



Theses and Dissertations

2005-01-19

Computer-Simulation-Assisted Lean Manufacturing Training

Luoding Wang

Brigham Young University - Provo

Follow this and additional works at: <https://scholarsarchive.byu.edu/etd>



Part of the [Computer Sciences Commons](#), and the [Construction Engineering and Management Commons](#)

BYU ScholarsArchive Citation

Wang, Luoding, "Computer-Simulation-Assisted Lean Manufacturing Training" (2005). *Theses and Dissertations*. 339.

<https://scholarsarchive.byu.edu/etd/339>

This Thesis is brought to you for free and open access by BYU ScholarsArchive. It has been accepted for inclusion in Theses and Dissertations by an authorized administrator of BYU ScholarsArchive. For more information, please contact scholarsarchive@byu.edu, ellen_amatangelo@byu.edu.

COMPUTER-SIMULATION-ASSISTED LEAN
MANUFACTURING TRAINING

by

Luoding Paul Wang

A thesis submitted to the faculty of

Brigham Young University

in partial fulfillment of the requirements for the degree of

Master of Science

School of Technology

Brigham Young University

April 2005

Copyright © 2005 Luoding Paul Wang

All Rights Reserved

BRIGHAM YOUNG UNIVERSITY

GRADUATE COMMITTEE APPROVAL

of a thesis submitted by

Luoding Paul Wang

This thesis has been read by each member of the following graduate committee and by majority vote has been found to be satisfactory.

Date

Charles R. Harrell, Chair

Date

Thomas L. Erikson

Date

Val D. Hawks

RIGHAM YOUNG UNIVERSITY

As chair of the candidate's graduate committee, I have read the thesis of Luoding Paul Wang in its final form and have found that (1) its format, citations, and bibliographical style are consistent and acceptable and fulfill university and department style requirements; (2) its illustrative materials including figures, tables, and charts are in place; and (3) the final manuscript is satisfactory to the graduate committee and is ready for submission to the university library.

Date

Charles R. Harrell
Chair, Graduate Committee

Accepted for the Department

Kevin L. Burr
Graduate Coordinator

Accepted for the College

Douglas M. Charbries
Dean, Ira A. Fulton College of Engineering
and Technology

ABSTRACT

COMPUTER-SIMULATION-ASSISTED LEAN MANUFACTURING TRAINING

Luoding Paul Wang

School of Technology

Master of Science

This thesis assesses the potential of using computer simulation to aid existing lean manufacturing training methods such as lecture and live simulation. An investigation of this possibility was carried out in conjunction with UMEP's Lean 101 class. In the study, two experimental computer simulation models demonstrating the push and pull production scenarios were constructed using ProModel software. Simulation models were equipped with a Visual Basic interface to aid trainees to manipulate the model via ActiveX.

Constructed computer simulation was compared with live simulation to answer these research questions:

1. Was computer simulation able to teach additional lean concepts not covered in live simulations?

2. Was training time less for trainees going through a computer simulation than for those going through a live simulation?
3. Was a computer simulation quicker and easier to set up than a live simulation for trainers?
4. Did computer simulation achieve comparable educational objectives as live simulation?

Objective measurements for first three questions were positive and conclusive. For the fourth one, a survey was conducted among trainees of a treatment group (computer simulation only) and a control group (live simulation only) to collect responses.

Statistical analysis of the subjective responses indicated the computer simulation aided the trainees to learn and implement lean manufacturing, but was not as effective as live simulation.

Holistically, these results did not warrant the complete changeover from live simulation to computer simulation. Yet, a combined implementation of computer simulation and live simulation was proposed to reap the benefits from the best of both approaches.

ACKNOWLEDGMENTS

Thank you to Dr. Charles Harrell and other committee members for their patient and kind help.

Thank you to my wife, family and friends for their love and support.

Also this thesis received kind support from staff and administrators of Brigham Young University, PROMODEL Corporation and UMEP.

Thank you to Ted Naone for originating the concept behind this project.

