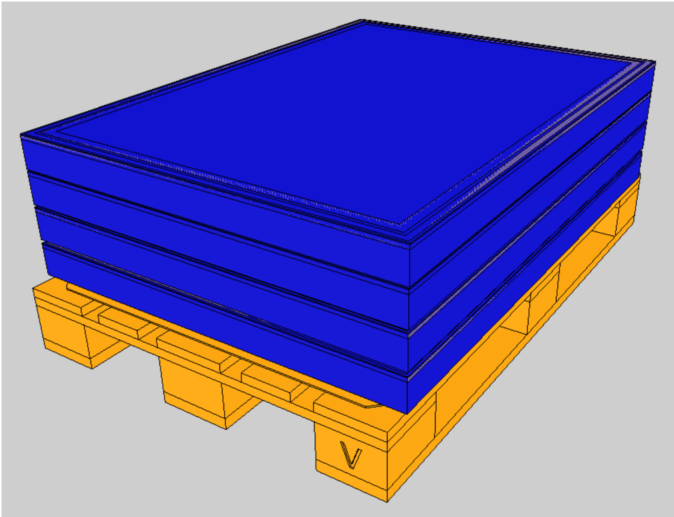

Figure 4.4: The body parts with pure shapes added to the body parts. This pure shapes are hidden when the simulation runs, but the physics calculation are made on them.

4.3.2 Module based use of V-REP

To save time it is a good idea to make the equipment you create in V-REP to objects, which then are placed in the library. These objects can be reused in other simulations and you do not need to do the same work again. For example when a robot is imported from SketchUp and all the joints are inserted and the settings are made you can save it and use it again.

Much of the code in the scripts are repeated and can be reused. Therefore a code structure can be saved with the objects stored in the library which makes the development of new scenes faster. In the S*Solar scene, targets are used when creating the movement of the robot and the lifting arms. In a new scene the targets can be exchanged, the movement properties can be configured and other behaviors can be customized.

4.3.3 Targets

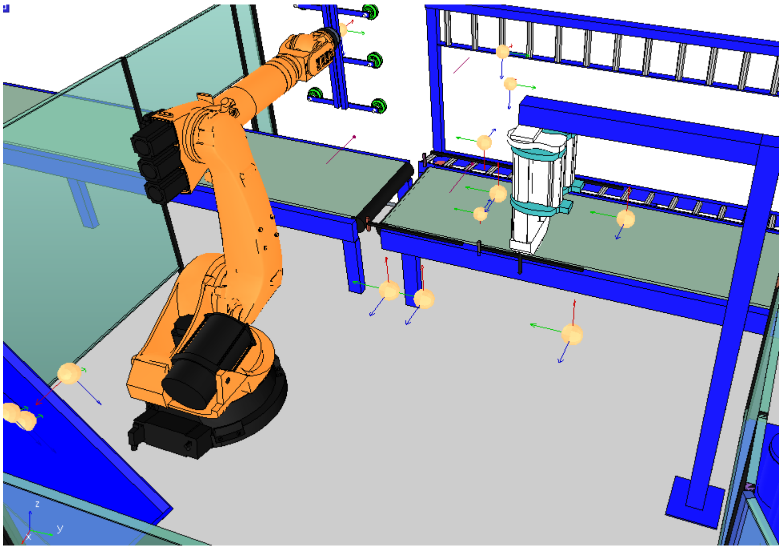

Figure 4.5: Targets for the robot positioning.

It is time consuming is to make the movements accurate. In the simulation of the S*Solar cell targets were used for positions and movements of the robot and lifting arms, see Figure 4.5. The placement of these targets is time consuming. It is easy to place the target in an approximate position by using the move and rotate tools in V-REP. The exact position and rotation can be achieved by entering numbers in the Coordinates/Transformations dialog as can be seen in Figure 4.6.

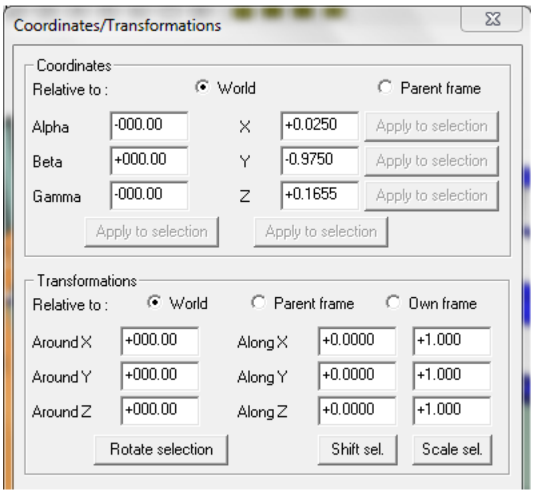

Figure 4.6: Positioning settings in V-REP.

