DESIGN OF A MOBILE WORKSTATION FOR VALUE STREAM MAPPING

Submitted to
122 Hitchcock Hall
College of Engineering
The Ohio State University
Columbus, Ohio 43210

Honors Thesis
By:

Russell Mahoney
94 W. Maynard Ave.
Columbus, Ohio 43202

May 18, 2006

Table of Contents

List of Figures

List of Tables

Abstract:

Lean Thinking is a manufacturing strategy that is focused on eliminating waste. Value Stream Mapping (VSM) is a tool that has globally become the accepted foundation for strategic planning to implement Lean Thinking in industry. However, most VSM's are still created using a pencil-and-paper method or simple drawing software; these methods lack the power of Industrial Engineering (IE) tools for stochastic modeling with simulation. This thesis will address the limitations of traditional Value Stream Mapping and propose a merger of computer technology with Industrial Engineering theory to create a mobile workstation for Value Stream Mapping in real project situations. A trained IE could use this workstation to generate VSM's and dynamically evaluate operational improvements even as he/she leads a team of top managers on a walkthrough of the facility.

Introduction:

Value Stream Mapping

Lean Thinking is a manufacturing strategy to improve production output and reduce costs to maintain a competitive advantage in business. Lean means "More for Less," and by capitalizing on the principles and best practices of this manufacturing strategy, a company can increase profit and productivity with less costs and investments in raw material, labor, and other capital resources. Essentially, the Lean philosophy is to "compete against *perfection* by identifying all activities that are *muda (waste)* and eliminating them" [Womack & Jones, 49].

Value Stream Mapping is a manual method for mapping a Value Stream which "is all the actions (both value added and non-value added) currently required to bring a product through the main flows essential to every product: (1) the production flow from raw material into the arms of the customer, and (2) the design flow from concept to launch" [Rother & Shook, 3]. A non-value added activity is any activity that does not increase the value of the final product in the customer's eyes. An example of a non-value added activity is waiting time for a product before it is processed at a workcenter. A value-added activity is any activity that adds actual value for the customer. It is an accepted fact in industry that value-added time, at best, accounts for 1% to 5% of the total duration of time that a product moves in a value stream. VSM's help to visualize the current material and information flows and are most effective at identifying muda (any kind of waste) due to non-value added activities. Value Stream Mapping is done in two phases: first a Current State Map is created, and after the waste and sources of waste have been identified, a Future State Map is created to target changes that will achieve improvements in flow. The objective of a VSM effort is to eliminate non-value added activities in order to have a "Lean" process flow in the Future State of the manufacturing system.

Value Stream Mapping is by no means a perfect tool. *Learning to See* [Rother and Shook, 14&15] goes so far to say "Always draw by hand in pencil," and "Resist the temptation to use a computer." However, there is much room for improvement in this manual pencil-and-paper method, especially if Industrial Engineering tools such as queuing theory and simulation were incorporated with advanced technology for mobile wireless computing to do "VSM on the fly".

Once a map is drawn by hand or by computer, it needs to be displayed, discussed, and have action plans created. Activities required to do these steps could include setting up a meeting, booking a conference room, getting all of the appropriate managers together, organizing the applicable information into presentation form, simulating potential what-if scenarios to improve current operations, accessing offline data via shop-floor terminals, etc. Drawing VSM's by hand, displaying them, and making changes to them is a cumbersome process with much waste built into the process. If the goal of VSM is to eliminate waste, then the pencil-and-paper method is itself a wasteful process! All of the aforementioned wasteful activities could be done on the fly on the factory floor. Eliminating the non-value added activities involved in VSM, and doing instantaneous data collection, data processing, mapping, and simulation are feasible if a mobile, wireless workstation was developed and utilized for this purpose. By melding Lean Thinking concepts and simulation, along with new VSM software in the creation, editing, and instantaneous displaying of dynamic performance of VSM's on the factory floor, an organization can better determine where resources should be applied [McDonald et al, 6].

By utilizing a mobile workstation designed for VSM, a trained Industrial Engineer could walk through the factory with his/her managers, mapping the flow as he/she moves from workcenter to workcenter, identify the wastes, simulate what-if scenarios, and immediately get the results of these scenarios. There would be no need to book a conference room, cover entire walls with butcher paper and Post-Its, or participate in other non-value added activities. Current and Future Value Stream Maps could be quickly created, and action plans to achieve the Future State could be immediately tested based upon reliable simulations. The integration of computer technology, simulation, and IE tools will revolutionize the manual method of VSM thereby creating a tool that is suited for the 21st century. This Dynamic Value Stream Mapping (DVSM) tool will be extremely powerful and useful to every organization that is trying to implement Lean Thinking. It is important to note that in this project, the Dynamic Value Stream Maps will also be referred to as 'models' because in reality they are simulation models.

Company Background: Tigerpoly Inc.

In order for this research to be worthwhile, it had to be motivated by a real manufacturing problem in industry. This led to a search of companies that would be willing to participate in this project. The main search criteria for companies with whom to partner was their acceptance of Lean Thinking in terms of both management and the workforce. If the philosophy of Lean Thinking is not "bought in to" at all levels of the organization, Lean will inevitably fail. If upper management does not buy into Lean, there will be no initiative to reduce waste. If the workforce does not understand Lean, they will continue to work as they always have. A company cannot just do a series of unplanned Lean projects and become Lean; all aspects of the business have to be Lean and all departmental functions have to be striving to be Lean in order for it to be successful.

The search for a Lean-savvy company in Central Ohio led to Tigerpoly Manufacturing Inc. (henceforth referred to as Tigerpoly). Tigerpoly is a manufacturer of quality blow-molded, injection-molded, and extruded plastic and rubber parts. Tigerpoly is a Tier One supplier to Honda; their parts go directly into Honda automobiles. They are a successful company with $53.5 million in sales in 2003 and in 2004 they received all three Honda Supplier Awards: Quality, Delivery, and Productivity Excellence [Source: www.tigerpoly.com].

Tigerpoly produces 7 types of parts for Honda: engine covers, resonator chambers, air intake ducts, air conditioning shrouds, reserve tanks, shock covers, and rubber intake hoses. The focus of this project will be on two parts in a family of injection molded resonator chambers that are welded around the blow molded tube. These two parts (PVJ and RDJ) utilize the same machines and go through almost the exact same process except for a slight difference in cycle times, therefore, they can be analyzed as one part family. These parts account for 98% of the utilization of the machines in the process (B6 and I15) with the remaining machine capacity being used to make service parts. The blow molding machine produces a tube which later gets welded into the injection molded chamber. In the current process, ground rubber is liquefied and blow molded into the specific part shape in B6, the blow molding machine. After the part is blow molded, it has to cool for approximately two minutes before the extra material can be sheared off. This shearing of the extra material is the deflash stage. Then the part gets put into a two-step cutting machine and a final assembly is done before the part is stacked up in a Work In Progress (WIP) crate. This entire blow molding process is done by two dedicated workers. Once this crate fills to its capacity of 160 tubes, the crate is transferred by a material handler to I15, the injection molding machine. At the injection molding machine, two halves of the chamber (half A and half B) get produced to merge with the tube from the blow molding process. The tube is placed inside the two halves of the chamber and the parts are welded together in a welding machine. A final assembly is then done to inspect and complete the product. The entire injection molding process is done by two dedicated workers. These finished goods are put in a bin which can hold 30 units, which is then carried by the material handler to the finished goods racks. Deliveries are made to Honda three times a day. This is a push system based on production demand volumes. Blow molding starts production and pushes on whatever product it makes to the injection molding machine. Figure 1 is a process flow diagram for the entire production process.

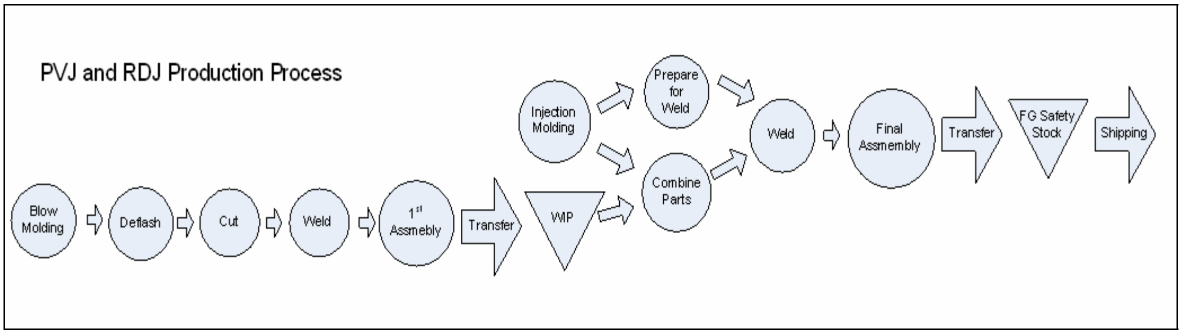


Figure 1: Process flow diagram for PVJ and RDJ production.

Company Background: CreateASoft Inc.

Finding the company with an applicable manufacturing problem was only one component of this project. Another component was finding or creating the software capable of Dynamic Value Stream Mapping. In terms of scope of this Undergraduate Honors Research project, it was determined early on that creating the software or even modifying an existing software to create Dynamic Value Stream Maps would be too difficult. That led to a search of companies that are already attempting to do Dynamic Value Stream Maps with their own proprietary software. Several companies offer in depth Value Stream Mapping drawing capabilities, but few have combined Value Stream Mapping with the power of simulation/modeling.

CreateASoft Inc. (henceforth referred to as CreateASoft) is one company who is venturing into the arena of Dynamic Value Stream Maps. CreateASoft is a dynamic process improvement company providing process based operations leading edge predictive technology, industry independent solutions, scalable business intelligence, risk free analysis and end-to-end support [Source: www.createasoft.com]. CreateASoft offers several process simulators for different industries including SimCAD for manufacturing, SimService for service, and SimCARE for health care. It also offers Value Stream Mapping software which is only capable of static VSM's. The software used for this project is SimCAD Process Simulator 7.1. This product offers many features and capabilities including Contingency Planning, Real Time Scheduling, Process Simulation, Dynamic Value Network Mapping, Capacity Planning, Real Time Data Connectivity, 2D and 3D Visualization, and Dynamic Value Stream Mapping. Obviously, the most important feature of the software for this project is Dynamic Value Stream Mapping. SimCAD was used in this project to create Current and Future DVSM's to test potential Lean improvements and evaluate the results.

Problem Statement

The goal of this research is to determine the effectiveness of developing a mobile workstation for Value Stream Mapping for use in any and all manufacturing and service industries. A mobile workstation will be developed, used, and tested for creation of Dynamic Value Stream Maps "on the fly" on the factory floor. I intend to prove that, by combining the skills of an Industrial Engineer with the technology of Value Stream Mapping software on a mobile computing workstation, the currently cumbersome and simple VSM method can revolutionized. A major component of this research will be to evaluate existing VSM software in order to gauge the current improvements over the pencil-and-paper method but also to identify opportunities for improvement from an Industrial Engineering standpoint.

Methodology and Approach

The first part of this project included finding the right companies to partner with. The next part of the project included procuring the necessary items to build the workstation. The necessary items were a mobile cart, a laptop computer with SimCAD installed, and a portable printer. Optional items that were considered to possibly be of value were a digital camera, a projector, and a back-up power source. Time was spent researching audio/video and computer mobile carts, but none of them truly met all the criteria. They were heavy, non-collapsible, and expensive. What was needed was a cart that could be built up and taken down in minutes and was easy enough to rapidly move throughout the plant. The solution was relatively simple and cheap. A plastic storage rack was purchased along with four caster wheels. These wheels were attached to the base of the storage rack via wood runners and industrial strength adhesive. The creation was a light, cheap, yet durable cart that could effectively hold all of the necessary items and could be broken down to fit in the trunk of a car. Industrial cart options were considered, but it was determined to be most feasible and reasonable to simply make one. However, they may be reasons to purchase an industrial cart especially if the plant is exceptionally dusty and the computer needs dust protection. A breakdown of different cart options can be seen in Table 1.

Method	Price	Positives	Negatives	Picture
Build Own	$20-$50	Cheap! Light Weight Collapsible	"Not professional looking" Made of plastic No dust protection	
Metal A/V Cart Source: http://www.gilmorekramer.com//more_info/black_tuffy_ utility_carts/black_tuffy_utility_carts.shtml	$240-$500	Relatively Cheap Strong - Metal Optional Outlet Accessory	Heavy: 30-60lbs Not Easily Collapsed No dust protection	
Industrial Comp Cart Source: http://www.listaintl.com/files/compcab.htm	$1,600	Designed for Industrial Use Intake fans & filters protects from dust Lockable - protect from theft Sliding drawers	Relatively costly Not Collapsible Very Heavy - 330lbs	

Table 1: Cart alternatives for workstation.

The laptop was procured on loan from the IWSE Department, and a mobile printer with a battery was purchased to complement the computer. The purpose of the printer is to print out copies of the map or reports to give to managers while on the factory floor; eliminating the need for everyone to crowd around one computer screen. The completed workstation can be seen in Figure 1.

Figure 2: Completed Dynamic Mobile Value Stream Mapping Workstation.

Purchasing the optional items was considered, but determined too costly for this project. A digital camera could be used to take snapshots and document the potential improvement areas; i.e. if there is a large pile of WIP in front of a process. These pictures could then be included in a final report. The projector could be used to eliminate the need for a conference room. While walking managers along a Value Stream, the cart could be pulled up to any white wall and the DVSM could be displayed to everyone. The best place to pull aside to would be a white wall with a power source in order to power the projector and recharge any other items on the cart. The purpose of the backup power source is obvious; to power the laptop, printer, projector, and any other items on the cart. American Power Conversion Corp. offers battery backup units that can give thirty minutes of back-up power [Source: www.apc.com]. These units cost around $250; the purchase of one was not deemed necessary for this project.

In order to create any simulation, the most important part is to truly understand the system to be modeled. In order to do this, several trips were made to Tigerpoly to watch the processing of the PVJ and RDJ parts at both the blow molding and injection molding machines. Managers and workers were interviewed in order to understand the intricacies of the process. There has also been much email communication with the Lean Manufacturing Engineer, Jim Fuchs. He has been quite valuable in answering questions. To get the assembly times, a time study was done on the workers for both the PVJ and RDJ assembly tasks. These times were fit to a triangular distribution and the times were used in modeling of the process. The get the processing times of the blow molding and injection molding processes, Jim used Tigerpoly's MRP system. The scheduler gave daily demand quantities; 780 PVJ parts a day and 300 RDJ parts a day.

The actual modeling of the DVSM is a tremendously important part of the project. If the maps do not accurately model the true production system, they are essentially worthless. A Current State and a Future State DVSM were created for this project. These maps capture the process from raw material flow into the blow molding operation through the injection molding process and finally to finished goods shipping. After the maps were created, they had to be validated against the real process to see that the simulated system actually does represent the actual system. This was done by checking key variables from the Current State model versus the real life system with Tigerpoly engineers and schedulers. Key variables include overall lead time for PVJ and RDJ and the amount of objects completed in one week of production. This validation processes will be discussed in detail in the model results section.

The most powerful tool that simulation has available to it is the power to test "what-if" scenarios. That is the main purpose of creating the two DVSM's in this project. The Current Map is used to show how the process is currently running and to identify waste in the process. The Future Map improves on those identified areas. In order to prove that the changes to the Future State Map made a difference, an experiment was conducted. The experiment involved running the Current State model and Future State model and comparing the key variables. The key variables tracked were product lead time, count of completed objects, and the amount of WIP in the system at the end of the run. The values will be compared, and this will be discussed in the results section.

The next part of the process involves refining and potentially optimizing the Future State model. However, as is discussed later, a modified version of the Current State model is actually used as the 'Future State', Once it was proven that the Future State model is an improvement over the Current State, experiments were done to the Future State simulation to fine-tune production. The two main points of interest were how material handler availability and WIP crate batch size effect production. A Design of Experiments was done to see if these factors have an effect on product lead time, objects completed, and WIP in the system. Based on this data, refined recommendations have been made to Tigerpoly.

Current State Model Description

First off, there is a need to preface this section that SimCAD Value Stream Maps are not true Value Stream Maps. The processes are made to look like VSM symbols, but the models themselves more resemble a process flow diagram than a VSM. This being said, the Current State SimCAD model of RDJ and PVJ production is almost identical to the process flow diagram previously seen in Figure 1. This section of the report will go greater detail about the process at every step. The standard SimCAD process view of the full Current State model can be seen in APPENDIX A. The VSM view of the process can be seen in APPENDIX B. The VSM view has a few processes hidden because they are not crucial to visually interpret the Value Stream. The SimCAD VSM view also shows greater detail since it represents processes with different symbols while showing dynamic process data within the process boxes.

First, the SimCAD Dynamic Value Stream Map will be compared to a standard static Value Stream Map. A Microsoft Visio drawn VSM can be seen in APPENDIX C. The two maps are similar; however, the SimCAD map is more broken out because greater detail had to go into each of the parts of the model. The static VSM has the luxury of combining processes as long as the processing times are added together. This is what happens at 'Assm' block which combines deflash, cut, and welding stages. The maps are also similar because they have kaizen bursts around the focus area in blow molding and at the transfer from blow molding to injection molding. The static map displays cycle times for the processes. The dynamic Value Stream Map can do that, and a lot more for each process. In each process option box, the user has the option to 'check' data that they want to see the process display. This VSM option box can be seen in Figure 3.