TABLE OF CONTENTS

List of Figures	x
List of Tables	xi
CHAPTER ONE: INTRODUCTION	1
Thesis Statement.....	2
Justification.....	3
Limitations.....	4
Definition of Terms.....	5
CHAPTER TWO: LITERATURE REVIEW	7
Introduction.....	7
Lean Manufacturing.....	7
Computer –aided Simulation Training.....	9
Lean Manufacturing Training.....	12
Lean Manufacturing Training with Live Simulation.....	13
ProModel & ActiveX.....	14
Computer Simulations Used in Lean Manufacturing.....	15
Existing Computer Simulation Lean Teaching Application.....	16
CHAPTER THREE: METHDOLOGY	17
Introduction.....	17
Lean 101 Class with Live Simulation.....	18
The Computer Simulation.....	21

Survey Method and Form.....	30
CHAPTER FOUR: RESULTS AND ANALYSIS	33
Introduction.....	33
Answer to First Three Quantitative Questions.....	33
Answer to Fourth Qualitative Question.....	34
CHAPTER FIVE: CONCLUSIONS AND RECOMMENDATIONS	39
Conclusions.....	39
Implementation Recommendations.....	40
Recommendations for Additional Features.....	41
Recommendations for Further Study.....	42
 BIBLIOGRAPHY	 43
APPENDIX I: Excerpts from Lean 101 Manual on Live Simulation.....	45
APPENDIX II: Visual Basic Code for Setup Window.....	49
APPENDIX III: ProModel Code for Unpopulated Push Model Shell.....	69
APPENDIX IV: ProModel Code for Unpopulated Push Model Shell.....	83
APPENDIX VI: Lean Simulation Survey.....	97
APPENDIX V: T-test Statistics Analysis for Survey Results.....	99
Approval page.....	iii
Abstract	v
Acknowledgements	vii

List of Figures

Figure 1 Manufacturing System Tab Window	23
Figure 2 Floor Layout Tab Window.....	25
Figure 3 A Sample ProModel Model Layout	26
Figure 4 Samples of Report Window	28
Figure 5 Density Plot	37
Figure 6 Average Score Dot Plot	37

List of Tables

Table 1 Survey Results and Grouped Averages for Comparison Group35

Table 2 Survey Results and Grouped Averages for Treatment Group35

Chapter 1

Introduction

Lean manufacturing has been widely accepted as a proven method to improve productivity and reduce cycle time (Loyd 2000). These profitable improvements have generated a strong demand for training programs. Lean manufacturing training programs commonly integrate live simulations as part of the training (Verma 2003). This integrated teaching method not only covers the principles of lean manufacturing, but also introduces specific tools and techniques to lean novices. With the help of live simulation, the trainees are able to quickly deploy lean manufacturing practices in their operations to improve productivity and reduce cycle times.

One goal in current introductory lean manufacturing educational programs is to convey the lean concepts from trainer to trainees as quickly and completely as possible. As lean emphasizes continuous improvement, the lean manufacturing education community is constantly seeking ways to make lean manufacturing training itself leaner – “a way to do more and more with less and less” (Womack 1996).

Inspired by this continuous improvement mindset of lean manufacturing, trainers and trainees are constantly seeking opportunities to improve these well-established lean manufacturing training programs. For teachers and trainees of one typical and popular lean manufacturing training program – Utah Manufacturing Extension Partnership (UMEP)’s Lean101 class, these main areas for improvement were identified:

1. The live simulation sessions are rather long – a total of 80 minutes for four sessions, which represents the different stages of implementing lean manufacturing concepts. To reduce overall time, some trainers have attempted condensing four live simulation sessions into three. While saving time, this reduces the trainees' ability to learn the importance of sequencing in the implementation of different lean measurements through small increments of reiterations. Also, by limiting the number of iterations, it is difficult for trainers to demonstrate and trainees to learn the practice of continuous improvement in lean manufacturing.
2. Hardware needed for the live simulations causes inconvenience for trainers due to the large container size for the kits. The kits are time consuming and tedious to setup, breakdown and transport.
3. A few important lean elements, such as TPM, Poke-Yoke, & TQM are not presented in live simulations.

One well established approach to reduce the time and cost for training is using computer simulation (Rouse 1992). Though lacking hands-on experience, discrete-event process simulation software can offer these advantages in training:

- ease of use
- realistic animation, and
- powerful reporting capability

Thesis Statement

This thesis tests the postulation that a computerized simulation could be a beneficial substitute for live simulation in the lean manufacturing training. The investigation was

conducted to answer the following research questions regarding computer simulation in comparison with live simulation:

1. Was computer simulation able to teach additional lean concepts not covered in live simulations?
2. Was training time less for trainees going through a computer simulation than for those going through a live simulation?
3. Was a computer simulation quicker and easier to setting up than a live simulation for trainers?
4. Did computer simulation achieve comparable educational objectives as live simulation?