By running the simulation the wanted position can be localized. This is however quite inefficient and takes long time. Usually several simulations must be done before the position is adjusted correctly. If the movements are in the end of the simulation a lot of time is spent on waiting for the part where the targets are used. But there are some tricks that make the process faster.

First of all it can be convenient to have the same model in another scene separated from the main scene. In this scene another program can be added so the interesting parts can be observed directly. There is also an manipulation tool to use with which the robot can be manipulated in the position desired (Freese, 2011b). The position can be noted and the target can be placed according to it.

In the simulation of the S*Solar cell all the panels runs throw the whole cell. By making copies of the panels at importent points in the simulation, it is possible to see the several sequences in the cell from start of the next simulation (Freese, 2011b).

4.4 Advantages with a physics engine

The two physics engines included in V-REP provides an approximate simulation of physical systems. These engines are mainly used in video games where the simulations are in real time, but can also be used for computer graphics and film. All physics engines have one main task, to solve the forward dynamics problem. This problem can simply be described as follows; What is the motion of a system, given the forces acting on the system (Boeing and Bräunl, 2007). In this section an evaluation of the advantages and disadvantages of having an physics engine in a simulation software is made. Areas where a physics engine can be used are also described.

One area where a physics engine could be used is in the development of robots. The software Gazebo is developed for simulating outdoor environments for robotic vehicles for outdoor applications. It is designed to reproduce the dynamic environments a robot may encounter in a accurately way. The simulated objects have for example mass, velocity and friction. This makes them behave in a realistic way when they are being pulled or pushed or in other ways affected (Koenig and Howard, 2004). Because of this the robots behavior can be tested in different environments, the development goes faster and to a lower cost.

Another area where a physics engine can be used is simulation of self- reconfigurable robots. A self-reconfigurable robot is a robot build up by a number of identical modules. These modules can manipulate each other and in that way the robot can change its own shape (Christensen et al., 2008). The physics engine allows dynamic interaction between the robot and with the simulated environment and effect as friction and gravity is added to the simulation. The self-reconfigurable robot can then adapt to the environment to be able to handle the situation it the best way. It can for example change it shape to be able to pass an obstacle like a wall. For the robot to be able to adjust as well as possible support can be added by combining physical shapes with sensors, actuators, communication devices and connectors.

A physics engine is suppose to make simulations as similar to the real world as possible. There is a number of engines that makes this, but still it has to be considered that it is a simulation and that the reality may differ. ODE is the most popular rigid-body dynamics implementation for robotics simulation applications, still it has shortcomings which affects the simulations in a negative way (Drumwright et al., 2010). A number of these areas are, lack of computational efficiency, poor support for practical joint-dampening and friction approximation via linearization.

While simulating an industrial cell as the S*Solar cell there are a number of advantages with the physics engine. As described in Chapter 4.2.1 the use of dynamic conveyors made the simulation realistic. Due to gravity and friction it is possible to observe how a product is falling down on the conveyor and then follows the conveyor with the same velocity. Besides realistic appearance a realistic underlying code is achieved. When the simulated actuators have the same physical behavior as the physical actuators, the simulation can be built easier since it is more intuitive to program the simulation and no extra code is needed to visualize physical behavior.

4.5 V-REP at Teamster AB

V-REP is a good alternative to make quality simulations of Google SketchUp models. It includes all the features needed for automation solutions and the software is continually updated with new objects and functions. To further illustrate how the proposed solution is supposed to work and make an even greater impression on the customer a simulation can be advantageous.

Making a simulation can often be time consuming and to a high cost for the user because of the large solid body modelling effort needed just to get started (Craig, 1997). The ability to import a Google SketchUp model in to V-REP makes this step convenient and fast. It is also possible to make objects in V-REP but for the S*Solar cell it has only been used for smaller changes in the SketchUp model.