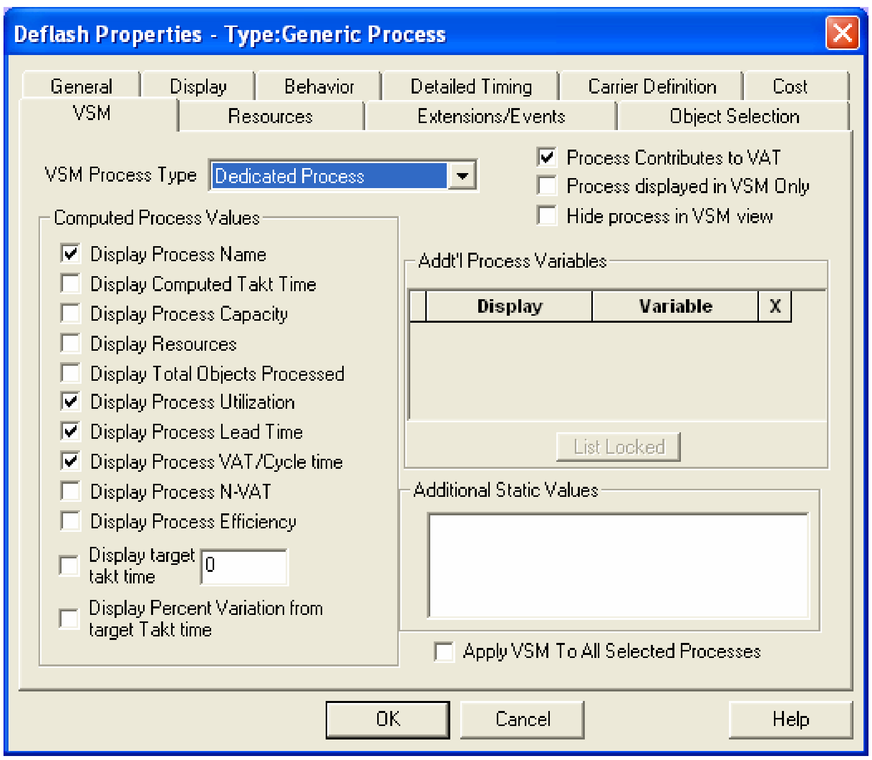


Figure 3: SimCAD Option box for VSM processes.

The SimCAD DVSM does do a good job at showing data specific to the individual processes. What it does not do well is showing overall model information. A main component of a VSM is the Value Added Ratio (VAR). This is the ratio of value added time over the total lead time. SimCAD does not calculate this ratio. It tracks value added time for each process, but it never totals it up and divides it by the total lead time. This would be a relatively simple thing for the SimCAD programmers to do, it seems this metric has just been left out of the software. Also, it can be noted that the static map has a line below the processes that tracks value added time and non-value added time. This line is also absent in the DVSM and it is impossible to put in. SimCAD is not a drawing tool, and lines cannot be drawn unless they connect processes together. The only way to get an object on the DVSM is to make it a process. These processes can simply be dummy process; the kaizen bursts are dummy processes. The static map also shows informational flows to and from the production control entity. These informational flows are another luxury that cannot be done in SimCAD. For more information on the SimCAD DVSM's capabilities versus traditional VSM capabilities see APPENDIX J.

To get back to the SimCAD Current State Dynamic Value Stream Map, the general model parameters will be discussed. The model produces both PVJ and RDJ parts. It starts by producing 750 PVJ parts, then a two hour changeover occurs and RDJ production occurs. After 300 RDJ parts get produced, another two hour changeover happens and it switches back to producing PVJ parts. This cycle continues for one week of simulation time. These production numbers differ from the actual daily demand for PVJ and RDJ. The actual daily demand is 780 PVJ and 300 RDJ. However, because lots are transferred in crates of 160 for the WIP tubes and then 30 for the finished goods, the actual daily demand does not create even multiples of full crates, i.e. 780 parts is equivalent 4.875 PVJ crates. SimCAD would not work with the incomplete crates, so PVJ production was reduced to 750 and the crate size was reduced to 150. This way, even multiples of both the WIP crates and the finished goods crates could be produced. The model batches the blow molded parts at 'Transfer1' into lots of 150 and un-batches the lots at the next process, which is 'WIP'. The model also batches parts into lots of 30 at 'Transfer2' and un-batches them at 'Finished Goods'. It is important to note that this act of arbitrarily decreasing crate size does affect the product lead time. This is considered in the results though. The change had to be made in order for the model to successfully run.

The production of two parts was actually a relatively complicated task to accomplish in SimCAD. SimCAD is designed for multiple parts and every process is capable of handling the different parts and processing them with different times. Figure 4 shows a good example of the different processing times in the injection molding machine.

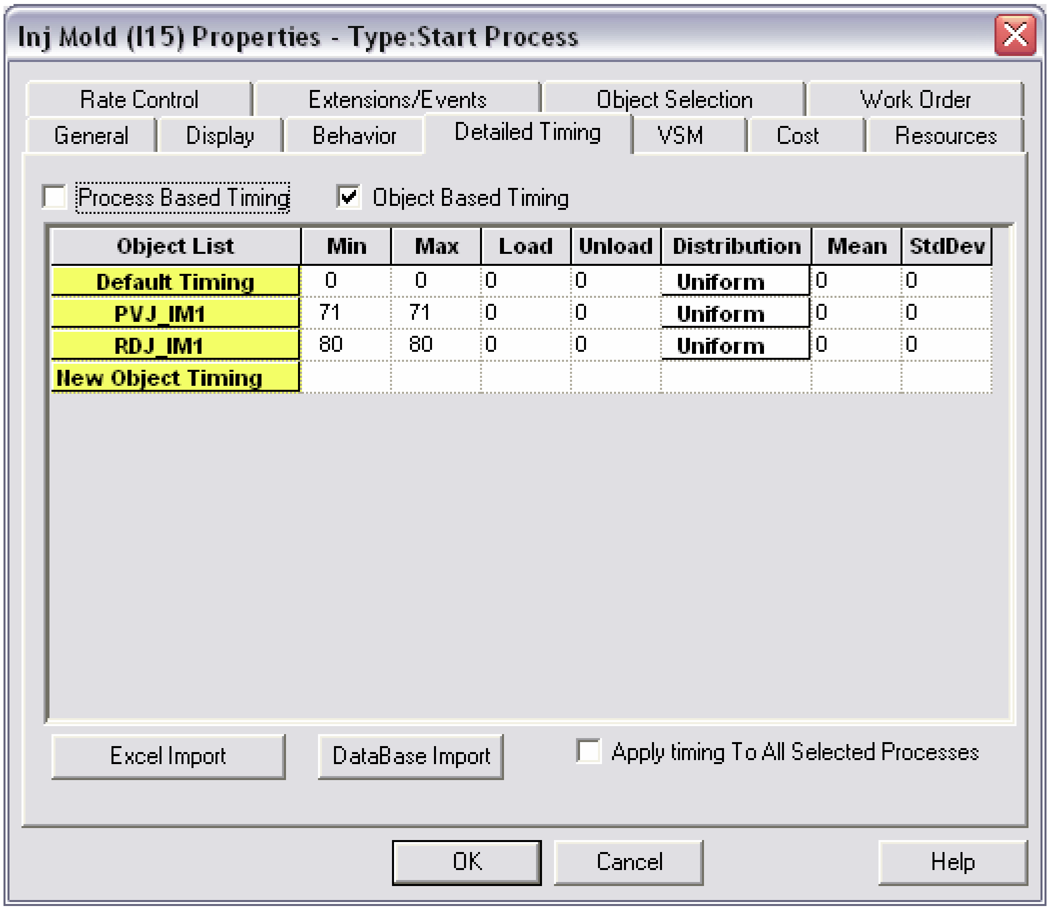


Figure 4: Detailed timing for the injection molding production process.

Note that the PVJ_IM1 timing is different than that of the RDJ_IM1 timing. Start processes are also designed to be able to create different types of objects. The start processes can produce a set number of different products based on a work order. However, because of other complexities in the process, mainly the fact that B6 produces scrap, these static work orders could not be used. B6 has roughly a six percent scrap rate and this number was used in the model. Since six percent of the products are randomly scrapped, it was not possible to set the work order for 750 units of PVJ. The model needed to dynamically keep track of part production and signal for the changeover to occur once the daily demand was filled. This was done by incrementing counters in conditional statements of code. Essentially, if PVJ parts are being produced and if 750 units have yet to be produced, keep making them and add one to the counter every time. The same works for RDJ parts only with 300 units of demand. Once the daily demand is reached, it triggers a changeover in the Blow Molding process. The coding for this can be seen in APPENDIX D. The Injection Molding process changeover is triggered when the first item of a new type is activated in the process. This is not the most efficient method to do this because it really should start the changeover when the last object of a certain type is processes, but this way more closely matches the real current state.

In the current state model, there are five resources. There are two resources for the blow molding process (B2_1 and B2_2), two resources for the injection molding process (I15_1 and I15_2), and one material handler (mat_hand). Resources can be assigned to multiple processes which allowed the object to move from process to process and the resource to move with it. All resources work on a shift (entitled WorkerShift in SimCAD) that takes into account breaks for meals and shift changes. The shift schedule can be seen in Figure 5.

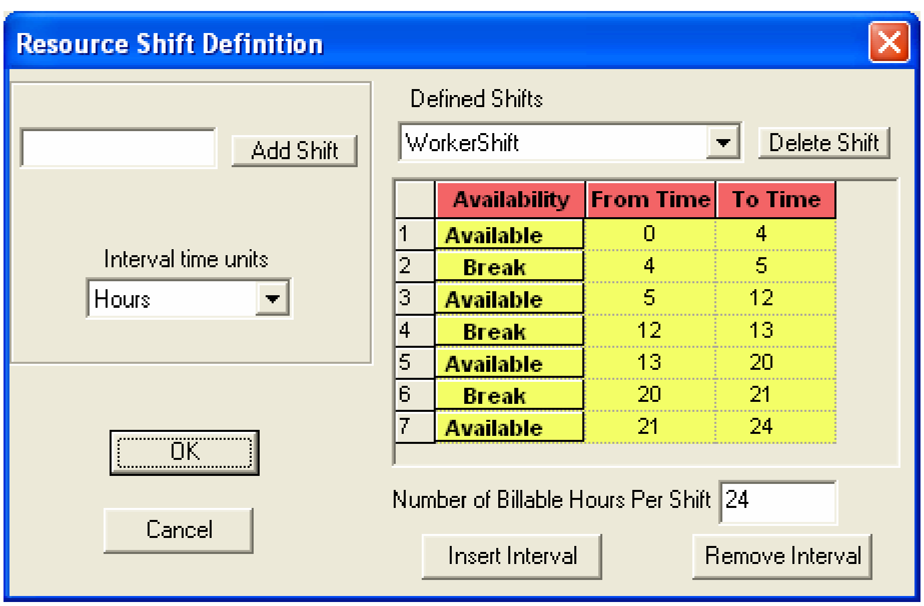


Figure 5: Shift breakdown for all workers in the Current State model.

Each process has its own processing time. Except for the blow molding and injection molding times, most of the processing times are based upon triangular distributions obtained from time studies on the floor. See Table 2 for in depth process information. It has a process by process breakdown with the resource assigned to it and whether or not they are assigned for the whole cycle time or just a load and unload time. It also contains the detailed timing for each process, and how many units it processes at a time. It is important to note that most connection lines are left off of this table because most of them have a zero processing time and no resources assigned to it. However, if the connection time is important, say the transfer time for a batch of PVJ_BM parts, then that connection is included in the table. In that case, the objects get batched at 'Transfer1' then travel together to get un-batched at "WIP". The 'Transfer' process only serves the purpose of batching the objects, the actual transfer time occurs in the connection between the two processes. The other batch operation occurs at 'Transfer2' and the un-batch happens at 'Finished Goods'. The table also has the part type that is outputted by the process. The process can output different part types because the object gets batched and combined with other parts to form new parts types. Initially, the blow molding machine makes either PVJ_BM or RDJ_BM. The injection molding machine outputs two PVJ_IM1 or RDJ_IM1 parts that are split. One half goes to 'Assm1' and the other goes to 'Assm2'. The BM part combines with the IM part at the 'Assm2' process; the output is PVJ_IM or RDJ_IM. This new IM part is combined with the IM1 part at 'Load Weld' producing the final PVJ or RDJ part respectively.

Process Name	**Resource**	**PVJ Timing**	**RDJ Timing**	**Process Capacity**	**Part Output**
Blow Molding (B2)	B2_1: Load (1s), Unload (1s)	85s	84s	1	PVJ_BM, RDJ_BM
Deflash	B2_1: Cycle	Tria(23,24,30)	Tria(23,24,30)	1	PVJ_BM, RDJ_BM
Cut1	B2_1: Load (1s), Unload (1s)	Tria(17,19,20)	Tria(17,19,20)	1	PVJ_BM, RDJ_BM
Cut1	B2_1: Load (1s), Unload (1s)	Tria(17,19,20)	Tria(17,19,20)	1	PVJ_BM, RDJ_BM
Weld1	B2_1: Cycle	Tria(13,15,17)	Tria(13,15,17)	1	PVJ_BM, RDJ_BM
Assembly	B2_2: Cycle	Tria(44,50,61)	Tria(44,50,61)	1	PVJ_BM, RDJ_BM
Transfer1	n/a	0	0	1	PVJ_BM_Basket, RDJ_BM_Basket
Connection	Mat_Hand: Cycle	180s	180s	150	PVJ_BM_Basket, RDJ_BM_Basket
WIP	n/a	1s	1s	1	PVJ_BM, RDJ_BM
Inj Mold (I15)	I15_1: Unload (1s)	71	80	1	PVJ_IM1, RDJ_IM1
Assm1	I15_1: Cycle	Tria(33,37,40)	Tria(33,37,40)	1	PVJ_IM1, RDJ_IM1
Assm2	I15_2: Cycle	Tria(10,15,24)	Tria(10,15,24)	1	PVJ_IM, RDJ_IM
Load Weld	I15_1: Cycle	Tria(12,13,14)	Tria(12,13,14)	1	PVJ, RDJ

Weld2	I15_1: Unload (1s)	Tria(47,47.5,48)	Tria(47,47.5,48)	1	PVJ, RDJ
Final Assembly	I15_2: Cycle	Tria(29,38,46)	Tria(29,38,46)	1	PVJ, RDJ
Transfer2	n/a	0	0	1	PVJ_Tote, RDJ_Tote
Connection	Mat_Hand: Cycle	300s	300s	30	PVJ_Tote, RDJ_Tote
Finished Goods	n/a	0	0	30	PVJ, RDJ
To Honda	n/a	0	0	30	PVJ, RDJ

Table 2: Detailed information for each process in the Current State model.

There are a few more important discrepancies to note about the model that are different from the actual real world process. First, at Tigerpoly there is a large stack of WIP sitting in front of the injection molding machine. When a part count of this WIP was done, there were five crates of blow molded PVJ parts, or 800 parts and two crates of RDJ parts, or 320 parts. The simulation runs without any pre-existing work in progress (WIP) in the system. Work in progress may build up during the course of the simulation, but it does not start with WIP. This was done for two reasons. The first and foremost was that it is difficult to accurately create this WIP in the system in SimCAD. The model can be started with certain objects loaded up in the initial state; however, this creates problems with data accuracy. The objects start to get created in the line at time zero but continue to spawn while the simulation time is increasing. Also, since these objects are created in the 'WIP' process, downstream of the actual start process, their lead times end up being significantly less than objects that travel the whole value stream. Creating WIP during the initial state also caused major issues with PVJ/RDJ part rotations. Since the production of these parts is based upon counters, starting with any objects in the system interfered with these counters getting incremented correctly. This pre-existing WIP was eliminated; however, it should not dramatically affect the results of the simulation. That is because the Current State model results will be compared to the Finished State model results, and the Finished State model will also not have pre-existing WIP. Both models will run from a 'dry state,' meaning that there will be absolutely no WIP in the system at the start of the model. This will give lead time and production capacity results based on actual system performance that is not factored by WIP build ups that occurred in the in the past.

Another factor in the model that is different from the real world case is finished goods safety stock. By policy, Tigerpoly keeps five days of finished goods safety stock in the plant. This safety stock is not modeled in the Current State simulation. The reason for this again is relative to the Future State model. It was decided that the important area to track product lead time is from the beginning of blow molding production until the product gets completed and enters the finished good flow racks. Thus, the lead time that the product incurs after entering the flow racks is not important to track in this model. In both the Current State and Future State models, the extra lead time would have been five days. The safety stock is left out of both models because it is would have caused the exact same results in both the Future and Current state model. In both models, products exit the system as soon as they get transferred to the finished goods area.

The current state model will be run until completion of one production week. That is 120 hours of production (five days * 24 hours). Product lead time (average time for a object to complete from start to finish) and objects completed (number of objects that exited the system) are automatically tracked by SimCAD. Cycle time (total time/objects completed) is also tracked by SimCAD, however, because of the production variety between RDJ and PVJ, these times get skewed to the high side since total time keeps increasing while only one type of product is produced. Since PVJ and RDJ are not in dedicated production, SimCAD's default cycle time cannot be used. Another metric tracked is WIP at the end of the simulation time. This is one static value, not an average. It is calculated by incrementing a counter when an object is created, and only decrementing that counter when an object exits the system.

The current state model is highlighted by kaizen bubbles to showcase the potential areas of improvement. The main area is the WIP Transfer area. The future state model will test to see what happens if the lot size of the crates is reduced and how that effects production. That area is also highlighted because the reduction of WIP in front of the injection molding process should be a priority because WIP costs money and increases product lead time. Another highlighted area is in the blow molding process. There is an opportunity to set the blow molding machine on kanban to match production demand. Also, there is opportunity to equal the workload between the blow molding resources. Another potential improvement area not shown on the map is the changeover process. In the Current State, changeovers occur in blow molding after production volume is met. Changeovers occur in the injection molding machine once the first part of the new production batch arrives. There is no communication between the two machines. The Future State will be different so that the injection molding machine is aware that a blow molding changeover has occurred so that it can start the changeover as soon as the last part of a batch is processed.

Current State Model Results

The Current State model was run for five runs at its default settings; crate size was 150 units and the material handler is on the same schedule as the other workers. Lead time, objects completed, finished WIP level, and resource utilization were tracked and an average was calculated based on these runs. This data can be seen in Table 3.

Run	PVJ LT (s)	PVJ Obj Comp (units)	RDJ LT (s)	RDJ Ob Comp (units)	End WIP (units)	B2_1 Ut%	B2_2 Ut%	I15_1 Ut%	I15_2 Ut%	Mat Hand Ut%
1	22597	2160	27579	600	161	68	36	66	79	9
2	23317	1500	29598	450	171	50	26	45	57	6
3	22539	1350	27293	300	171	43	22	39	47	5
4	22600	2100	27144	600	145	68	35	64	76	8
5	22142	1350	31214	300	171	43	22	39	47	5
Avg	**22639**	**1692**	**28565.6**	**450**	**163.8**	**54.4**	**28.2**	**50.6**	**61.2**	**6.6**

Table 3: Production data from the Current State simulation.