Justification

Incorporating computer simulation in a lean manufacturing training program was anticipated to have these benefits as mentioned in the thesis statement:

1. Reduced setup time and minimize the need to transport hardware for teachers.
2. Reduced time spent in the lean manufacturing training sessions for students.
3. Included additional lean manufacturing elements not well demonstrated in existing live simulation: TPM, Poke-Yoke, and TQM.

Other anticipated educational benefits associated with using computer simulation were:

1. Eliminated some of the unfair comparisons of throughputs in existing live simulations – between systems, one system which was initially warmed-up (i.e. in a steady state), and one that was not.

2. Offered trainee a bird-eye view to observe the computer simulation game as live simulation trainees were confined to their specified work stations.
3. Potential to be distributed and/or remote accessed as a reviewing tool of lean concepts and techniques.
4. Potential to be customized into a production environment the trainee is familiar with to ease the lean acceptance and implementation.

These improvements will further enhance existing training programs and accelerate spreading the knowledge of lean manufacturing. Additionally, the acceptance and implementation of lean will improve manufacturing productivity. If this innovation is implemented on a national scale, the productivity gain for the entire manufacturing industry will be tremendous when typical lean manufacturing results are achieved by all manufacturers.

Limitations

Limited by the pool of resources, only undergraduate manufacturing engineering technology students enrolled at Brigham Young University participated in the survey portion of this investigation. Due to the subjective nature of the survey results, it is recommended that the answer for the fourth research question be revalidated if the application is used with a different audience. In case of using this application for the professional work force trainees, additional testing should be conducted among professional workers to ensure applicability.

Definition of Terms

Computer-simulation: The act of mimicking or replicating an actual system on a computer and then being able to draw inferences that would apply to the actual system.

Discrete-event-process computer simulation: It is an analytical technique utilizing complex simulation models to understand the behavior of a system or process. It has been primarily provided a cost effective means of analysis for resolving problems in tasks such as facility scheduling, capital justification, communication of ideas, and bottleneck identification.

Kaizen: Continuous, incremental improvement of an activity to create more value with less muda (Japanese term for “waste”). It is also called point kaizen and process kaizen.

Kanban: A small card attached to boxes of parts that regulates pull in the Toyota Production System by signaling upstream production and delivery.

Live-simulation: The act of mimicking or replicating an actual system in a hand-on game setting and then being able to draw inference that would apply to the actual system.

Poke-Yoke: A mistake-proofing device or procedure to prevent a defect during order-taking or manufacturing. An ordering taking example is a screen for ordering input developed from traditional ordering patterns and questions those ordering falling outside the pattern. The suspect orders are then examined, often leading to discovery of inputting errors or buying based on misinformation. A manufacturing example is a set of photo cells in parts container along an assembly line to prevent components from progressing to the next stage with missing parts. The poke-yoke in this case is designed to stop the movement of the component to the next station if the light beam has not been broken by

the operator's hand in each bin containing a part of the product under assembly at the moment. A poke-yoke is sometimes also called baka-yoke.

Pull: A system of cascading production and delivery instructions from downstream to upstream activities in which nothing is produced by the upstream supplier until the downstream customer signals a need. The opposite of push. See also Kanban.

Push: Traditional production system that upstream dictates the production rate.

Simulator: A program which provides the capability of simulation, that is data driven but lacks programming abilities.

TPM: Acronym for Total Production Maintenance. A series of methods, originally pioneered by Nippondenso (a member of Toyota group), to ensure that every machine in a production process is always able to perform its required tasks so that production is never interrupted.

TQM: Acronym for Total Quality Management. It introduces statistical tools for manufacturing decision making based on facts.

Training: A planned and organized program for developing skills in the area of manufacturing system design and management.

Value: A capability provided to a customer at the right time at an appropriate price, as defined in each case by the customer.

WIP: Acronym for Work In Process or Inventory. It is a key measurement for a production system.

Chapter Two

Literature Review

Introduction

To prepare for the research and to avoid repeating any known effort, a literature review was conducted to cover these topics: lean manufacturing, computer-aided simulation training, lean manufacturing training, lean manufacturing training with live simulation, ProModel & ActiveX, computer simulation used in lean manufacturing, and existing computer simulation lean manufacturing teaching applications.