V-REP can to some extent be used to make robot simulation with for example collision detection. But Teamster AB already have satisfying tools like KUKA Sim and ABB Robot Studio for these tasks. These softwares are developed by the robot manufactures which makes them custom made for their robots. In the simulation environment the robots behaves exactly as in reality and real robot code is written to define the robot movements. Even if there are many settings in V-REP it is hard to make it as accurate as in these specialized software tools. In V-REP the movements are written in LUA code which you cannot use to anything else than for simulation. Still V-REP can be used in an early stage of a project to control that the robot can reach to where it is suppose to reach and that the solution is possible.

A connection for the PLC program is something that had been useful for Teamster AB. At present their is no such connection due to the fact that the simulation is based on LUA code.

Chapter 5

Results

In this chapter the results of our work with Sequence Planner, V-REP and Google SketchUp is presented.

In the development process of manufacturing systems late modifications are costly and time consuming. To find errors caused by the human factor in an early stage virtual models can be used. A virtual model can be a static three dimensional description of the manufacturing system or a dynamic model that also describes the movements of robots and other equipment. The dynamic model should preferable include the control code to verify the correct behavior. Figure 5.1 describes the intended workflow of creating a dynamic model with the evaluated softwares.

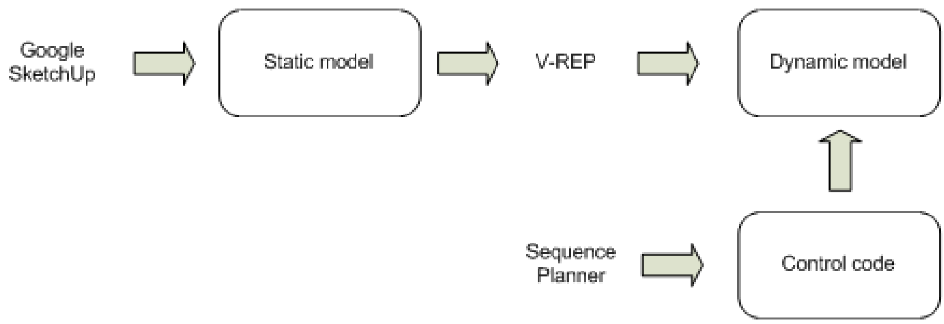

Figure 5.1: Intended workflow to create a dynamic model.

A manufacturing process consists of different operations. For large scale production systems it can be unmanageable to determine in which sequence the operations should be performed. The manufacturing industry demands flexible systems because of the short product life cycles of today. With the available methods today the generation of the sequences are made manually.

In order to build dynamic models V-REP have been evaluated as a candidate. Importing a Google SketchUp model into V-REP is convenient and the model has the same appearance as in Google SketchUp but with two exceptions, the grouping and the colors are not included. This means that this has to be done again in V-REP. Even though an investigation was done to rectify the problem no solution was found to this problem.

During the master thesis a simulation of a real automation project at Teamster was performed. The S*Solar cell is simulated in V-REP with a model from Google SketchUp as base. The simulation includes the whole intended product flow and the resources are modeled to perform as in reality. The movements of the robot are realistic and generated through the inverse kinematic feature included in V-REP. Further, the simulation also contains several conveyors and two lifting arms with operators attached to them. Because of the programmed movements of the lifting tools the operators are used to increase the understanding of the cell and how it is intended to work.

A physics engine makes a simulation very realistic in V-REP. With for example gravity and friction the object behaves as in the real world, and the appearance is impressive when for example an object falls down from a height or when objects are pulled or pushed by other objects. It can also be used for testing devices that acts according to the surrounding environment. Different kinds of robots can be tested in different simulated environment. This could result in faster development to a lower cost.

When simulating an industrial cell the benefits with a physics engine is, besides realistic behavior and appearance, that a realistic underlying code is achieved which makes it more intuitive to program the simulation. When the simulated actuators have the same physical behavior as the physical actuators, the simulation can be built easier since no extra code is needed to visualize physical behavior. An example is when modeling a conveyor with physical properties. Due to friction and gravity the objects that are placed on the moving conveyor will move with the same velocity as the conveyor. Therefore no code is needed to control the movements of the objects. The reduced need for code also makes the simulation more lightweight.

Three different conveyor solution was implemented in the simulation and then evaluated. They all have both advantages and disadvantages. The choice between them depends on what type of simulation that is wanted with respect to factors like computational load, appearance, control and realistic behavior.