This data shows us that a PVJ unit can be produced in 22,639 seconds or 6.2 hours and a RDJ unit can be produced in 28,566 seconds or 7.9 hours. The discrepancy between the two is justified because the injection molding cycle time for RDJ is higher than PVJ. In the simulated week, 1,692 units of PVJ and 450 units of RDJ were produced on average. This is well below the demand for weekly production (goal: 3900 PVJ and 1500 RDJ, actual: 3810 PVJ and 1560 RDJ). This reason for this discrepancy is not quite known and there just was not enough time to completely fix the model to 100% accurately model the number of objects completed per week. However, when these values were checked with Jim Fuchs, he said that they were good enough to use for simulation purposes because the values will be compared with a Future State model. The resource utilization data shows that the first blow molding worker, the one in charge of deflashing, cutting, and welding the part is utilized 54% of the time. The second blow molding worker, the one who does the final assembly is only utilized 50.6% of the time. The first injection molding worker is utilized 50.6% of the time while the second one who retrieves the blow molded part and does the final assembly and inspection is utilized 61.2% of the time. These utilization percentages are feasible because they take into account worker breaks and the times they are waiting for parts. There is no actual data available to validate these results. The material handler, who in this model is only in charge of transferring the blow molded WIP crate to the injection molding machine and transferring the finished goods crate away from the injection molding section, is only utilized 6.6% of the time. This is reasonable because in actually this worker is responsible for a lot more than just PVJ and RDJ part movement.

Future State Model Description

The Future State Dynamic Value Stream Map is very similar to the Current State model. The general process view of it can be seen in APPENDIX E. The DVSM view of it can be seen in APPENDIX F. The complete process breakdown for the Future State model can be seen in Table 4. One change in the Future Model is that production volumes have been changed back to levels that match Tigerpoly's goals. PVJ is 780 units and RDJ is 300 units. In turn, 780 and 300 are the number of respective units produced in the model before a changeover occurs. By default, the model also limits the WIP basket size to 30. This is lower than the 160 used in the actual process or the 150 used in the Current State model. This value was chosen because it matches the crate size for the finished goods. Although this model is not one piece flow, this now becomes 30 piece flow and the effects of that will be tested.

The Future State Dynamic Value Stream Map (APPENDIX F) can be compared to the static Future State Value Stream Map (APPENDIX G).. In both maps, the post blow molding and post injection molding processes are shown as U-shaped cells. The creation of these U-shaped cells will be discussed later in this section. The static map shows information flow from the blow molding assembly to the injection molding machine. Since these machines are on separate ends of the plant, they have no lines of communication. The blow molders cannot let the injection molders know they have a full WIP crate waiting to be processed. Vice versa, the injection molders cannot let the blow molders that they are starved of work. By creating a line of communication between these two cells, they become one virtual cell. At a minimum, this signal should tell the injection molders that a finished crate is done at the blow molding cell. This is what is meant by the "Crate Full Signal." Besides the general differences between DVSM and Static VSM's discussed in the Current State Model Description section, there is one more difference between the Future maps. The static map shows a kanban card going from a finished goods supermarket to the blow molding machine. This is not modeled in the DVSM. As in the Current State Model, the Future State model produces parts based on the same principle: production demand. It is still a push system. There is a reason why a kanban system was not implemented; plain and simply a true kanban solution required a lot of extra time and information to set up. Time was at a premium in this project.   Kanbans would create a pull system, which would be a great to set up. That pull would start from Honda. In order for this kanban pull system to work, Tigerpoly would have to have a finished goods supermarket with enough daily demand and buffer stock that Honda could take what they want on a daily basis. Once Honda removed crates from the supermarket, production kanbans would be sent to the blow molding machine. The blow molding machine is the constraint because it has a larger cycle time than the injection molding machine. These kanbans would tell how much and what type of product needs to be produced in that day based on what Honda removed from the supermarket. This is a great system, and I highly recommend Tigerpoly and Honda strive to do this; however, this kanban pull system is not implemented in this model. With limited time and technical support constraints, the actual Honda pull system was never correctly set up. First iterations of the model had kanbans being sent from the injection molding machine to the blow molding machine, but that is not logical because the injection molding machine is not the constraint. These kanbans functioned off of counters and were very unreliable; often causing the model to stop production only after about a day's worth of simulation time.   Since that part of the model never functioned properly and limited information was known about Honda's actual production process (the pull), it was decided to only simulate the process from blow molding to immediately after the finished injected molded parts are transferred to the finished goods area. Thus, the Future State model is still a push system.

When you look at the two SimCAD DVSM's, the first difference between the Current State (APPENDIX B) and the Future State (APPENDIX F) is that the Future has a 'BM cool' process after the deflash stage. This unofficially happens in the in the real system because there usually is a stack of about 14 parts waiting to be deflashed directly after the blow molding machine. These waiting parts are put in two different racks with no real organization. A cooling process could standardize the WIP by putting the blow molded tubes on a first-in-first-out gravity rack. The rack would need to be long enough that the tube was forced to wait enough cycles so that it could cool. Since the blow molding machine cycles at 85 seconds, and the tube needs to only wait for 120 seconds to cool, the rack could be as short at three tube lengths. This step would help standardize and control WIP in the assembly cell. Table 4 has the informational breakdown of all processes in the Future State Model.

Process Name	Resource	PVJ Timing	RDJ Timing	Process Capacity	Part Output
Blow Molding (B2)	n/a	85s	84s	1	PVJ_BM, RDJ_BM
BM Cool	n/a	120s	120s	1	PVJ_BM, RDJ_BM
BM Assm Cell	BM: Cycle (2 available)	Tria(113,127, 147)	Tria(113,127,1 47)	2	PVJ_BM, RDJ_BM
WIP SupMkt	n/a	0	0	1	PVJ_BM_Basket, RDJ_BM_Basket
Connection	IM	180	180	30	PVJ_BM_Basket, RDJ_BM_Basket
Withdrawal	n/a	0	0	1	PVJ_BM, RDJ_BM
Connection	IM: Cycle	1	1	1	PVJ_BM, RDJ_BM
Inj MoldI15	IM: Unload (1s)	71	80	1	PVJ_IM, RDJ_IM
IM Assm Cell	IM: Cycle (2 available)	Tria(121,135, 148)	Tria(121,135,1 48)	1	PVJ, RDJ
Transfer	n/a	0	0	1	PVJ_Tote, RDJ_Tote
Connection	Mat_Hand	300s	300s	30	PVJ_Tote, RDJ_Tote
FG SupMkt	n/a	0	0	30	PVJ, RDJ
To Honda	n/a	0	0	30	PVJ, RDJ

Table 4: Detailed information for each process in the Future State model.

The next difference between the two maps is that the Future map combines the 'Deflash', 'Cut', 'Weld1', and 'Assembly' processes into one process. This becomes the

‘BM Assm Cell’ process. This can be seen as a U-shaped cell in the Value Stream Map view. The theory behind this is that operator movement and WIP can be reduced in the cell if a U-shaped cell is set up. To do this, two workers would still be needed. Instead of splitting the responsibilities between the two workers though, the workers would follow the blow molded tube from the blow molded machine until the final assembly. This way, each worker is assigned a part that they have to take from start to finish. This is typical Lean strategy. It is theoretically possible because the total processing time post blow molding is on an average 127 seconds and at most 147 seconds. The processing time for two PVJ or RDJ parts is 170 seconds (85s times 2). So if the workers alternated parts, they would have 170 seconds to complete there part before the blow molded machine had another part for them. This U-shaped cell will reduce in process inventory because only two units can be processed in the cell at time and it will also evenly share work load between the workers. Tigerpoly would have to work out some other details to actually implement this, but in terms of a Lean standpoint it makes a lot of sense. A Mockup of the possible setup is located in Figure 6.

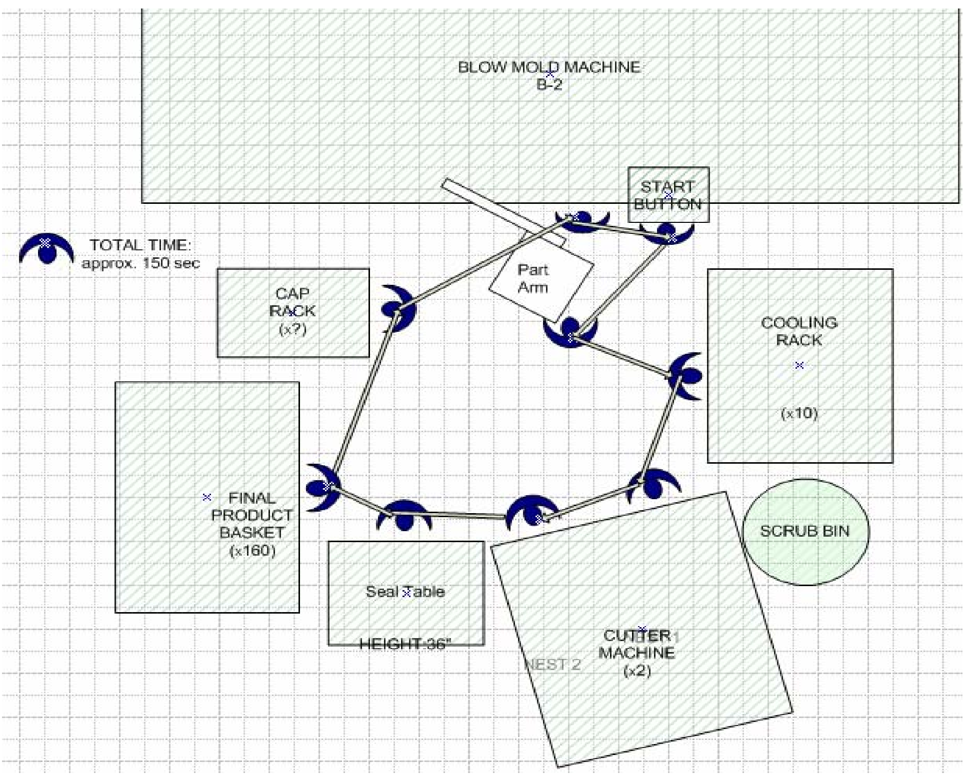


Figure 6: Possible setup for U-shaped cell which implements the Lean strategy of part ownership. [Source: Bullock, et al, Ohio State University, 2006]

Next, the Future DVSM shows a WIP Supermarket. In the model, this is not actually a supermarket. It was supposed to be a location for buffer stock between the blow molded machine and the injection molded machine. This buffer would have been tested to see how large it should have been to accommodate for production variability. However, because of time and resource constraints, this was not modeled. This process does however serve the purpose of batching the blow molded parts into the 'WIP_basket'. Objects are loaded into the basket until it is filled to the 30 unit capacity. After that, the objects are all transferred to the withdrawal process by the material handler. At the next process, 'Withdrawal', the objects are un-batched and put into a queue. They are now ready to be processed into the IM Assm Cell.

In the Future DVSM, it can be seen that the individual injection molding steps that occurred in the Current State Map (APPENDIX B) have been combined into one U-Shaped cell for the future. This is very similar to the blow molding U-shaped cell. Since the total post processing time (average 135) is less than twice the cycle time of injection molding (PVJ 142, RDJ 160), this U-shaped cell is possible. However, setting up this cell would be more difficult then the blow molding cell because some shared work and part sharing would still more than likely be required while one operator is waiting during the welding processing time. The operator waiting for the chamber to weld to the tube would have to help with a small portion of assembly with the most recent injection molded part. This part sharing could be eliminated it the process was made more efficient and work was eliminated (perhaps with an ergonomic movement reduction focus). This could make the U-shaped possible. The cell design is not a major portion of the project though; it is just in the Future State VSM to show that it is an option.

The final difference with between the Future State model and the Current State model is how they handle changeovers. In the Current State, the blow molding machine changes over parts once it reaches its daily demand for a part; i.e. when either 780 PVJ_BM's or 300 RDJ_BM's have been produced consecutively. The injection molding machine starts the changeover when the first new part is pulled from the WIP crate. There is a chance that the injection molding machine has finished all of the parts of a given run and it is just waiting to changeover until it receives the first part of the new run.

This is clearly not the most efficient way to do this. In the Future State model, the blow molded changeover happens the same way, but the injection molding changeover is handled differently. The ‘Withdrawal’ process keeps track of parts that have been pulled from it using counters (PVJ_count2 and RDJ_count2). Once the counters reach the daily demand for a part, the code recognizes that the next parts to come will be the first parts of a new run. So as soon as the last part is pulled, it creates a part called ‘Change_Signal’ in the injection molding machine. When the injection molding machine begins to process this part, it recognizes it as not being a normal part and shuts the machine down to changeover. This part is scrapped afterward as to not affect the rest of the simulation. Once the changeover occurs, the injection molding machine is ready to process the next parts of the new batch. The coding for this can be seen in APPENDIX H.

Future State Model Results

The Future State model was run for five runs at its default settings; crate size was 30 and the material handler was on the same schedule as the rest of the workers. Table 5 shows the production data from the Finished State model versus the production data from the Current State model which was originally shown in Table 3. The Table also shows the result of a paired two-tailed T-test to measure the probability of the two means being equal.

Current State, Batch = 150

Run	PVJ LT (s)	PVJ Obj Comp (units)	RDJ LT (s)	RDJ Ob Comp (units)	End WIP (units)
1	22597	2160	27579	600	161
2	23317	1500	29598	450	171
3	22539	1350	27293	300	171
4	22600	2100	27144	600	145
5	22142	1350	31214	300	171
Avg	**22639**	**1692**	**28565.6**	**450**	**163.8**

Future State, Batch = 30

Run	PVJ LT	PVJ Obj Comp	RDJ LT	RDJ Ob Comp	End WIP
1	8640	1710	10247	600	39
2	8589	1710	10682	600	60
3	8205	1560	11480	600	76
4	8888	1710	10615	600	53
5	8641	1770	10331	600	54

Avg	8592.6	1692	10671	600	56.4
T-Test Results for Paired Two-Tailed Test of Current State vs. Future State					
P()	0.00%	100.00%	0.00%	8.90%	0.01%

Table 5: Current State Values compared to Future State values.

As can be seen from this table and proved by the T-test, the lead time values for both PVJ and RDJ and the End WIP values are significantly lower in the Future State versus the Current State. This should happen because the Future State should improve upon the Current State by reducing lead time and WIP in the system. Also, the Future State values for completed RDJ units are statistically greater than the Current State values. However, what is worrisome is that the completed PVJ objects are exactly equal for Current and Future States. This should not happen. If the changes that were supposed to happen to the system were all functioning properly, the number of PVJ objects created in the Future State should exceed the number created in the Current State. Since this is not the case, the reasonable conclusion is that something is wrong with the model. This model does prove that the Future State lead times and WIP inventory are significantly lower than the Current State. It does not however prove that the Future State will have more production output; because of this, this model will not be used to give results and justifications to Tigerpoly. Instead, a modified version of the Current State model will be used to achieve more accurate data.

Modified Current State Model Description

Since the Future State Map is giving data that is not of statistical significant difference from the Current State map, it was determined that modifying the Current State map and running an experiment on that would be more worthwhile. First, however, it has to be determined whether the modified Current State map yields better results than the Future State map. In the modified map, the process flow was kept exactly the same; there were only two changes made. The first change was reducing the default WIP crate size to 30. The second change was changing the production volumes back to the real numbers (780 PVJ units instead of 750 and 300 RDJ units). Theoretically, with these changes, the modified Current State model should be very close to the Future State model. In fact, it now is a new Future State model. However, in order to limit confusion it will be referred to as the modified Current State. The one difference remaining between modified Current State and Future State is the way changeovers happen. In the modified Current map, changeovers still happen as they did in the original map. However, using this modified Current State model is by no means a negative. Technically, the Future State map was a modified Current State map, only somehow some of those modifications affected the production process. The Future State had extra noise in it. Now, with the modified Current State, it can be proved that testable inputs affect the outputs and that the results are not interfered with by the noise of the model. First, however, it is necessary to prove that the modified model is different and delivers better results than the original model. The data from Table 6 proves this. Across the board, the modified values are better than the original values and they are proven statistically different from the T-test.

Current State, Batch = 150

Run	PVJ LT (s)	PVJ Obj Comp (units)	RDJ LT (s)	RDJ Ob Comp (units)	End WIP (units)
1	22597	2160	27579	600	161
2	23317	1500	29598	450	171
3	22539	1350	27293	300	171
4	22600	2100	27144	600	145
5	22142	1350	31214	300	171
Avg	**22639**	**1692**	**28565.6**	**450**	**163.8**

Modified Current State, Batch = 30					
Run	**PVJ LT**	**PVJ Obj Comp**	**RDJ LT**	**RDJ Ob Comp**	**End WIP**
1	7506	2070	8304	600	40
2	7300	1500	8245	570	51
3	6906	1500	8140	570	51
4	7138	2250	9934	600	74
5	7217	2250	9777	660	69
Avg	**7213.4**	**1914**	**8880**	**600**	**57**

T-Test Results for Paired Two-Tailed Test of Current State vs. Modified Current State					
P()	0.00%	27.48%	0.00%	10.65%	0.04%

Table 6: Comparison of process values of the Current State to the modified Current State model.

Modified Current State Model DOE Results

The modified version of the Current State model was used for a Design of Experiment to see what relationship material handler availability and WIP Batch Size have on product lead time (both PVJ and RDJ), on completed products (PVJ and RDJ), and on WIP in the system. Since production demand constrains WIP batch size, the only batch sizes that were tested were 30 and 60. Table 7 shows the Key Input Variables (KIV) in their randomized run order and Key Output Variables (KOV) to be tested in the DOE. The tool to evaluate the effects of the KIV's on the KOV's will be regression. The KOV's will be evaluated one at a time. Table 8 shows the data collected from the DOE. The values are based on an average of two runs. APPENDIX I contains all data from all of the SimCAD runs.

Key Input Variables

Run	Material Handler Availability Percentage	WIP Batch Size
1	50%	30
2	75%	30
3	25%	60
4	100%	30
5	75%	60
6	50%	60
7	25%	30
8	100%	60

Key Output Variables

1	PVJ Lead Time
2	PVJ Parts Completed
3	RDJ Lead Time
4	RDJ Products Completed
5	End WIP Amount

Table 7: KIV's and KOV's for the DOE.

Run	Mat_Hand Av%	Crate Size	PVJ LT (s)	PVJ Obj Comp (units)	RDJ LT (s)	RDJ Ob Comp (units)	End WIP (units)
1	50%	30	7381.5	2325	9001	645	43.5
2	75%	30	7287	2310	9478	600	40
3	25%	60	11774	1830	14592	570	88.5
4	100%	30	6973	1935	8870.5	585	51
5	75%	60	10950.5	2240.5	14638.5	600	72
6	50%	60	11283	1890	12789.5	570	73
7	25%	30	8208	2055	9615	600	63
8	100%	60	11469.5	1770	13286	570	79

Table 8: Results from the DOE.