Lean Manufacturing

NIST-MEP Lean Network defined Lean Manufacturing as “a systematic approach to identifying and eliminating waste (non-value-added activities) through continuous improvement by flowing the product at the pull of the customer in pursuit of perfection” (www.mep.nist.gov/hottopics1/lean/). It is also known as flow, or synchronous flow manufacturing, one piece or piece part flow manufacturing, just-in-time manufacturing, demand flow technology, world class manufacturing, or Toyota Production System.

The key to lean manufacturing is to compress the cycle time between receipt of an order and receipt of the payment. Compressing cycle time yields greater productivity, shorter delivery times, lower costs, improved quality, and increased customer satisfaction.

The origins of lean manufacturing can be traced back to the early 1900s in the US.

Henry Ford introduced a new manufacturing system - mass production. Ford's philosophy

was to build a small, strong and simple car at the lowest cost. The key elements of the Ford manufacturing system were conveyors, division of labor, and an integrated supply chain (Imai 1986).

The Toyota production system evolved from the Ford manufacturing system. Managers and employees learned to question the need for every work sequence, every piece of in-process inventory, and every second that people, material and machines are idle. As a result, not only did production increase, but quality also increased when people learned to identify and eliminate waste (Ohno 1988; Monden 1993).

Lean manufacturing evolved from the Toyota production system. Lean manufacturing promotes a way of thinking; a culture where all employees continuously look for ways to improve the process with the philosophy of eliminating all non-value added activities.

Companies, such as Boeing and Parker Hannifin, which embraced lean manufacturing, have documented considerable financial gain. Numerous surveys of these companies have reported major market-share gains and sales growth after the implementation of lean manufacturing. The typical productivity gains are (Shewmaker Center LeanMFG Webpage):

- 60% reduction in cycle times
- 98% on-time deliveries
- 80% reduction in floor space
- 40% WIP reductions
- 50% improvement in quality
- 95% machine availability
- 80% reduction in changeovers

- <1% scrap/rework

Computer-aided Simulation Training

The National Science Foundation has declared that information technology is instrumental in training and education, with its ability to (National Science Foundation 1996):

- provide access to world-wide resources;
- facilitate the accumulation, generation, and presentation of data;
- provide tools for analysis and modeling of more or deeper and more realistic examples in a short time;
- enable enquiry and extend the human capability to visualize, organize, and analyze data;
- provide immediate feedback to the student, either from the technology itself or the facilitator/instructor.

Also, National Science Foundation characterized the effective use of IT in education applications as:

- stimulate students and engage them with the material, such as role playing simulations;
- illustrate the workings of complex systems by exploring cause-and-effect relationships, or demonstrate microscopic, molecular, or hypocritical scenarios;
- encourage collaboration with other individuals, teams, or institutions to coordinate a group effort while exposing students to different ideas and perspectives;

- foster development of critical skills, visualization, conceptualization, integration of disparate data, and resolution of patterns within data;

Other research also pointed out, collectively, that IT tools (Eskicioglu 2003):

- enable experimentation with complex, real-life problems through modeling and simulation;
- create interactive environments to receive immediate feedback;
- facilitate collection and presentation of data;
- provide access to world-wide information sources;
- allow self-paced learning;
- support the development of interpersonal communication skills;
- encourage collaboration among students and instructors.

Computer simulation or modeling, among these IT tools, is an effective training aid.

It is more cost effective and time efficient than other traditional training methods such as live simulation, lab/prototype, or on-the-job training (Topham 1997). These are some illustrative examples:

- The earliest documented computer simulation application in training is the flight simulator to train pilots. The simulator not only minimized life-threatening crashing but also reduced flight operating cost (US Senate Report 1984).
- Hospitals and medical schools have started using computer chip imbedded patient simulators to provide training for medical personal due to the health and life risks involved. The Human Patient Simulator – a computer-model-driven, full sized mannequin – delivers experience in true-to-life scenarios that swiftly change to meet instructor's goals. This product offers trainees hand-on patient care

experience without concern over the medical liability associated with live patients (Egan 2004).

- A case of using virtual amphibians to replace the need of dissecting of frogs in high-school biology class. Students paid more attention to the topic of subject matter as the “gross” factor is removed from the lab (Robertson 1994).