Although V-REP is an easy software to build simulations in, it is time consuming and there is a lot that have to be done in order to produce a simulation. Over time the development of the simulation will go faster when the V-REP library gets filled with robots, lifting tools and other equipment. A lot of the code can be reused and knowledge about the program will also accelerate the process.

Teamster could mostly use the V-REP for demonstration of today but as the software develops we see a future to use V-REP as a simulation tool that might be able to replace other simulation tools. In order for V-REP to replace other softwares some functions needs to be added. One of the most important features is the ability to implement and test the control code.

There is almost no limitations on what can be modeled and simulated in V-REP. Teamster could therefore use V-REP to simulate and test mechatronic equipment. In a lot of there automation designs there is custom made mechatronic equipment. The company K-Team uses V-REP in order simulate and test there mobile robots.

To generate control code that can be implemented and verified in the dynamic simulation, Sequence Planner have been evaluated. In an convenient way models of manufacturing systems can be created and the logic is easily implemented. The model gives an good overview of the system and is easy to understand. But Sequence Planner is also missing several important features in order to work properly and be used as a sufficient tool to generate PLC code with. The most important things that needs to be corrected are:

- The auto generation of the SOPs must be implemented.
- The ability to work with variables need to be implemented in a better and working way.
- The liaison function needs to be completely implemented.
- The ability to convert the SOPs to PLC code or export the SOPs to a format that can be used for creating PLC code.
- Possibility to verify the models, for example for deadlocks.
- The ability to model multi product cells.

Due to the fact that Sequence Planner is under development a comparison between the SP method and other planning methods were made to investigate what Sequence Planner can offer in the future. With Gantt, PERT and SFC the programmer manually has to define the sequences of operations. Models from Gantt and PERT gets unclear when large systems are modeled. A huge amount of information have to be defined to create a complete model. The logic behavior of a system can to a small extent be modeled with these methods, but far from a complete logic model. The SFC models are similar to the SP models but cannot be auto generated. In the SP method the idea is to define as few demands as possible and the SOPs will automatically be generated. This makes the model flexible and if a demand is changed the model can easily be updated. Thus, the SP method has several advantages compared to other methods and can be a great help in manufacturing design in the feature.

After evaluating Sequence Planner the conclusion is that in this early stage of its development it can not be used at Teamster in an automation project. V-REP on the other hand could be used in order to simulate any automation project. The advantage of using V-REP is the physical engine combined with possibility of simulating almost anything. These properties makes it possible to build very realistic simulations.

When Sequence Planner are fully developed PLC code can be generated in an new way. If V-REP offers the feature to implement PLC code Sequence Planner can be used with Google SketchUp and V-REP to generate a dynamic model including the control code that will be used in the finished automation project, as described in Figure 5.1. The dynamic model can be used to verify the control code and the mechanical part of the project. This will reduce late modifications and make the development process shorter.

Bibliography

K. Bengtsson. *Operation Specification for Sequence Planning and Automation Design.* Licentiate thesis, Chalmers University of Technology, 2009.

K. Bengtsson, B. Lennartson, and C. Yuan. The origin of operations: Interactions between the product and the manufacturing automation control system. In *13th IFAC Symposium on Information Control Problems in Manufacturing Technology, INCOM'09, Moscow, Russia*, pages 40–5, 2009.

A. Boeing and T. Bräunl. Evaluation of real-time physics simulation systems. In *5th international conference on Computer graphics and interactive techniques in Australia and Southeast Asia*, pages 281–288, 2007.

Bullet. Bullet Physics Engine, 2011. URL `www.bulletphysics.org/`. 2011-05-11.

L. Caponetti and A. M. Fanelli. Computer-aided simulation for bone surgery. *IEEE Computer Graphics and Applications*, 13:86–92, 1993. ISSN 0272-1716.

L. Chao. Automated generation of PLC program in PLCopen XML format. Master's thesis, Chalmers University of Technology, 2009. Tech. Rep. EX037/2009.

D. Christensen, D. Brandt, K. Stoy, and U. P. Schultz. A unified simulator for self-reconfigurable robots. In *IEEE/RSJ International Conference on Intelligent Robots and Systems, Nice, France*, pages 870–876, 2008. ISBN 978-1-4244-2057-5.