The first KOV to be tested is PVJ Lead Time. A regression analysis was run in MINITAB. The MINTAB analysis follows. The most important areas is highlighted.

‘The regression equation is:

PVJ LT (s) = 4186 - 1009 Mat_Hand A%_1 + 130 Crate Size

Predictor	Coef	SE Coef	T	P	VIF
Constant	4186.2	455.4	9.19	0.000	
Mat_Hand A%_1	-1009.1	420.7	-2.40	0.062	1.0
Crate Size	130.229	7.838	16.61	0.000	1.0

S = 332.559 R-Sq = 98.3% R-Sq(adj) = 97.6%

PRESS = 1702545 R-Sq(pred) = 94.63%’

This regression equation indicates increasing material handler availability and decreasing crate size reduces lead time. This is intuitively reasonable. It is important to note that there are constraints on these factors. Lead time is bound from .01 (1%) to 1 (100%). Crate size has a lower bound of one. Decimal values are used in the model, so if the material handler is 100% available, 1009*1 would be subtracted from the lead time. This equation however does not capture the diminishing return factor of crate size. It suggests that the crate size can be reduced all the way to one unit. However, this is not realistic because eventually the travel times and availability of the material handler (second order relationship) will caused increased lead time. The other note of importance in this analysis is that the adjusted R^2 value is 97.6%. This is a very strong value that means the 97.6% of the variation is explained by a first order model. The normal plot of residuals is included in Figure 7. It shows that there are no outliers in the data.

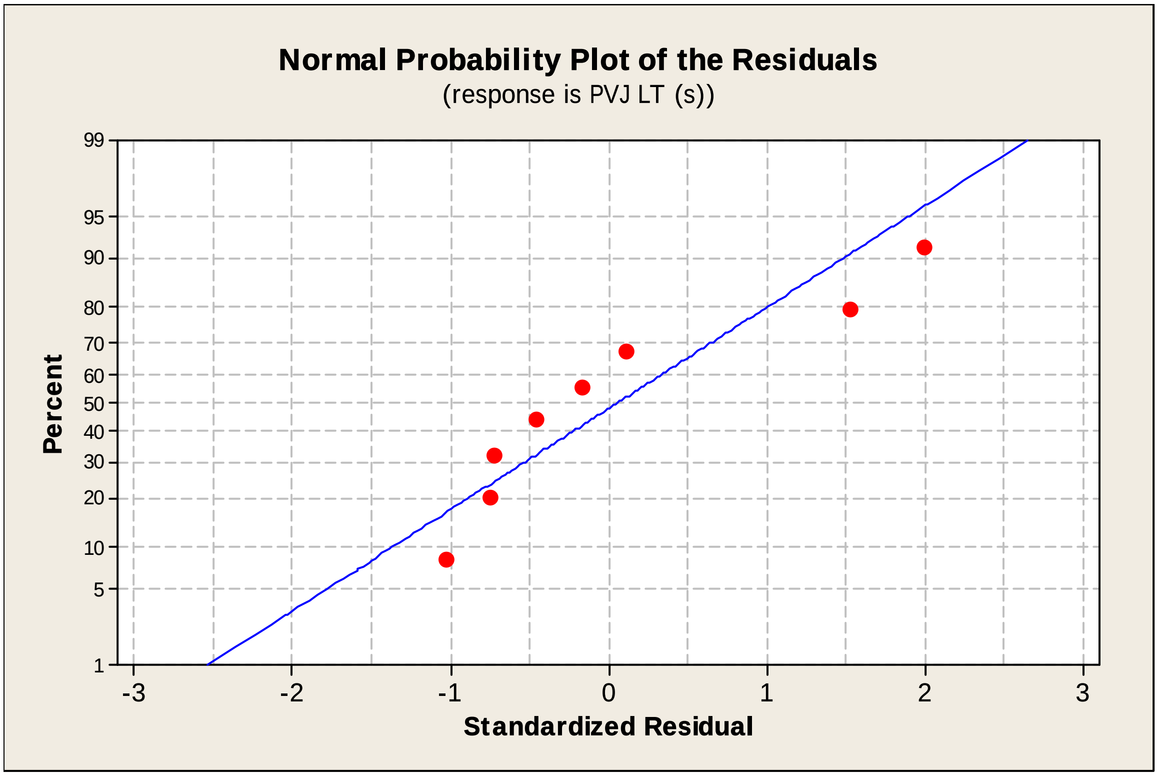


Figure 7: Normal Probability plot of the residuals for KOV PVJ Lead Time.

The second KOV to be tested is Number of PVJ Objects Completed. The MINITAB Analysis follows:

'The regression equation is:

PVJ Obj Comp (units) = 2405 - 41 Mat_Hand A%_1 - 7.45 Crate Size

Predictor	Coef	SE Coef	T	P	VIF
Constant	2405.4	302.4	7.95	0.001	
Mat_Hand A%_1	-40.9	279.3	-0.15	0.889	1.0
Crate Size	-7.454	5.205	-1.43	0.212	1.0

S = 220.842   R-Sq = 29.3%   R-Sq(adj) = 1.0%

PRESS = 622004   R-Sq(pred) = 0.00%'

This data is a lot worse than the lead time data. Essentially, with an adjusted R^2 of 1%, this regression model cannot be trusted at all. In a sense, it seems that the material handler availability and crate size do not have an identifiable first order relationship on the number of objects completed. Intuitively, this is confusing since the factors affect lead time, it would make sense that they would affect the number of objects completed. However, the data suggests that this is not the case. Objects completed must have stronger influence from factors other than the material handler and the crate size. Figure 8 shows the normal probability plot. It contains no outliers.

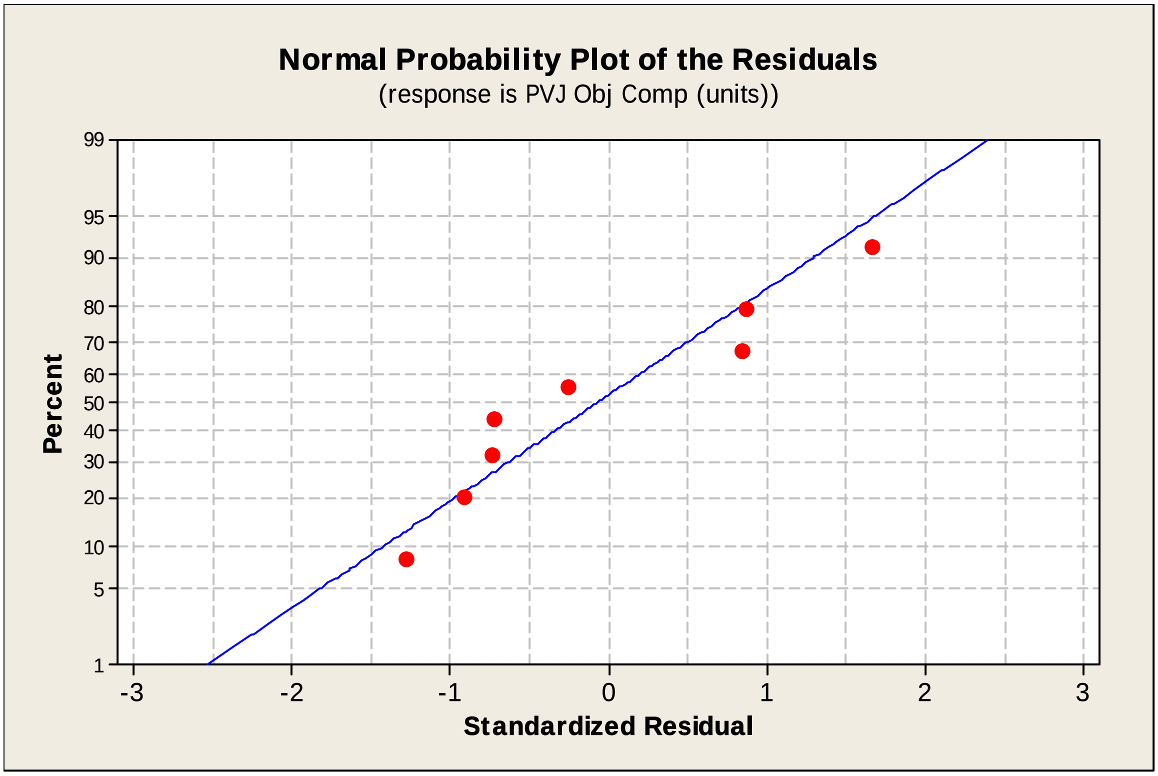


Figure 8: Normal Probability Plot of the residuals for the KOV PVJ Objects Completed.

The third KOV is RDJ Lead Time. It is expected that these results will be similar to the PVJ Lead Time results.

‘The regression equation is:

RDJ LT (s) = 5134 - 765 Mat_Hand A%_1 + 153 Crate Size

Predictor	Coef	SE Coef	T	P	VIF
Constant	5133.9	994.6	5.16	0.004	
Mat_Hand A%_1	-765.1	918.8	-0.83	0.443	1.0

Crate Size 152.85 17.12 8.93 0.000 1.0

S = 726.347 R-Sq = 94.1% R-Sq(adj) = 91.8%

PRESS = 5549950 R-Sq(pred) = 87.68%'

These results do in fact closely resemble the PVJ results. Again, by increasing material handler availability and by reducing crate size, RDJ Lead time is reduced. The normal residual plot shows no outliers.

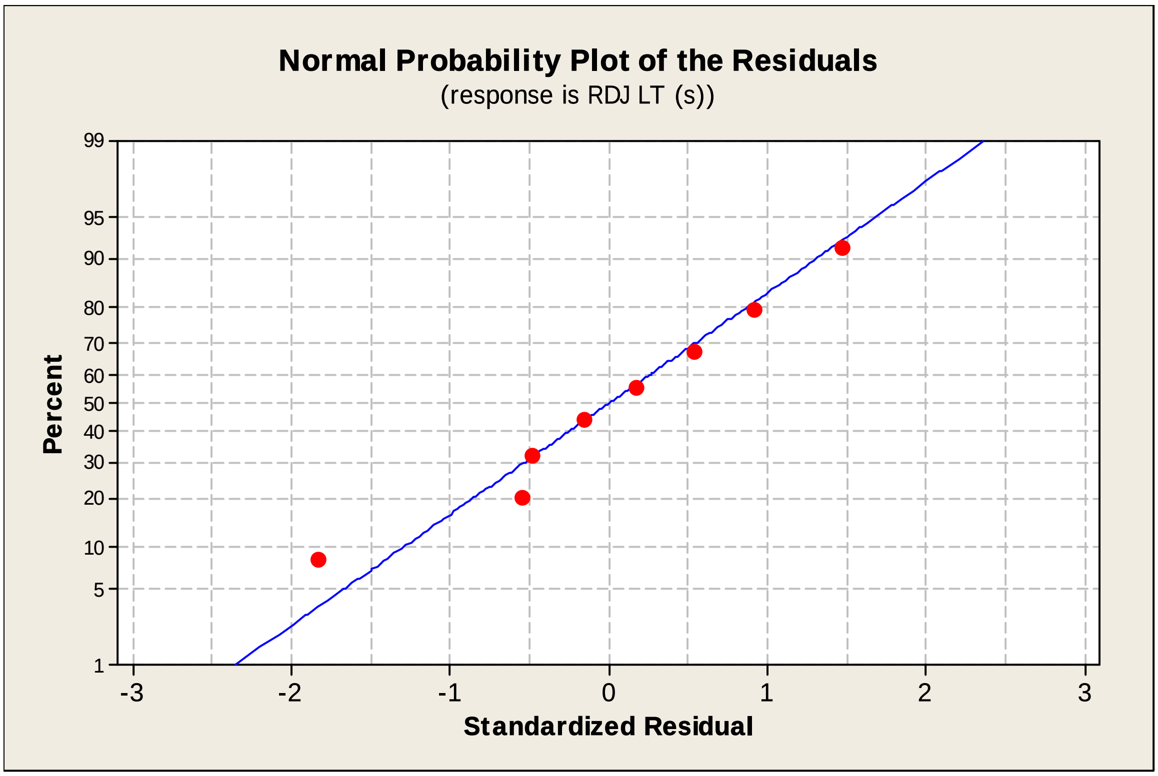


Figure 9: Normal Probability Plot of the residuals for the KOV RDJ Lead Time.

The fourth KOV is PVJ Objects Completed. The MINITAB analysis follows.

'The regression equation is:

RDJ Ob Comp (units) = 645 - 12.0 Mat_Hand A%_1 - 1.00 Crate Size

Predictor Coef SE Coef T P VIF

Constant 645.00 31.28 20.62 0.000

Mat_Hand A%_1 -12.00 28.90 -0.42 0.695 1.0

Crate Size -1.0000 0.5385 -1.86 0.122 1.0

S = 22.8473 R-Sq = 42.0% R-Sq(adj) = 18.8%

PRESS = 6037.13 R-Sq(pred) = 0.00%'

This is another very week correlation. This indicates the material handler availability and crate size do not have a strong influence on the number of objects completed. The regression plot follows; there are no outliers.

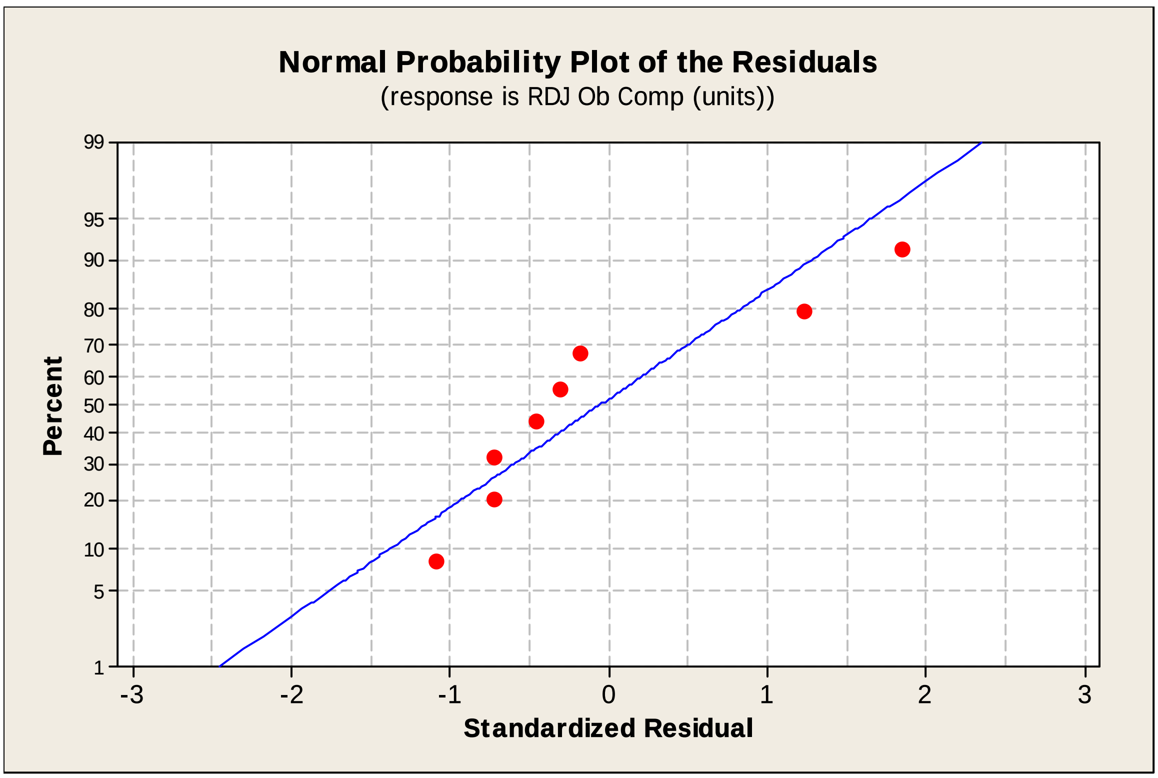


Figure 10: Normal Probability Plot of the Residuals for the KOV RDJ Objects Completed.

The fifth and final KOV to be tested is the amount of WIP in the system at the end of the simulation. It is important to note again that this is not an average, but a static measurement taken at the end of the simulation run. Thus, it may have a large amount of variability. The MINITAB analysis follows.

'The regression equation is:

End WIP (units) = 29.3 - 13.8 Mat_Hand A%_1 + 0.958 Crate Size

Predictor	Coef	SE Coef	T	P	VIF
Constant	29.25	11.68	2.50	0.054	
Mat_Hand A%_1	-13.80	10.79	-1.28	0.257	1.0
Crate Size	0.9583	0.2011	4.77	0.005	1.0

S = 8.53053   R-Sq = 83.0%   R-Sq(adj) = 76.2%

PRESS = 1005.05   R-Sq(pred) = 52.95%'

The regression equation indicates that the material handler availability has an indirect effect on the amount of WIP in the system and the crate size has a direct effect on the WIP. This makes sense intuitively because the more often the material handler is available, the more often the product gets moved which will decrease lead time. Decreasing lead time results in decreased WIP in the system. As crate size is increased, the amount of WIP increases. This makes perfect since because there are more blow molded parts waiting to be transferred to the injection molding process. More waiting parts translate to more WIP. The normal probability plot follows, there are no outliers.