The wide-spread use of simulation-aided training is due to the fact that simulation enables trainees to make real-time and interactive decisions rather than the static batch-decision making in the typical lecture setting. After extensive field training with these applications, educators have compiled generalized guidelines to help the simulation-training maker create more effective software and hardware. Most simulation training applications consist of some type of interface which allows trainees to input responses and then generates animated feedback on its display unit.

Previous research work on computer simulation training related to manufacturing applications summarized these advantages:

- Gives the trainee training similar to hands-on or practical training. This is accomplished through the fidelity of the simulation. It is the fidelity that allows trainees to monitor, control, and make realistic decisions about the process in real time (Leon 1993).
- “Promotes critical thinking, develops problem solving skills, enhances creativity, and becomes a cost effective alternative to expensive equipment”. Computer simulation maintains trainees’ interest and offers flexibility in setting learning pace. Also computer simulation offers the flexibility of the location of training –

as long there is a computer terminal available, there is a place of learning and training (Topham 1997).

- Reduces cost and time when justified economically. For large training sessions that performed frequently, the initial investment can be justified (Maul 1993).

On the other hand, if not executed correctly, the training program not only has the danger of not being economically justified, it also can face these possible drawbacks:

- Difficult to achieve reality by the computer simulation. ProModel's realistic animation of the simulation addressed this issue in our case.
- Ease of operation may be absent – additional training might be required. This barrier was overcome with a Visual Basic ActiveX interface in the project.

Lean Manufacturing Training

At the time of publication, a number of lean training options are available for both industry and educational institutions (Verma 2003). For industry, these are typical options.

- Consultants who conduct lean manufacturing training programs in traditional classroom lecture setting to teach lean manufacturing concepts which are applied to case studies or to actual shop process improvement efforts. Sometimes, the consultants incorporate live simulation exercises into their training programs or use computer simulation to demonstrate some of the lean manufacturing elements and principles.
- Various workforce training course CD's that are available through different distribution channels.

- Videos, books, and manuals which can be purchased and are often used to teach lean manufacturing concepts. Some consultants use computer applications to teach and support process improvement efforts.

For educational institutions,

- Many institutions have the training programs which are closely tied to MEP – almost 65% of educational institutions offer different versions of Lean 101 classes.
- Traditional lecture setting by the instructors preparing their own teaching materials and case studies.

Lean Manufacturing Training with Live Simulation

As of October 2003, there were 17 different simulation games available to help teach lean manufacturing (Verma 2003). With the exception of one case of web-based simulation, the other 16 are in the format of live simulation games. The ones associated with MEP are:

- TimeWise simulation by MSI (MEP-MSI),
- 5S simulation (MEP-NIST),
- Setup reduction simulation (MEP-NIST),
- Circuit board simulation (Buzz Electronics' red devil and blue avenger) by NIST-MEP

These MEP programs and other live simulations generally have these strengths (Verma 2003):

- Successfully demonstrate that lean manufacturing techniques can be used in a variety of manufacturing environments.

- Adequately address lean implementation in production area.

Meantime, these common weaknesses are identified:

- only represent the high-volume, low-product-variation type manufacturing
- usually don't address peripheral lean issues such as production scheduling, repair processes, supply management, and production design

NIST-MEP's Lean 101[®] class is the most common and popular training program or classes. It incorporates a live-simulation exercise to train participants in the principles of lean manufacturing. During the training program, trainees participate in three or four rounds of twenty-minute live simulations in addition to the lectures (Lloyd 2000).

Depending on the preference of the trainer, either TimeWise or circuit board assembly, simulation is used in conjunction with the standard lean 101 lecture materials. These simulation exercises simulate the manufacturing of clocks or circuit boards. For UMEP's Lean 101 class program, circuit board assembly was used. This live simulation session will be explained in further detail in Chapter 3.