J. J. Craig. Simulation-based robot cell design in adeptrapid. In *IEEE International Conference on Robotics and Automation, Albuquerque, New Mexico , USA*, volume 4, pages 3214–3219, 1997. ISBN 0-7803-3612-7.

E. Drumwright, J. Hsu, N. Koenig, and D. Shell. Extending open dynamics engine for robotics simulation. In *Proceedings of the Second international conference on Simulation, modeling, and programming for autonomous robots*, SIMPAR'10, pages 38–50, Berlin, Heidelberg, 2010. Springer-Verlag. ISBN 3-642-17318-7, 978-3-642-17318-9.

M. Fabian. *Industrial Automation, Lecture Notes (SSY 065).* Department of Signals and Systems, Chalmers University of Technology, 2006.

M. Freese. V-REP. Virtual Robot Experimentation Platform., 2011a. URL `v-rep.eu`. 2011-04-14.

M. Freese. Marc freese. Mail conversation, 2011b. 2011-04-14.

W. A. Halang and A. H. Frigeri. Methods and languages for safety-related real-time programming. In *Proceedings of the 17th International Conference on Computer Safety, Reliability and Security*, SAFECOMP '98, pages 196–208, London, UK, 1998. Springer-Verlag. ISBN 3-540-65110-1.

C. Johnsson and K.-E. Årzén. Grafchart and grafcet: A comparison between two graphical languages aimed for sequential control applications. In *Preprints 14th World Congress of IFAC, Beijing, P.R. China*, volume A, pages 19–24, 1999.

N. Koenig and A. Howard. Design and use paradigms for gazebo, an open-source multi-robot simulator. *2004 IEEERSJ International Conference on Intelligent Robots and Systems IROS IEEE Cat No04CH37566*, 3:2149–2154, 2004.

B. Lennartson, K. Bengtsson, C. Yuan, K. Andersson, M. Fabian, P. Falkman, and K. Åkesson. Sequence planning for integrated product, process and automation design. *IEEE Transactions on Automation science and Engineering*, 7(4):791 – 802, 2010. ISSN 1545-5955.

D. G. Malcolm, J. H. Roseboom, and C. E. Clark. Application of a technique for research. and development program evaluation. *Operations Research*, 7(5):646–669, 1960.

PLCopen. PLCopen.com, 2011. URL `http://www.plcopen.org/pages/tc6_xml/`. 2011-05-12.

L. Schneider-Vidi. K-team about v-rep. Mail conversation, 2011. 2011-05-04.

M. Skoldstam, K. Åkesson, and M. Fabian. Modeling of discrete event systems using finite automata with variables. In *46th IEEE Conference on Decision and Control*, pages 3387–3392, 2007.

R. Smith. Open Dynamics Engine, 2007. URL `www.ode.org`. 2011-05-11.

N. Walsh. O'REILLY XML.com, 2011. URL `http://www.xml.com/pub/a/98/10/guide0.html`. 2011-05-12.

J. M. Wilson. Gantt charts: A centenary appreciation. *European Journal of Operational Research*, 149:430–437, 2003.

K. Åkesson, M. Fabian, H. Flordal, and R. Malik. Supremica - an integrated environment for verification, synthesis and simulation of discrete event systems. In *Discrete Event Systems, 2006 8th International Workshop on In Discrete Event Systems*, pages 384–385, 2006.

Appendix A

Sequence Planner overview

In this appendix the two Sequence Planner models of the S*Solar cell can be seen in Figure A.1-A.3.