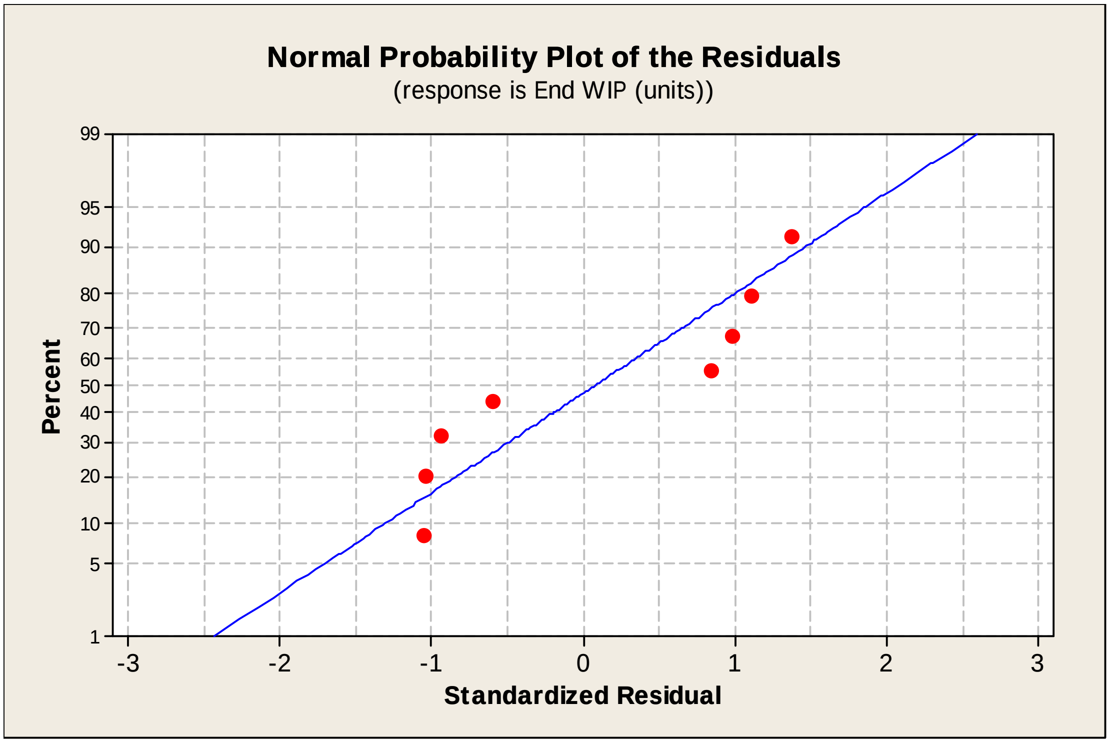


Figure 11: Normal Probability Plot of the Residuals for the WIP Level at the End of the Simulation.

Recommendations to Tigerpoly

Based on the results from the last section, it is clear that both the material handler availability and the crate size have an impact on the lead time of parts produced and the amount of WIP in the system. Let's look at the relationship between the material handler and availability first. The more that the material handler is available, the more the lead time gets reduced. However, the effect of the material handler is not too strong of a factor in the regression equation. If the PVJ Lead Time Regression equation is analyzed (PVJ LT = 4186 - 1009 Mat_Hand Av%_1 + 130 Crate Size), the material handler can only affect the lead time by 1009 seconds either way. This occurs because the material handler availability is bounded from 0 to 1. If crate size was taken out of the equation, the lowest the lead time can get is 3177s and the highest it can reach is 4186s.

Crate size, however, has a very heavy impact on the regression equation. Assuming material handler availability is not a factor (Availability = 1000%), for every one unit of crate size increase, the lead time increases by 130 seconds. The Current State lead time with the crate size equal to 150 calculates to 4186s – 1009*1 + 130*150s = 22,677s. The average from five runs of the model was 22,639s. In comparison, if we look at the Modified Current State lead time with the crate size equal to 30, the value calculates to 4186 – 1009*1+ 130*30 = 7077s. The average for two runs of the Modified Current State is 6,973s. It can be seen how strong of a correlation these factors have because the regression values and the model average values are both within 100s of each other. The actual model data shows a PVJ lead time reduction of 69% when crate size is reduced. It should be noted though that the model data is a reduction from 150 to 30 units. Using the regression equation, reducing the crate size from 160 (which is the real current value) to 30 (which is the suggested future value) reduces the product lead time by 70.4%.

The RDJ lead time regression equation is (RDJ LT = 5134 - 765 Mat_Hand Av%_1 + 153 Crate Size). The Current State lead time with the crate size equal to 150 calculates to 5134 – 767*1 + 153*150 = 27,317s. The average from five runs of the model was 28,565s. In comparison, if we look at the Modified Current State lead time with the crate size equal to 30, the value calculates to 5134 – 767*1 + 153*30 = 8,957s. The average for two runs of the Modified Current State is 8,870s. The actual model data, with crate size reduction from 150 to 30, shows a 69% RDJ lead time reduction. Using the regression equation, reducing the crate size from 160 to 30 reduces the product lead time by 69.2%.

The WIP inventory level has a similar regression relationship to material handler availability and crate size. The regression equation is as follows (End WIP = 29.3 - 13.8 Mat_Hand Av%_1 + 0.958 Crate Size). Again, the material handler availability is not too much of a factor because it can only cause 13.8 units variation. The crate size is the driving factor in this regression. Assuming material handler availability is not a factor, the WIP amount for the default Current State is 29.3 – 13.8*1 + .958*150 = 159.2 units. The model average from five runs is 164 units. For the modified Current State, the WIP value is 44.2 units. The model average is 51 units. The model data shows 69% WIP reduction. Again, it should be noted that the model data is a reduction from 150 to 30 units. Using the regression equation, reducing crate size from 160 to 30 reduces the WIP levels by 74%

The results of this model should not be surprising at all. By reducing the crate size from 160 to 30 there will be a 70.4% reduction in PVJ lead time, 69.2% reduction in RDJ lead time, and a 74% reduction in WIP inventory. The final recommendation to Tigerpoly is to work hard and work quickly in order to reduce this crate size to 30. It plain and simply makes sense. Work in progress numbers go down, which saves money, and lead times go down, which means more products can be produced and money is created. The smaller crates will also eliminate the dependence that Tigerpoly currently has on forklifts. The crates are currently transferred via forklifts from the blow molding station to the injection molding station. If the crates only hold 30 parts, they would be light enough to move by a pallet jack, or even better specially designed carts on wheels. This way the material handler, or even the cell workers themselves would be able to transfer the carts by hand from cell to cell. This would eliminate the forklift maintenance costs and costs incurred by time spent waiting for a forklift. Smaller crate size should also greatly reduce potential ergonomic risks involved with pushing and pulling the large WIP crates. Manual maneuvering of the large 160 unit crates has been witnessed; this is a very dangerous activity. By shrinking the crate size, the potential for ergonomic injury would be greatly reduced. All of these costs savings should easily cover the cost of capital for the new crates.

Evaluation of my Mobile Workstation Idea

The last section had solid, quantitative results from the SimCAD simulation. The results detailed in this section are qualitative and more or less theoretical. This section will evaluate the capacity of a Mobile Value Stream Mapping Workstation. While the workstation was never actually used at Tigerpoly to model in SimCAD or show managers the VSM while on the floor, the potential is still there and I think the workstation did a good job to convey the potential. The strongest asset of the workstation is the ability to walk the Value Stream with upper level management and discuss the maps, what-if scenarios, and results. The reason why the Dynamic Value Stream Maps were never shown to Tigerpoly management while walking the Value Stream was because the

Tigerpoly managers and engineers were very busy people who were unreceptive to computer-oriented VSM's. We never had the opportunity to get together and walk the line and discuss the map and the results.

Just because the workstation was not actually used does not mean that it is not a worthwhile tool. In fact, I still firmly believe that this could be an incredibly powerful tool for creating and showcasing Value Stream Maps. I have all the tools right now; I am ready to go into another plant and start making new Dynamic Value Stream Maps. Once those maps are done, managers can walk the line with me and the intricacies of it can be discussed. The VSM's become much more tangible, more meaningful, and more accessible while viewed on the floor of the factory. Ideas that would never come out of a boring conference room will be presented, and they can be tested in minutes. Even if the coding takes a little longer than minutes, the idea is still in the air and the model can be changed to get results. I think this is an exciting tool and I definitely plan to utilize it when I enter the workforce in the near future. Value Stream Mapping does not have to be a 'pencil and paper' tool. SimCAD will keep improving as will other simulation programs like Arena, and soon enough Dynamic Value Stream Maps will be the hot new tool in Lean Thinking.

Further Research

This project could easily be extended into a Master's Thesis. A lot of work has already gone into it, but there is room for improvement and extensions. The SimCAD models can be improved. The Objects Completed values seem strange and the reasons for that can be investigated. There simply was not enough time in the project to get the SimCAD models up and running at the original intended levels. Plenty of features can be added to the models so that more "what-ifs" can be completed. The effect of kanbans, WIP buffer stock, and fluctuating production demand quickly come to mind. Also, a more realistic material handler schedule could be very useful. The model could also be used to test potential layout changes and resource interactions in the cells. The models could also become a much more powerful tool if costs were added into the model. This could include labor cost, cost of each object, and costs associated with late deliveries. This way, the model could be optimized in terms of cost by balancing the amount of labor versus the amount of WIP in the system and on time deliveries. The current SimCAD models could be improved and new features could be explored in order to give Tigerpoly more and better recommendations in their goal to becoming Lean.

The other area of this project that could be extended into a Master's Thesis would involve working with SimCAD to improve their Dynamic Value Stream Mapping capabilities.   SimCAD has many strong attributes, but it is still a new software and has room for improvement. If a trained IE was to partner with SimCAD in order to capture what the user really wants out of a Value Stream Mapping program and how that program should accomplish it, a very valuable and easy to use tool could be created. For more information on SimCAD's strengths and weaknesses please see APPENDIX J.

Another potential area for expansion is working on the mobility of the workstation. In fact, a workstation may not even be necessary. As technology continues to advance, computers keep adding more features while becoming smaller and smaller. The potential of using Dynamic Value Stream Maps could be investigated using tablet PC's in which you can physically draw on the map and do simulations. Also, viewing and coding DVSM's on a palm pilot could be investigated. These are both viable options in the future. A large touch screen or LCD monitor could also be investigated for displaying the DVSM on the cart.

Conclusion

This project had a lot of components to it. It included learning a great deal about a real world process, learning new software, modeling that process in software, testing alternate scenarios and achieving results, and building an innovative tool to showcase these results. There were a good amount of stakeholders in this project including myself, Tigerpoly Inc., and CreateASoft Inc. I can honestly say that I am excited about the potential for Dynamic Value Stream Maps. I think they can be an extremely powerful tool that can revolutionize Lean Thinking. DVSM's still have a ways to go until they can be utilized on a mainstream level, but I think they will get there, and I want to be part of that process.

References

Bullock, M., Murrison, M., Xiong, P. "Group2-1 Cell Design Progress Update." Ohio State University Industrial and Systems Engineering Students, January,2006.

McDonald, T., E. Van Aken, R. Butler, and A. Rentes, 2000, "Integration of Simulation and Value Stream Mapping in Transformation to Lean Production." IIE Annual Conference, IIE Online Resource Library: http://www.iienet.org/login.cfm?nextpage=/resourcelibrary/index.cfm, February, 2005.

Rother, M., and J. Shook, 2003 *Learning to See: Value-stream mapping to create value and eliminate muda*. Brookline, MA: The Lean Enterprise Institute.

Womack, J.P. and D.T. Jones, 2003, *Lean Thinking: Banish Waste and Create Wealth in Your Corporation.* New York, NY: Free Press.

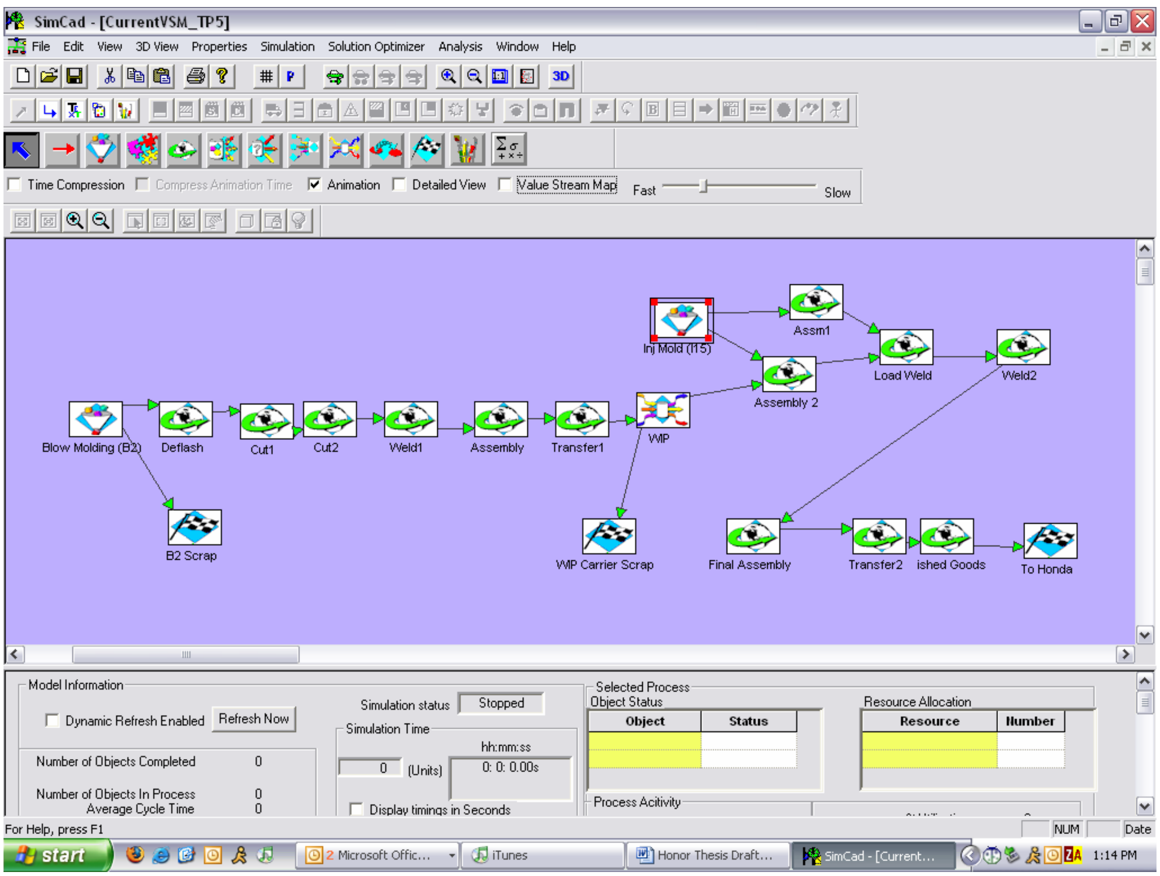
Website Sources

http://www.gilmorekramer.com//more_info/black_tuffy_utility_carts/black_tuffy_utility_carts.shtml

http://www.listaintl.com/files/compcab.htm

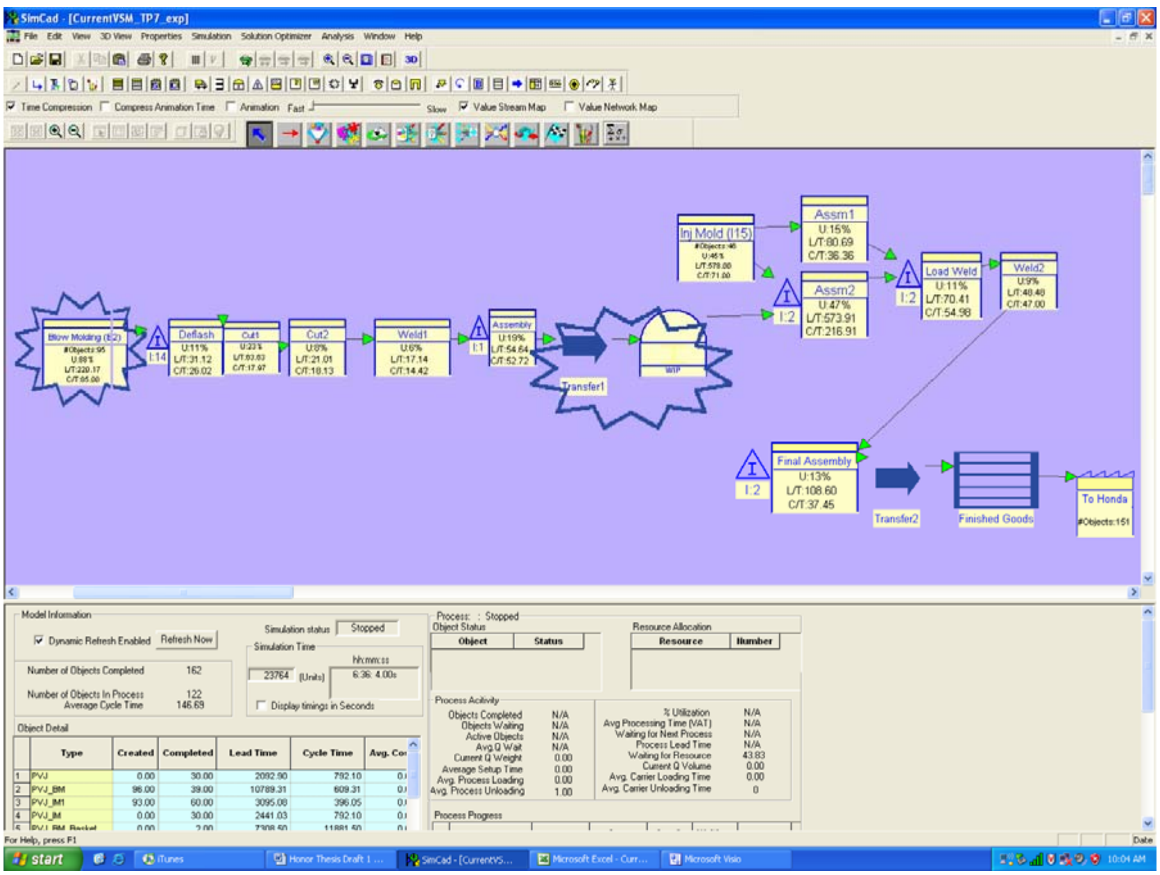
APPENDIX A:

Screen capture from SimCAD of the Current State PVJ and RDJ production process. This is the whole model in standard view. It can be noted that while the processes serve very different purposes, most of them are Generic Process boxes because SimCAD allows most of the parameters of the Generic Process to be changed. The exceptions are the Start, End, and one Routing Process.