ProModel[®] & VB ActiveX:

Dr. Charles Harrell founded ProModel Corporation[™] in 1988 with the flagship product ProModel[®]. This discrete-event process simulation software package gained its popularity in a variety of industries (e.g. manufacturing, service and medical) due to its user-friendly model building menu, flexible animation options, and powerful statistical output reporting capabilities. The adoption of the ActiveX component since Visual Basic (VB) release 3 made the software application even more versatile with this added external control capability. This addition extended the functionality of the simulation software application from mere model building, modification and replication to complex

model manipulation, iteration, and even optimization. In particular, the ActiveX component enabled novice end-users to reach into a model and alter the parameters without ever learning the building menu portion of the application. This technology made this thesis project possible as the computer-process-simulation is able to (Law 1991; Harrell 2002):

1. Compress the time frame – taking advantage of the inherent time scaling feature of computer simulation.
2. Provide animated visual feedback, which makes the experience interesting and memorable.
3. Provide accurate statistical output – eliminating any error in result collection.
4. Provide a user-friendly environment – without the need for learning the simulation software itself.
5. Eliminate the physical setup time in live simulation sessions – this is avoided as the computer application loading time/efforts are minimal or none if software is distributed over Internet.

Computer Simulation's Used in Lean Manufacturing

Computer simulation applications supported lean manufacturing in these ways (Czarnecki 2001):

- Current state assessment – to assist documenting along with value stream mapping. The performance metrics derived from simulation can be used to identify and prioritize problems within the manufacturing process.

- Use a training tool by showing the current and future states of the simulations to involved lean manufacturing members. To help them understand the difference and achieve buy-in.
- Evaluate possible future states with simulation to show the impact resulting from various improvement approaches.
- Document opportunities for improvement with the simulation results.
- Measure impact of improvements – simulation allows lean manufacturing team to evaluate their proposed changes by seeing the impact on the computer. This lets the team make incremental (or drastic) changes and observe the effects of implementing proposed changes while having the benefit of without disrupting the manufacturing process and before possible costly investment.

Existing Computer Simulation Lean Training Application

As of the time this thesis, there is no interactive computer simulation game associated with lean training lecture found in printed media or that has been published in the public domain such as Internet (World Wide Web). Only a few Internet websites mentioned various discrete-event simulation software applications in a lecture setting to demonstrate and teach lean manufacturing. However, they are merely used by lecturers or consultants to play out different manufacturing scenarios in front of their classes. Computer simulation so far has been limited in exhibiting the difference between different manufacturing scenarios rather than used as an interactive training tool for individual trainees.

Chapter Three

Methodology

Introduction

Due to its availability and popularity, UMEP's Lean 101[®] class with live simulation was selected as a benchmark or reference. The computer models emulating the live simulation's production scenarios were constructed using ProModel[®] discrete-event process simulation software. To assist the trainee in manipulating the computer simulation models, a Visual Basic interface was also coded.

This computer simulation program was used to answer the four research questions outlined in the thesis statement. Through features comparison and actual measurements between computer and live simulations, the first three questions can be quantitatively assessed and established objectively. The fourth question – whether computer simulation accomplishes the educational objectives in comparison to the existing live simulation – was assessed using a survey. The survey asked for subjective responses from participants as indicators of the effectiveness of the simulation.

Positive evaluation results in all areas will establish the possibility of integrating the computer simulation method into existing lean training programs. In the case of partial fulfillment, alternative resolution(s) will be identified.

Lean 101[®] Class with Live Simulation

UMEP is a branch of the National Institute of Standards and Technology (NIST – a non-regulatory agency under US Commerce Department). The UMEP website (www.mep.org) claims the organization is “a not-for-profit manufacturing consulting group that leverages resources to deliver world-class solutions.” Its “mission is to raise the level of competitiveness and profitability of Utah's manufacturers.” Among its industrial training curriculum, one class is called Lean 101[®], which was developed and licensed by NIST.

The Lean 101[®] class intends to help trainees improve productivity in their own manufacturing operations through the adoption of lean manufacturing principles. UMEP reported that involved organizations had significant productivity gain through the post-training implementations (www.umep.org, 2004).

The training program consists of both lectures and live simulations. Four live simulation sessions representing different operation modes are followed by sessions of lectures.

MEP-NIST recommends 15 to 25 trainees participate in this eight hour long training program. After being presented the agenda of the training program, trainees are given an orientation on the live simulation and were given employee role without any lecture on lean manufacturing concepts (Appendix I). Trainees are to assume different functions of employees in a mock circuit board assembly job shop – Buzz Electronics – as assembly operators, shipping clerk, scheduler, etc. Even one is assigned as an “industrial engineer” who records key performance data such as product lead-time and station cycle times.