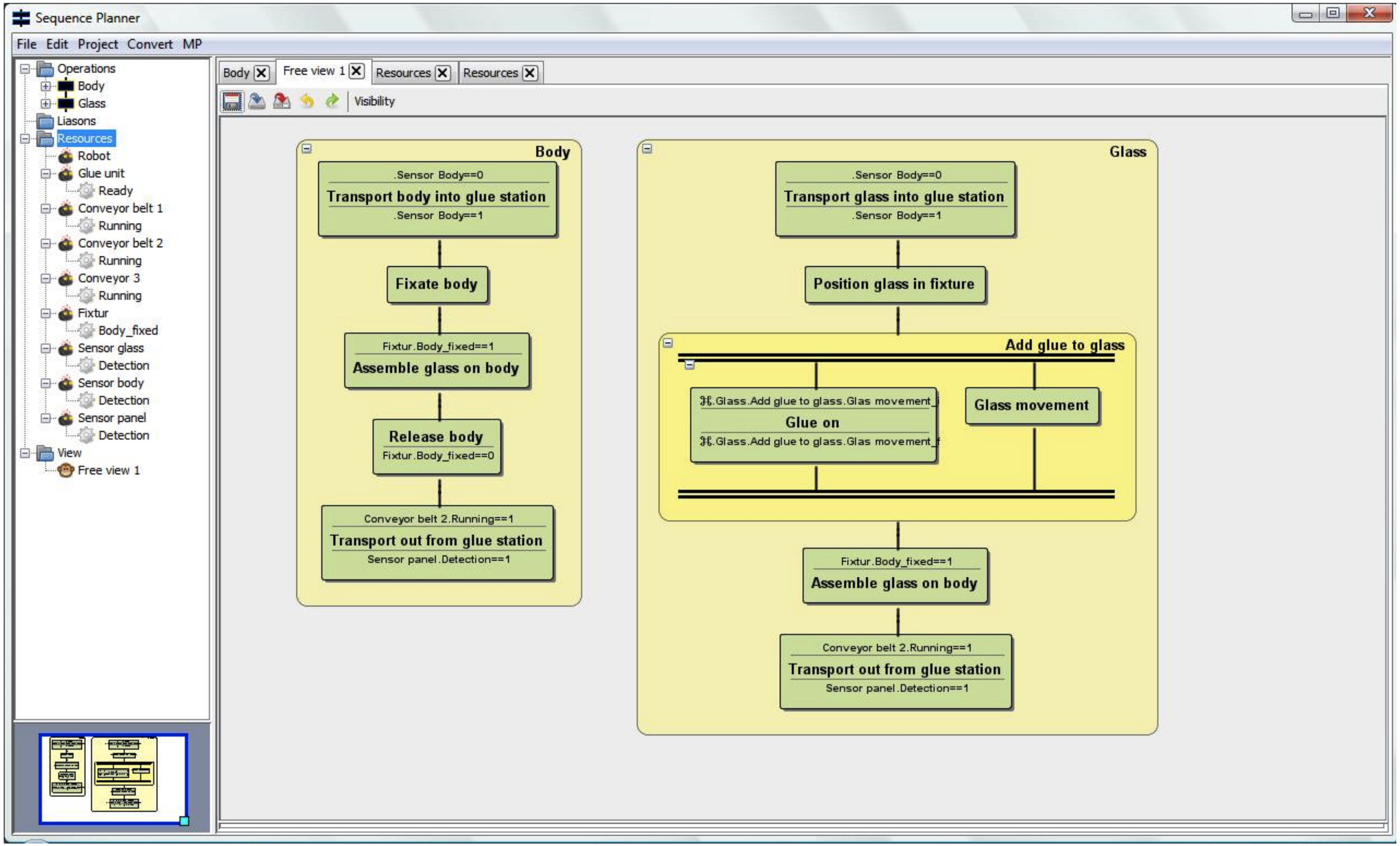

Figure A.1: S*Solar modeled with product focused SOPs.