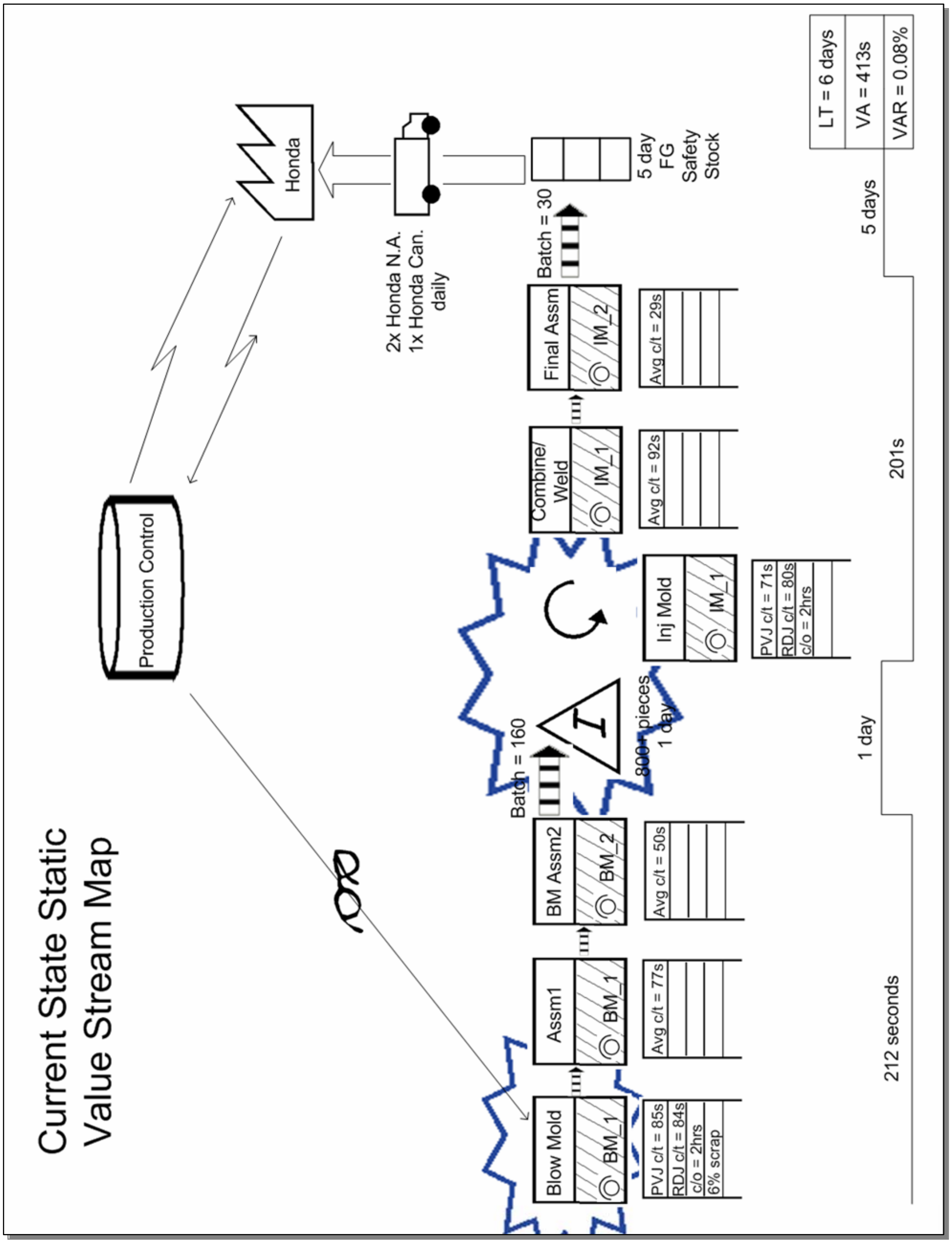
APPENDIX B:

Screen capture from SimCAD of the Current State PVJ and RDJ production process in the Value Stream Mapping view. This is the exact same model it just has some of the finish processes hidden. These processes were necessary for the accurate modeling, but do not necessary exist in the real system. Inside of the process boxes, real time simulation data such as objects completed, utilization percentage, cycle time, and lead time can be seen.



APPENDIX C:

Static Current State Value Stream Map drawn with Microsoft Visio. This is used for comparison purposes with the SimCAD Current State DVSM.



APPENDIX D:

Model coding and explanation for the Current State Model. This is a modified version of a SimCAD report.

Defined Variables

Model Variables	**Type**	**Start Value**
ModelID	Integer	0
PreviousObject	String	PVJ_IM1
PVJ_Count	Integer	0
RDJ_Count	Integer	999
WIP_tracker	Integer	0

Process/Connections Extensions

B2 Scrap

	ObjectActivated		
		Expression	HoldObject =

Blow Molding (B2)

	ObjectActivated		
		Condition	IF (PVJ_Count ==0) is TRUE
		Expression	ObjectSetupTime = 7200
		Condition	END IF
		Condition	IF (RDJ_Count == 0) is TRUE
		Expression	ObjectSetupTime = 7200
		Condition	END IF

Explanation: This assigns the changeover time to the Blow Molding Process once the daily demand is filled.

Connection:	**Blow Molding (B2)**	**Deflash**	
	ObjectActivated		
		Expression	WIP_tracker = WIP_tracker + 1
		Condition	IF (PVJ_Count < 749) is TRUE
		Expression	= CreateObjectInProcess(GetProcessID(ModelID, 'Blow Molding (B2)'), 'PVJ_BM', 1, 50

			)
		Expression	PVJ_Count = PVJ_Count + 1
		Condition	END IF
		Condition	IF (RDJ_Count < 299) is TRUE
			= CreateObjectInProcess(GetProcessID(ModelID, 'Blow Molding (B2)'), 'RDJ_BM', 1, 50
		Expression	)
		Expression	RDJ_Count = RDJ_Count + 1
		Condition	END IF
		Condition	IF (PVJ_Count == 749) is TRUE
		Expression	PVJ_Count = 999
		Expression	RDJ_Count = 0
			= CreateObjectInProcess(GetProcessID(ModelID, 'Blow Molding (B2)'), 'RDJ_BM', 1, 50
		Expression	)
		Condition	END IF
		Condition	IF (RDJ_Count == 299) is TRUE
		Expression	RDJ_Count = 999
		Expression	PVJ_Count = 0
			= CreateObjectInProcess(GetProcessID(ModelID, 'Blow Molding (B2)'), 'PVJ_BM', 1, 50
		Expression	)
		Condition	END IF

Explanation: Increments the counters (PVJ_Count and RDJ_Count) when an object is activated in the connection between Blow Molding and Deflash and creates another object in the Blow Molding process if daily demand has yet to be filled. If an object is fulfills the demand, it resets the counters and creates an object of the alternate type. It also increments WIP_tracker counter for every part that enters the system.

Connection:	**Blow Molding (B2) ObjectActivated**	**B2 Scrap**	
		Condition	IF (PVJ_Count < 750) is TRUE
			= CreateObjectInProcess(GetProcessID(ModelID, 'Blow Molding (B2)'), 'PVJ_BM', 1, 50
		Expression	)
		Condition	END IF
		Condition	IF (RDJ_Count < 300) is TRUE

Expression	= CreateObjectInProcess(GetProcessID(ModelID, 'Blow Molding (B2)'), 'RDJ_BM', 1, 50)
Condition	END IF

Explanation: Creates the correct object type in Blow Molding if an object is scrapped.

Inj Mold (I15)

ObjectActivated

Condition	IF (CompareStr(ObjectType, PreviousObject)) is TRUE
Expression	ObjectSetupTime = 0
Condition	ELSE
Expression	ObjectSetupTime = 7200
Condition	END IF
Expression	PreviousObject = ObjectType

Explanation: When a object is activated, this checks to see if it is the same as the previous object. If it is not, it assigns a setup time to the Injection Molding Machine.

WIP

ObjectActivated

Condition	IF (CompareStr(ObjectType, 'PVJ_BM')) is TRUE
Expression	= CreateObjectInProcess(GetProcessID(ModelID, 'Inj Mold (I15)'), 'PVJ_IM1', 1, 50)
Condition	END IF
Condition	IF (CompareStr(ObjectType, 'RDJ_BM')) is TRUE
Expression	= CreateObjectInProcess(GetProcessID(ModelID, 'Inj Mold (I15)'), 'RDJ_IM1', 1, 50)
Condition	END IF
Condition	IF () is TRUE

Explanation: Creates the correct object in the Injection Molding process when an object is activated in the WIP process.

WIP

ObjectActivated

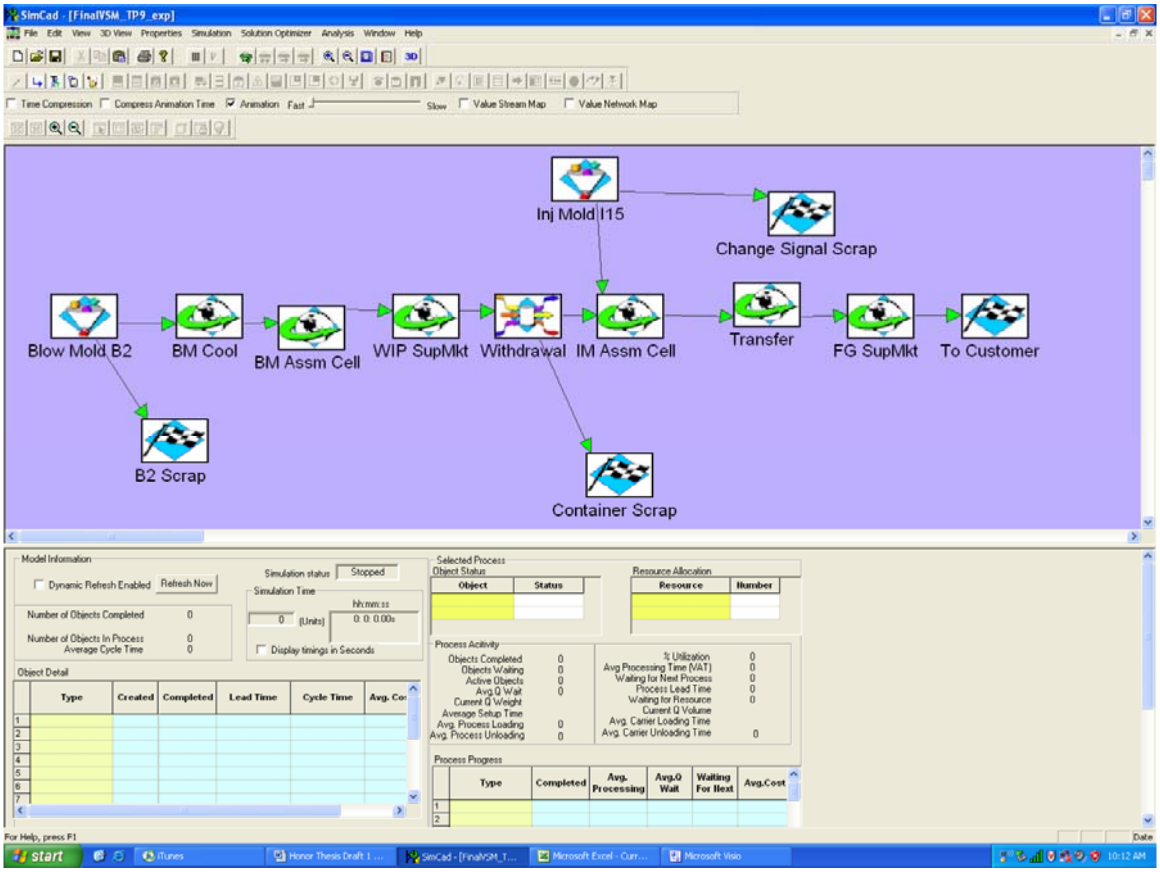
Condition	IF (CompareStr(ObjectType, 'PVJ_BM')) is

	TRUE
Expression	= CreateObjectInProcess(GetProcessID(ModelID, 'Inj Mold (I15)'), 'PVJ_IM1', 1, 50)
Condition	END IF
Condition	IF (CompareStr(ObjectType, 'RDJ_BM')) is TRUE
Expression	= CreateObjectInProcess(GetProcessID(ModelID, 'Inj Mold (I15)'), 'RDJ_IM1', 1, 50)
Condition	END IF
Condition	IF () is TRUE

Explantation: Decrements the WIP_tracker variable when a PVJ or RDJ part exits the sytem.

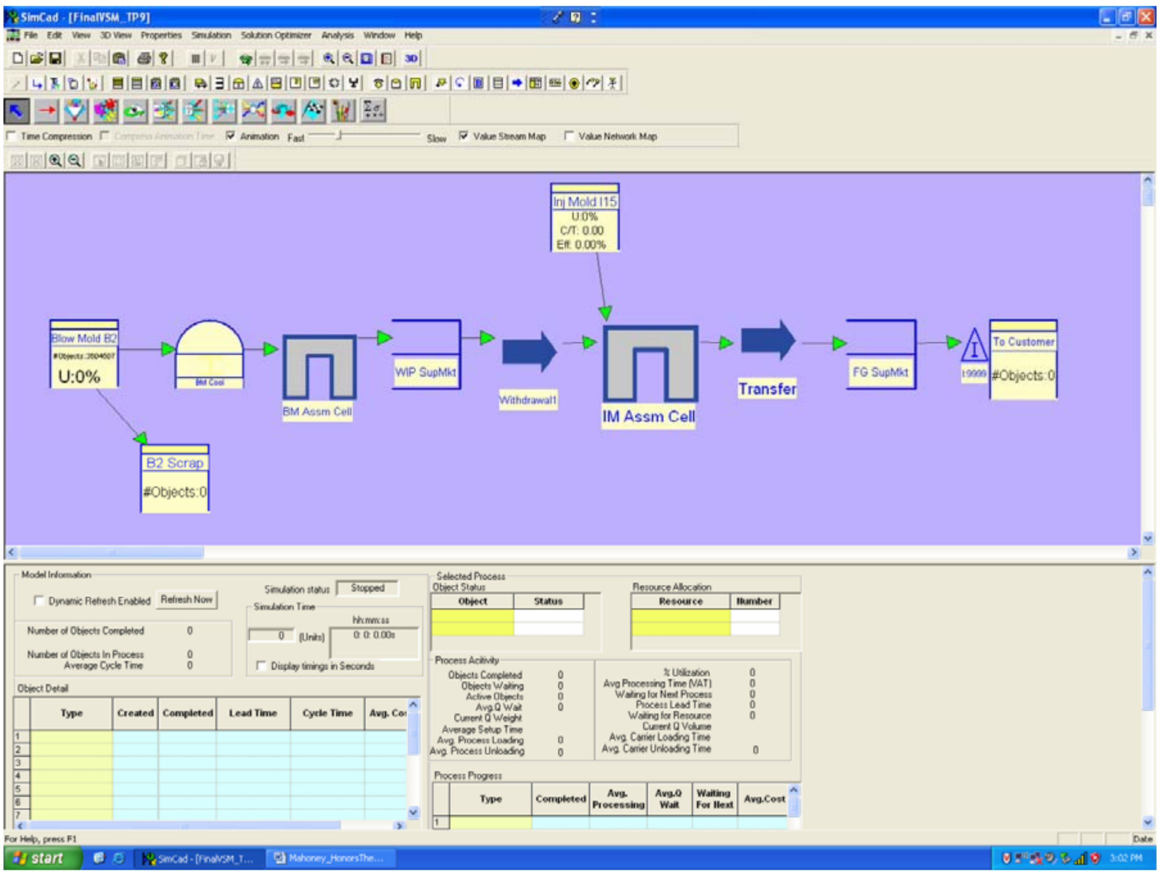
APPENDIX E:

Screen capture from SimCAD of the Future State PVJ and RDJ production process. This is the whole model in standard view. It can be noted that while the processes serve very different purposes, most of them are Generic Process boxes because SimCAD allows most of the parameters of the Generic Process to be changed. The exceptions are the Start, End, and one Routing Process.



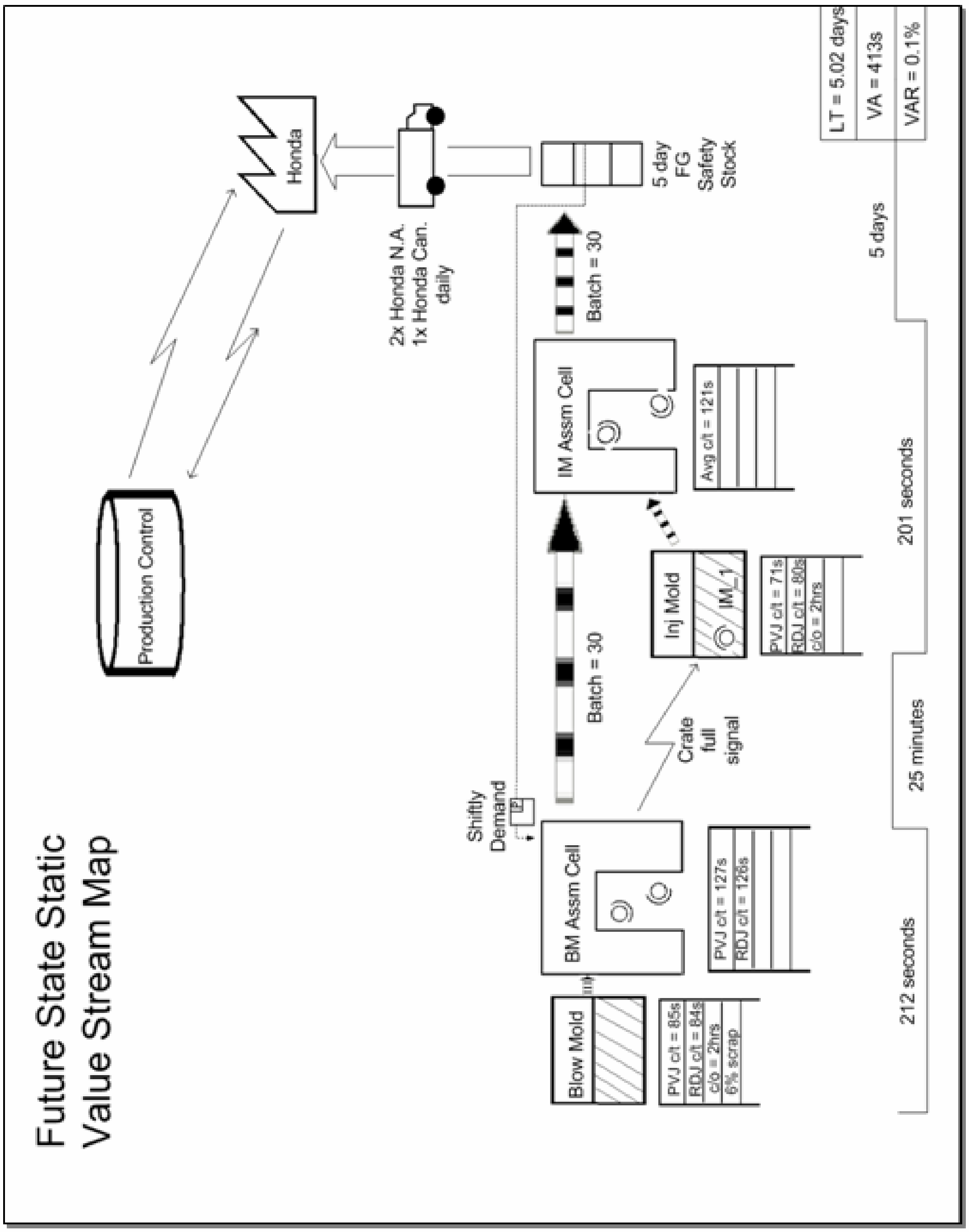
APPENDIX F:

Screen capture from SimCAD of the Future State PVJ and RDJ production process in the Value Stream Mapping view. This is the exact same model it just has some of the finish processes hidden. These processes were necessary for the accurate modeling, but do not necessary exist in the real system. Inside of the process boxes, real time simulation data such as objects completed, utilization percentage, cycle time, and lead time can be seen.



APPENDIX G:

Static Future State Value Stream Map drawn with Microsoft Visio. This is used for comparison purposes with the SimCAD Current State DVSM.



APPENDIX H:

Model coding and explanation for the Future State Model. This is a modified version of a default SimCAD report.

Defined Variables

Model Variables	**Type**	**Start Value**
ModelID	Integer	0
PVJ_Count	Integer	0
PVJ_Count2	Integer	0
RDJ_Count	Integer	999
RDJ_Count2	Integer	999
WIP_tracker	Integer	0

Process/Connections

Extensions

Blow Mold B2

	ObjectActivated		
		Condition	IF (PVJ_Count == 0) is TRUE
		Expression	ObjectSetupTime = 7200
		Condition	END IF
		Condition	IF (RDJ_Count == 0) is TRUE
		Expression	ObjectSetupTime = 7200
		Condition	END IF

Explanation: This assigns the changeover time to the Blow Molding Process once the daily demand is filled.

Connection:	**Blow Mold B2**	**B2 Scrap**	
	ObjectActivated		
		Condition	IF (PVJ_Count < 780) is TRUE
		Expression	= CreateObjectInProcess(GetProcessID(ModelID, 'Blow Mold B2'), 'PVJ_BM', 1, 50)
		Condition	END IF
		Condition	IF (RDJ_Count < 300) is TRUE
		Expression	= CreateObjectInProcess(

			GetProcessID(ModelID, 'Blow Mold B2'), 'RDJ_BM', 1, 50)
		Condition	END IF

Explanation: Creates the correct object type in Blow Molding if an object is scrapped.

Connection:	**Blow Mold B2 ObjectActivated**	**BM Cool**	
		Expression	WIP_tracker = WIP_tracker + 1
		Condition	IF (PVJ_Count < 779) is TRUE
		Expression	PVJ_Count = PVJ_Count + 1
		Expression	= CreateObjectInProcess(GetProcessID(ModelID, 'Blow Mold B2'), 'PVJ_BM', 1, 50)
		Condition	END IF
		Condition	IF (RDJ_Count < 299) is TRUE
		Expression	RDJ_Count = RDJ_Count + 1
		Expression	= CreateObjectInProcess(GetProcessID(ModelID, 'Blow Mold B2'), 'RDJ_BM', 1, 50)
		Condition	END IF
		Condition	IF (PVJ_Count == 779) is TRUE
		Expression	PVJ_Count = 999
		Expression	RDJ_Count = 0
		Expression	= CreateObjectInProcess(GetProcessID(ModelID, 'Blow Mold B2'), 'RDJ_BM', 1, 50)
		Condition	END IF
		Condition	IF (RDJ_Count == 299) is TRUE
		Expression	RDJ_Count = 999
		Expression	PVJ_Count = 0
		Expression	= CreateObjectInProcess(GetProcessID(ModelID, 'Blow Mold B2'), 'PVJ_BM', 1, 50)
		Condition	END IF

Explanation: Increments the counters (PVJ_Count and RDJ_Count) when an object is activated in the connection between Blow Molding and BM Cool and creates another object in the Blow Molding process if daily demand has yet to be filled. If an object is fulfills the demand, it resets the counters and creates an object of the alternate type. Note demand in the finished good state matches actual demand with PVJ = 780 and RDJ = 300. It also increments the WIP_tracker counter for every part that enters the system.

		To	
Connection:	**FG SupMkt**	**Customer**	
	ObjectActivated		
		Condition	IF (CompareStr(ObjectType, 'PVJ')) is TRUE
		Expression	WIP_tracker = WIP_tracker - 1
		Condition	ELSE IF (CompareStr(ObjectType, 'RDJ')) is TRUE
		Expression	WIP_tracker = WIP_tracker + 1
		Condition	END IF

Explantation: Decrements the WIP_tracker variable when a PVJ or RDJ part exits the system.

Inj Mold I15			
	ObjectActivated		
		Condition	IF (CompareStr(ObjectType, 'Change_Signal')) is TRUE
		Expression	ObjectSetupTime = 7200
		Condition	ELSE
		Expression	ObjectSetupTime = 0
		Condition	END IF

Explanation: Checks to see if the object being created is type Change_Signal. If it is that mean that this object was created to start the changeover and that daily demand of one part has been met.

Connection:	**Withdrawal1**	**IM Assm Cell**	
	ObjectActivated		
		Condition	IF (PVJ_Count2 < 30) is TRUE
		Expression	PVJ_Count2 = PVJ_Count2 + 1
		Expression	= CreateObjectInProcess(GetProcessID(ModelID, 'Inj Mold I15'), 'PVJ_IM', 1, 48)
		Condition	ELSE IF (PVJ_Count2 < 779) is TRUE

Expression	PVJ_Count2 = PVJ_Count2 +1
Expression	= CreateObjectInProcess(GetProcessID(ModelID, 'Inj Mold I15'), 'PVJ_IM', 1, 50)
Condition	END IF
Condition	IF (RDJ_Count2 < 30) is TRUE
Expression	RDJ_Count2 = RDJ_Count2 + 1
Expression	= CreateObjectInProcess(GetProcessID(ModelID, 'Inj Mold I15'), 'RDJ_IM', 1, 48)
Condition	ELSE IF (RDJ_Count2 < 299) is TRUE
Expression	RDJ_Count2 = RDJ_Count2 + 1
Expression	= CreateObjectInProcess(GetProcessID(ModelID, 'Inj Mold I15'), 'RDJ_IM', 1, 50)
Condition	END IF
Condition	IF (PVJ_Count2 == 779) is TRUE
Expression	PVJ_Count2 = 999
Expression	RDJ_Count2 = 0
Expression	= CreateObjectInProcess(GetProcessID(ModelID, 'Inj Mold I15'), 'PVJ_IM', 1, 50)
Expression	= CreateObjectInProcess(GetProcessID(ModelID, 'Inj Mold I15'), 'Change_Signal', 1, 49)
Condition	END IF
Condition	IF (RDJ_Count2 == 299) is TRUE
Expression	RDJ_Count2 = 999
Expression	PVJ_Count2 = 0
Expression	= CreateObjectInProcess(GetProcessID(ModelID, 'Inj Mold I15'), 'RDJ_IM', 1, 50)
Expression	= CreateObjectInProcess(GetProcessID(ModelID, 'Inj Mold I15'), 'Change_Signal', 1, 49)
Condition	END IF

Explanation: Increments PVJ_Count2 and RDJ_Count2. These counters are used to see how many objects have been created in the injection molding cell. If daily demand has not been met, the correct part is created in the injection molding process. If it is at the beginning of the run, the part has a low priority so that it cannot jump in front of the queue in front of the change signal object. If daily demand has been met, then the 'Change_Signal' object is created in the injection molding cell which forces a changeover.

APPENDIX I:

Data from various SimCAD runs and experiments. Data includes lead times, objects completed, WIP levels, and resource utilization percentages.

Current State Model, Batch = 150, Mat Hand Av% = Worker Shift

Run	PVJ LT (s)	PVJ Obj Comp (units)	RDJ LT (s)	RDJ Ob Comp (units)	End WIP (units)	B2_1 Ut%	B2_2 Ut%	I15_1 Ut%	I15_2 Ut%	Mat_Hand Ut%
1	22597	2160	27579	600	161	68	36	66	79	9
2	23317	1500	29598	450	171	50	26	45	57	6
3	22539	1350	27293	300	171	43	22	39	47	5
4	22600	2100	27144	600	145	68	35	64	76	8
5	22142	1350	31214	300	171	43	22	39	47	5
Avg	**22639**	**1692**	**28565.6**	**450**	**163.8**	**54.4**	**28.2**	**50.6**	**61.2**	**6.6**

Future State Model, Batch = 30, Mat Hand Av% = Worker Shift

Run	PVJ LT (s)	Comp (units)	RDJ LT (s)	Comp (units)	End WIP (units)	BM U%	IM U%	Mat_Hand U%
1	8640	1710	10247	600	39	76	95	9
2	8589	1710	10682	600	60	78	96	9
3	8205	1560	11480	600	76	72	95	9
4	8888	1710	10615	600	53	77	96	9
5	8641	1770	10331	600	54	79	95	10
Avg	**8593**	**1692**	**10671**	**600**	**56**	**76**	**95**	**9**

Modified Current State Model, Batch = 30, Mat Hand Av% = Worker Shift

Run	PVJ LT (s)	PVJ Obj Comp (units)	RDJ LT (s)	RDJ Ob Comp (units)	End WIP (units)	B2_1 Ut%	B2_2 Ut%	I15_1 Ut%	I15_2 Ut%	Mat_Hand Ut%
1	7506	2070	8304	600	40	64	33	58	71	11
2	7300	1500	8245	570	51	50	26	45	56	9
3	6906	1500	8140	570	51	48	27	45	54	8
4	7138	2250	9934	600	74	66	37	62	78	12
5	7217	2250	9777	660	69	68	38	64	78	12
Avg	**7213**	**1914**	**8880**	**600**	**57**	**59**	**32**	**55**	**67**	**10**

Modified Current State Model, DOE Data

Run	Mat_Hand Av%	Crate Size	PVJ LT (s)	PVJ Obj Comp (units)	RDJ LT (s)	RDJ Ob Comp (units)	End WIP (units)	B2_1 Ut%	B2_2 Ut%	I15_1 Ut%	I15_2 Ut%	Mat_Hand Ut%
1	50%	30	7384	2340	9721	690	62	70	39	33	80	18
2	50%	30	7379	2310	8281	600	25	67	37	31	74	16
Avg	**50%**	**30**	**7381.5**	**2325**	**9001**	**645**	**43.5**	**68.5**	**38**	**32**	**77**	**17**
3	75%	30	7187	2310	9214	600	41	67	37	31	75	13
4	75%	30	7387	2310	9742	600	39	67	37	31	75	13
Avg	**75%**	**30**	**7287**	**2310**	**9478**	**600**	**40**	**67**	**37**	**31**	**75**	**13**
5	25%	60	12004	2100	15600	600	96	66	35	30	73	31
6	25%	60	11544	1560	13584	540	81	51	26	23	54	24
Avg	**25%**	**60**	**11774**	**1830**	**14592**	**570**	**88.5**	**58.5**	**30.5**	**26.5**	**63.5**	**27.5**
7	100%	30	6992	1560	8834	570	51	49	27	23	55	8
8	100%	30	6954	2310	8907	600	51	67	37	31	75	11
Avg	**100%**	**30**	**6973**	**1935**	**8870.5**	**585**	**51**	**58**	**32**	**27**	**65**	**9.5**
9	75%	60	10975	2200	14982	600	89	67	36	32	76	9
10	75%	60	10926	2281	14295	600	55	68	37	32	77	9
Avg	**75%**	**60**	**10950.5**	**2240.5**	**14638.5**	**600**	**72**	**67.5**	**36.5**	**32**	**76.5**	**9**
11	50%	60	11123	1560	12401	540	81	50	27	23	55	10
12	50%	60	11443	2220	13178	600	65	66	36	31	74	13
Avg	**50%**	**60**	**11283**	**1890**	**12789.5**	**570**	**73**	**58**	**31.5**	**27**	**64.5**	**11.5**
13	25%	30	8010	2190	9115	600	65	66	36	30	72	33
14	25%	30	8406	1920	10115	600	61	64	31	26	67	32
Avg	**25%**	**30**	**8208**	**2055**	**9615**	**600**	**63**	**65**	**33.5**	**28**	**69.5**	**32.5**
15	100%	60	11518	1980	13695	600	77	66	32	28	70	8
16	100%	60	11421	1560	12877	540	81	55	26	22	57	6
Avg	**100%**	**60**	**11469.5**	**1770**	**13286**	**570**	**79**	**60.5**	**29**	**25**	**63.5**	**7**

APPENDIX J:

This appendix contains a detailed evaluation of SimCAD; it covers both its strengths and its opportunities for improvement. It is an honest evaluation from the author of this thesis and only covers areas of SimCAD that I personally dealt with; i.e. I will not discuss the strengths and weaknesses of the 3D modeling capabilities because I did not use them. This evaluation is entirely objective and is based on the strengths and weaknesses of the SimCAD software. It is important to note that I had no official training in SimCAD; I taught myself. I am sure that I could have avoided many of the issues that I had if I went through formal training. However, I did not have the luxury of training so this evaluation is based entirely on my self-learning experience. I sincerely hope this evaluation and the recommendations are taken seriously by the employees at CreateASoft Inc. because I believe the recommendations could greatly improve the SimCAD software.

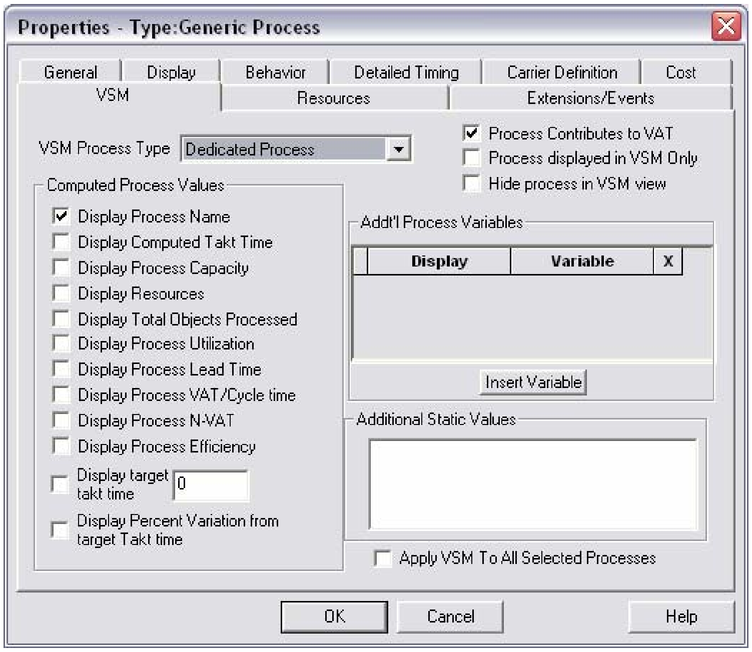
First, I will start with my simulation experience. I have previous experience with Rockwell Automation's Arena and also with Imagine That Inc.'s Extend software. I took two quarter's of ISE classes that taught simulation in Arena. Although I have never actually used it in industry, I consider myself relatively well-versed in Arena from the two-courses. Of all of the simulation programs I have used, Arena is the best. It is the most user-friendly and has intuitive, yet powerful features like OptQuest. Arena also has very good data tracking and report creation along with straight-forward debugging. However, I have heard an Arena license comes at a high cost, about $20,000. Quality clearly comes with a price. The main downside that I found using Arena was that sometimes it was necessary to use Visual Basic coding to modify the model. The next simulation program I used was Extend. I utilized this at Intel's development site while on co-op. I used Extend to build an in-depth tool-level simulation. Extend was a frustrating program to use because all aspects of the processes are very separated which requires the user to properly connect them. For example, in order to have a normal production process, a queue module has to be connected to a delay model that is fed by a statistical distribution module. This method is not intuitive at all and it also causes the model to become large. The logic and flow of the model are difficult to follow because it forces hierarchies upon hierarchies of processes. Extends debugging process is tedious and difficult. Extend's default data tracking and statistical reports are non-existent; it does however allow the user to easily control and change parameters in different runs without using Visual Basic coding. Extend also allows virtually any module to be tailored to a user's specific needs. Extend is also very cost effective; a basic Extend license only costs around $1,000, and with time, results can be achieved. Since I have used both Arena and Extend, I think I am fairly qualified to evaluate SimCAD.

I started learning and using SimCAD in January of 2006. I had the program on the laptop and I had the manual. I read the manual and started to model Tigerpoly's process. Going into this project, I thought SimCAD was a Value Stream Mapping tool that allowed for simulation. I quickly realized that SimCAD is a simulation tool that tries to accommodate Value Stream Mapping, and not the other way around. One of SimCAD's strength is in that it does allow the user to control every aspect of a process inside of the process box which is the exact opposite of Extend. This means that the process flow in a SimCAD model is very easy to follow. This suits well for Value Stream Mapping and it is how I think it should be. However, when I tried to draw information flows and other such things in SimCAD, I was unable to. SimCAD allows the creation of "dummy" processes in the VSM view; these can replicate some of the VSM drawing capabilities. Also, these "dummy" processes can be hidden from the processes view; as well as other processes can be hidden from the VSM view. However, in most cases, the user cannot interconnect "dummy" symbols to actual processes. Even if a connection is made, it cannot be differentiated from the standard connection that transfers objects. This current drawing interface needs to be made more robust.

There is another major aspect of the Value Stream Mapping portion in SimCAD that has to be improved; the data tracking. A major part of Value Stream Mapping is calculating the Value Added Ratio; the ratio of value added time over the total lead time. This value is not calculated in SimCAD and I was unable to find a way to track it. This value should be displayed, dynamically updated, and shown on the map. Also, traditional VSM's have a timeline underneath the objects that shows if the objects add to value added time or to non-value added time. Since SimCAD cannot draw, a static version of this line cannot be created. However, a static line should not be the goal. I think

SimCAD should dynamically create the time line and change the line lengths and values as the VAR changes.

Another issue I had with VSM data tracking involved data inside of the actual process boxes. There are several default metrics that can be tracked in the VSM process boxes; see below.