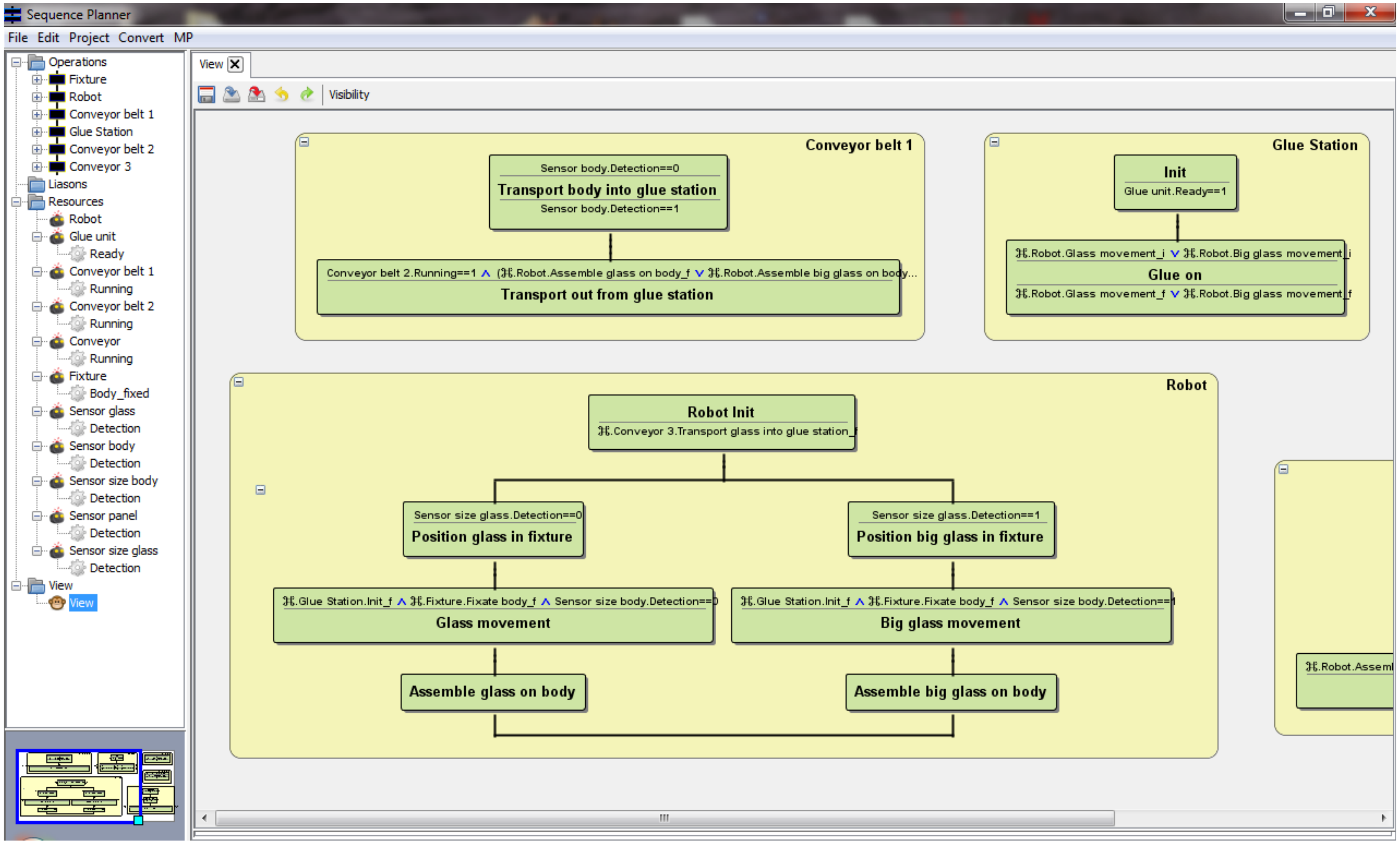

Figure A.2: S*Solar modeled with resource focused SOPs.