These values show up in the VSM in the process box as long as the VSM picture is a dedicated process. If it the process is represented by a U-Shaped Cell, the data is no longer shown. Regardless of what the process is represented by in the map, the data should still be visible. Like most normal Value Stream Maps, the data should be shown below the process symbol. If the drawing capabilities and data tracking aspects of SimCAD were improved, it would go a long way to making the models actual Dynamic Value Stream Maps. In my opinion, right now they are just simulated Process Flow Diagrams that use VSM symbols.

Along with the opportunities in terms of Value Stream Mapping, I think there are also several opportunities for improvement in the modeling. Several of these problems are serious and caused major delays in my progress while I was creating the models. Often, I had to wait upon technical support's help until I could continue my work. Oftentimes, if I did not hear back from technical support or if they did not have an answer, I either had to find a different way to accomplish the intended task, or leave that task entirely out of the model. One of the modeling portions that I had a lot of difficulty with was the resources. In the 'IM Assm Cell' module of my Future State model, I wanted each of the two resources to be able to process one item at a time. These two resources were created in the general Flow Properties of the model. The Flow Properties section contains the main modeling controls such as resource definitions, object definitions, variable definitions, etc. In the resource tab of the 'IM Assm Cell' process, there is a column called 'Required number.' I set this number to one, thinking that it meant one resource was necessary to work on one object, and the other was free to work on another object. See the figure below.

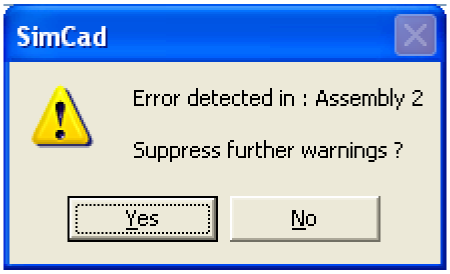
	Resource Name	Required Number	Is a Helper	%Efficiency	Cycle	Load	Unload	Setup
1	BM	0	☐	100	☑	☐	☐	☐
2	IM	1	☐	100	☑	☐	☐	☐
3	Mat_Hand	0	☐	100	☑	☐	☐	☐

However, when I set this number to one, the actually processing time ended up being twice as much as it was set to in the detailed timing tab. This extra processing time caused this model to be a bottleneck where it should not actually be. I determined that the problem was that I had to set the required number to two. Once I set that number to two, the processing time become normal. However, that does not make any sense to me. By putting a two in that column, to me it means it takes two resources to process one object, which would reduce the capacity by half. In the model, it doubled the capacity. This was very confusing for me and delayed my work.

Another major problem I had with the resources was that they get over-allocated in the wrong place quite often. They get allocated in one process, but another process needs them to do its task. This task cannot be done without the resource, but SimCAD never releases the resource. This means that objects never leave the process, and the model effectively shuts down. This occurred quite often in my model because I had resources that that were supposed to work on different processes; in the real world following the part through the different steps. This resource over-allocation most often occurred when the resource was supposed to load or unload a machine and afterward perform the cycle on the next machine. In my models, I was forced to limit resource allocation especially by not having them load or unload any machines. It over-simplified my models. These resource allocation problems also occurred frequently because of batching issues. For example, in my current state model, objects combine at ‘Load Weld’ from ‘Assm 1’ and ‘Assm 2.’ Sometimes an object from ‘Assm 2’ would enter ‘Load Weld’ process and detain the resource. However, an object in ‘Assm 1’ was waiting and required that detained resource. Of course, this creates a resource allocation problem, and no objects would move any more in the model. Other issues with batching will be covered more in depth later.

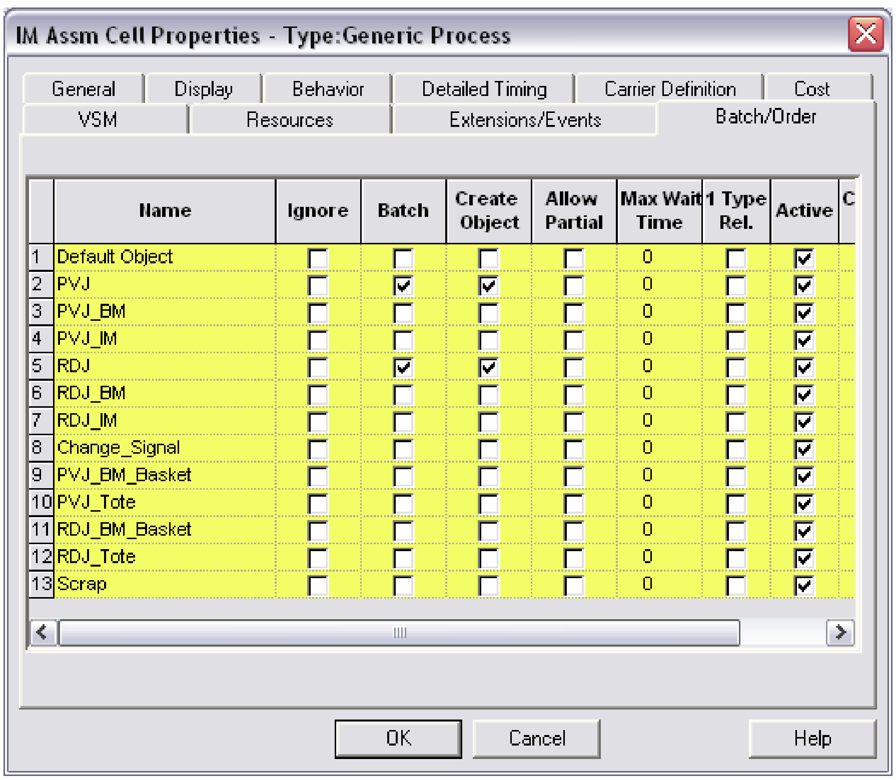
A related problem to resources being over-allocated is that once a resource becomes over-allocated, SimCAD does not even let you know that a problem has occurred. No objects move in the model, but SimCAD finishes the rest of the simulation and reports the results. The only way to check to make sure if the model ran to completion is by essentially with a judgment call of whether or not the results seem reasonable. I did it by reading the number of items completed, if they seemed low, I checked through the model to see if any resources were stuck in one spot. SimCAD should work first on informing the modeler that there is a problem in the system, and second work on eliminating these resource allocation problems.

This brings me to another debugging problem. Some SimCAD errors do stop the process, but the error messages are very vague and give very little instruction how to fix them. See below for a sample error message.



The only information it gives is the process that the error is located in. It gives no information as to what type of error it is or what is causing it. It would really help the debugging process if these error messages were more detailed and also linked to the help files for further explanation.

The next area that caused major frustration and weeks of busy work was batching items in SimCAD. In my finalized versions I did two different forms of batching. I batched items as they were assembled; so blow molded parts combined with injection molded parts to make the final parts. I also batched for the crate transfers and un-batched at the next process. I got all of these to work, but not without much lost time in trying different methods and debugging the models. First, I will talk about batching to assemble the parts. I tried to use the 'Join' process, which makes sense; however, the 'Join' process does not have the capability to make 'smart' joins. It only can combine one item from each of its connections. The 'Join' process has limited functionality compared to the 'Generic' process. In order to actually join the objects and create new objects, I had to select the option "Reassemble Objects Based on Type (Batch Definition)." This means that in the Flow Properties, I had to define the objects, and the objects that comprised them. The 'Batch/Order' tab can be seen below.

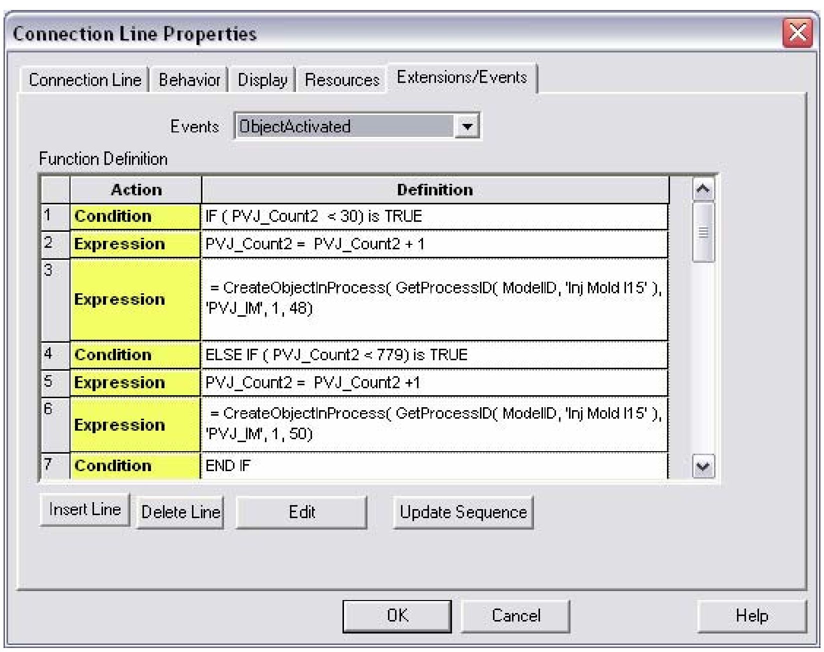


By selecting 'Batch' and 'Create Object' for the final objects, the objects get combined together and a new object is created. One problem however is that the older objects still exist and travel along with the new object. So when a PVJ object gets completed, so does its components (one PVJ_BM and one PVJ_IM). In one sense this actually helped with data collection. In order to get lead time data, the PVJ lead time could not be used because the object was created in the middle of the process. So the PVJ_BM lead time data was used because it travels through the entire Value Stream. This would have worked fine; however, the blow molding machine scraps parts by distributing 6% of objects to an ‘End’ process. These objects get counted as completed, and the lead time gets recorded. Their lead time is only about 80 seconds, which gets averaged with the lead time of objects that actually complete the whole process. This results in lower lead times for completed objects. I think when SimCAD joins objects to create new objects, there should be the option to make one encompassing object which takes the data properties of its components. In terms of lead time, it should use the lead time of its oldest component. This way, the lead time of the original component would not have to be used in data tracking. Also, the way items are scrapped needs be reworked. There is no possible reason for scrapped items to be counted in the lead time; yet I could find no way around it. The ‘Start’ process should have a built in ‘Scrap’ option where a certain percentage can just get thrown away or reworked. These scrapped items should not count in the final lead time or completed item data.

The second problem I had with batching was with large batch transfers. Eventually I did this by batching objects to create a carrier item, un-batching the items at the next step, and immediately discarding the carrier. This worked well once I found out how to do it, but originally I tried to use carriers. Carriers are objects that get combined with a certain number of other objects before they are all transferred. After the objects are unloaded from the carrier, the carrier returns on a path to the original process. This all sounds fine in practice, but using carriers is difficult and frustrating. I could not find any way to get the carriers to work with multiple items, so I had to use the batching route instead. A lot of time was wasted trying to get the carriers to work.

I had a lot of issues trying to get event extensions to work in SimCAD. An event occurs every time the status of a process, connection, or model changes. Extensions are built upon an event to make something happen in the model every time an event occurs. Extensions are the coding that underlay the models. This coding can be seen in APPENDIX D and APPENDIX H. Initially in my model, I tried to use event triggers to simulate a kanban pull as it is explained in the manual. These event triggers did not work at all, so I consulted SimCAD technical support and they accomplished the pull by setting up event extensions. They set the model to run based off of counters of 30. When the downstream queue was empty, it sent a trigger to the blow molding machine that it needed to make 30 more objects. This seemed worked in theory, but the model almost never ran to completion. The counters would always get off along the way and this would stop the objects from flowing. And when I tried to add in another part, these counters stopped functioning all together. Kanbans should be simple and easy to use; not a complicated process that requires coding using extension events. The kanban system needs to be reworked in SimCAD.

When I wanted to add a second part, I realized that the simple counter system I was using before was not adequate enough to accommodate two parts. This is when I really started to use and understand coding via extension events in SimCAD. First, I can say that extensions are handled very poorly in SimCAD. Extensions in my model are created off an event in either a process or connection. Most of them occur when an object is activated in either the process or the connection. When you look at the 'Extension/Event' tab in a process, by default it shows the extension modifiers that have been done to the 'Initialize Simulation' event. Then there is a drop down menu of all the other events that you can modify. As I mentioned before, most of my extensions were on 'Object Activate.' See below.



However, when the 'Extension/Event' tab is selected, it does not indicate what events are modified. The 'Extension/Event' tab by default should show which events have been modified and show those automatically.

It is possible to go to Analysis, then Model Information, then Extensions/Events to see a report of the event modifiers that are placed in the model. APPENDIXES D and H are modified versions of this report. This does help; however, it only views the extensions to the model. A user cannot modify the extensions from this screen. In fact, there is no central forum for modifying extensions. This central hub desperately needs to be created in SimCAD. Creating and updating extensions is a belabored process. A user has to go into each process and type line by line the extensions. Some of the coding in my model was up to 20 lines long for one event. Lets say though that the user did all of this coding on the 'Object Activate' event, but they really want it on the 'Step Started' event; the only option they have is to delete the extensions from the one and retype them in the other. There is no copy and paste option. I wasted a lot time typing lines of code, just to retype them again because I realized they were assigned to the wrong event or assigned in the wrong process. There needs be a central hub to view, modify, and add extensions in SimCAD. It would make the process a lot less complicated.

I also wasted a lot of time in the model attempting to modify extensions while the simulation was paused. I found the best way to debug was watch the model until something strange happened, maybe a resource allocation, and then pause the model. This way it can be seen how many objects each process has completed, or how many are objects are currently in the process or in the queue. Essentially, by looking at each process, the problem area can be identified. Even though the model is paused, SimCAD allows the user to go into the process boxes or extension events and make changes. However, SimCAD does not save or update these changes. Once the process box is closed SimCAD gives no indication that the changes were not saved. A user has to first stop the model to make the changes. This is fine in essence, but SimCAD should prompt the user that the model is paused and no changes will be saved. Or, even better, SimCAD should allow those changes to occur even though the model is paused because often times it is necessary to work in a paused model to debug.

I had to use extensions to model the changeovers in the 'Start' processes. I did this by using counters in the connection after the 'Start' process, and if the counter limit was reached the next object would cause a changeover in the start process. This go-around was necessary because the 'Start' process does not have setup options. The other

‘Generic’ processes do have setup options in which SimCAD allows setup times to be defined for each object. These options are not available in the start process, which to me is the process that it makes the most sense for setups to occur. Setups/changeovers should be easy to do in the ‘Start’ processes. The process should keep track of how many objects of a given type have been completed (not scrapped), and based on those numbers automatically start the changeover. There is no way it should be as complicated as I had to make it.

The other major issue I had with SimCAD was the user manual, which is also the help guide. In my opinion, it just was not good. It does a poor job of actually showing how to do things or what has to be done in order to accomplish certain tasks. As an example process setup time will be used. As I said in the last paragraph, ‘Generic’ processes can have setup times for different objects. The manual describes how these setup times can be defined for each object in a ‘Setup’ tab. However, when a ‘Generic’ process is opened, there is no setup tab. In order to get that setup tab, the ‘Use Default Process Setup Time’ box needs to be unchecked. This box is in the ‘Behavior’ tab and it is checked by default. The manual says nothing about that box, so when and new user goes to look for the setup tab, it just simply is not there which results in confusion and time lost in looking for the tab. This incomplete explanation is common in the manual. The SimCAD manual will say what it can do, but it does not do a good job of describing how to do it. There should be more concrete, detailed examples in the manual that show the user step-by-step processes in order to accomplish a task.

In SimCAD, the ‘Simulation Control’ parameters can be adjusted in the ‘Flow Properties’ menu. These parameters include model run time. However, when a user saves and closes their model, SimCAD does not save these ‘Simulation Control’ options. I had my model run for one week, but seconds were my default time unit, so it was 432,000 seconds. Every time I opened the model, I had to put in the 432,000 number. This was a major annoyance because sometimes I would forget and waste a lot of time waiting for the model to stop; which would not happen. Other preferences in SimCAD are not saved either. For instance, the run speed preference, animation preferences, or data refreshing preferences cannot be saved in SimCAD. These options need to be adjusted every time SimCAD is opened.

The next few points are more minor annoyances that I noticed in SimCAD, not major issues that have been previously discussed. First, when loading an initial state, a certain amount of objects are set to be created at time zero. The object creation starts at time zero, but objects continued to get populated in the process during the simulation time. All of these objects should be created in the process before the model can start. Second, if the model is accidentally closed, a prompt comes up for 'Save, No Save, or Cancel'. If 'Cancel' is selected, the model closes anyway with no indication of whether or not the model was saved. Third, SimCAD is not consistent in how information in the process boxes gets deleted. In some cases, there is a grey box with an 'X' above it in which the user has to 'check' that box, in other cases the user has to highlight the line and hit delete on the keyboard, and finally in yet other cases there is a 'Delete Line' box. SimCAD should try to be consistent in its deletion methods. Fourth, in the 'Detailed Timing' tab of a process box, users can assign resources to do loading and unloading of the object. However, there is no indication in the process box or in the help manual if this loading and unloading time is accounted for in the cycle time or is added extra to the processing time. Fifth, SimCAD has an optimizer function that can be used to systematically vary user-created variables. This is a very constrained optimizer. I wanted to use it to vary both crate size and material handler availability, but was able to do neither. This optimizer needs to become more robust. Sixth, shifts can be created for different operators. However, every shift is forced to be based on the same time units. I wanted to create a default hour shift for the regular workers and a repeating minute shift for the material handlers, but was unable to because both shifts have to be on the same time interval. I had to change everything to minutes. SimCAD should allow for time intervals to be set for each shift. Finally, there is a typo in the 'Generic' process 'General' tab. Default is spelled 'Defaut.'

In conclusion, my evaluation is that SimCAD still needs a lot of work and restructuring to become good simulation software. The VSM capabilities need to be improved so that drawing is easier and the data collection more resembles that of a true VSM. Resources need to get reworked so that the correct number is distributed among a cell and that they do not get over-allocated. The debugging process needs to be more specific and more helpful. Batching objects, using carriers, and data tracking therein significantly needs to be improved. Kanban triggers need to be changed so that they actually work without the need for event extension coding. There should be a central hub for editing and creating event extensions. The start process needs to allow for changeovers and scrap, and the manual needs to be improved with step by step examples that show where the user has to go and what they have to do. Those are the main issues that need to be fixed. Right now SimCAD is not a very intuitive program because there are too many little intricacies that cause a lot of frustration and a lot of lost time from the user. If the recommendations I mentioned above are implemented, SimCAD will be a much more intuitive and improved program which any user can pick up.