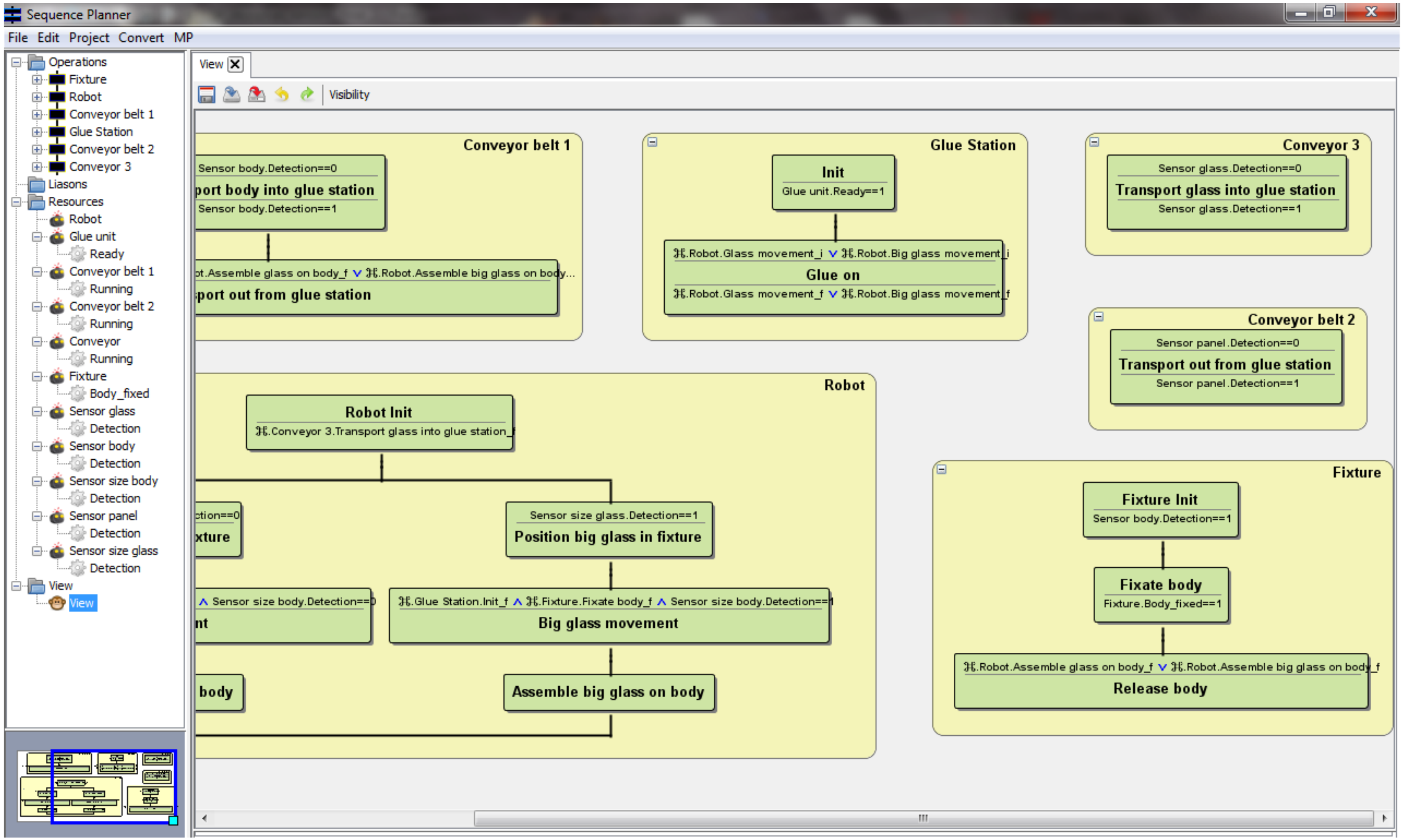

Figure A.3: S*Solar modeled with resource focused SOPs.

Appendix B

V-REP

B.1 Importing Google SketchUp models to V-REP

When importing models into V-REP from Google SketchUp some problems occur. For example there are problems with the color and textures of the model but also with grouped objects and objects that are made into component. This is a normal problem when converting CAD-files.

When Google SketchUp models are exported as .3DS files and then imported to V-REP every object of the model are assigned initializing physical properties. These properties are a mass of 1.00 Kg and friction coefficient of 1.00. All objects are also initialized to static objects.

B.1.1 Colors and textures

When Google SketchUp models are exported to V-REP, some problems occurs. First of all the textures that have been used to build the model in Google SketchUp is not linked to the model when imported to V-REP. This will result in a model that does not look like the original model.

Even though no textures are used in order to create the model in Google SketchUp, when imported into V-REP all objects get a randomly selected color. So weather textures or just coloring is used to create the model, the imported model will not reflect the original model in the color scheme.

B.1.2 Grouping of objects

When objects are grouped in Google SketchUp and the model is imported to V-REP the grouping does not translate to V-REP. This is a problem since all objects have to be grouped again in V-REP in order to work with that group of objects as a single unit. The same problem can also be seen if the objects are made into a component in Google SketchUp. Besides the inconvenient of having to regroup all objects, in order to optimize the model for V-REP all objects of a model should be grouped as far as possible.

B.2 How to make pure shapes

As described in Section 4.3.1 the original meshes from the SketchUp model can often easily be approximated by a simple pure shape, for example a rectangle as in Figure 4.4 in Section 4.3.1. This is done by controlling the size of the shape and then add a new pure shape with the same size at the same position and orientation as the original and then make it its parent. For all other areas like proximity sensor detection, minimum distance calculations and visualizations the meshes should be used (Freese, 2011b).

B.3 Video recorder

In V-REP a video recorder is included. With this tool you can make videos of your simulations and present for the customer. The video files gets very big and the quality could be better. There is a number of codecs to choose from which can reduce the size to some extent. It takes a lot of time to record the videos since the simulation slows down due to the computational load when the video is being created. The recording can be started and stopped whenever wanted during the simulation which makes it convenient to make short clips for a video.