Two product lines with slightly different routings go through this job-shop with eight

workstations complemented with a sales office, a scheduling office, a warehouse and a shipping area. The arrangement of the job shop work stations represents a traditional functional layout. The job shop is operated with a traditional batch manufacturing system – the production is push by the scheduling or forecasting. The “workers” are instructed to make as many finished products as possible. This first round of hands-on simulation of twenty minutes typically yields no or little shipment but large WIP, poor quality, discouraged “workers”, and a financial statement in the red.

The operation result is then reviewed by the instructor with the trainees. Following, the instructor teaches lean manufacturing tools and principles in the lecture format. Then the trainees are supposed to apply what they just learned in a second round of the live simulation game. Then, more lean concepts are presented before a third round of live simulation, as well as for the fourth round.

Under the instructor’s guidance, the trainees are expected to apply newly learned lean knowledge and principles from the lectures progressively in these four rounds of simulation games. The teaching method is considered as “an excellent approach to grasping the basics of lean and attacking underlying cultural issues involved in a typical lean transformation” (Loyd 2000).

Throughout the lecture and simulations, trainees are steered to identify the advantages of lean manufacturing as well apply newly acquired lean manufacturing knowledge and techniques. The trainee will participate in implementing lean manufacturing tools, such as layout improvement, raw material point of use, waste elimination, changeover reduction, and visual work instruction. The trainees will also notice the positive change

in the performance matrix such as WIP reduction, on-time delivery increase, and profit improvements.

In the simulation, the frustration experienced by the trainees in the first round helps them to realize that the pull production system improved productivity while individual works station output rate remained at about the same level. This aspect appeals to the floor worker trainees as their working intensity is not affected when involved in lean manufacturing. Often, the initial frustration leads to a deeper appreciation and understanding of the lean manufacturing principles.

Though the effectiveness of the Lean 101[®] class was well established (Stier 2003), the live simulation method has some of the following flaws based on trainee feedbacks and personal observations (Appendix VI):

1. Lacks consistency and repeatability on key elements. The individual's process cycle time (assembly speed) usually decreases after first session of the simulation game due to learning curve. The reduction of cycle time could skew the output results for comparison purposes.
2. Constrains the time frame and thereby prevents possible experiment with the sequencing of different lean manufacturing steps. To get meaningful results, every live simulation session is set to be 20 minutes. The time constraint not only limits trainees from the possibility of fine-tuning operating parameters, but also prevents them from learning the importance of sequencing the lean implementation steps. For example, reducing batch quantity before accomplishing setup reductions will only have an adverse effect on the productivity.

3. Biases the comparison between warmed-up systems with those that are not. The first round was not warmed-up as the system has no work in process (WIP) to begin with. For successive sessions, the facilitator or instructor adds semi-finished parts to the workstations as WIP to warm up the system. This discrepancy distorts the comparison of the throughputs from different rounds.
4. Confines one's view. Limited by designated manual assembly functions, individual trainees are confined to a small portion of the workshop in the live simulation – lacking the overall view, and not able to observe and understand the improvements in other areas.
5. Overlooks some lean elements. TQM (Total Quality Management), TPM (Total Production Maintenance), Poke-Yoke, were not included in the live simulation to demonstrate their critical roles in improving productivity.
6. Has a high physical set-up cost. In order to reuse the parts built from previous rounds of simulation, in addition to a lecturer, a facilitator is required to disassemble the components at the location. On average, a total of 80 minutes were spent setting up the simulation workshop and disassembly of the parts for the subsequent rounds. Despite this effort to reuse the components, large quantities of simulation components are required for the live simulation game – several cases-loads of parts need to be transported to and from training sites.

The Computer Simulation

Intending to mitigate the shortcomings of live simulation, a computer simulated workshop was re-created with ProModel[®] software. The computer models were coded to mimic the live simulation production systems. To minimize the time needed for trainees

to learn this simulation software and related model, an interface, or control panel, was programmed with Microsoft's Visual Basic (VB). An ActiveX control component connects between VB and ProModel[®] – it populates and manipulates the computer models, starts the simulation, and reports the results through the user interface.

The computer simulation represents an IC chip packaging operation for two different products: metal (hermetic) and plastic (injection molding). Though the naming of work stations and product in computer model is different from the ones in Lean 101[©] live simulation, the mechanics of the two are the same. Both represent a two-product-line workshop which shares the majority of work-stations. These are the IC shop work-stations in computer simulation:

1. die attach
2. wire-bond
3. trimming
4. molding (only used for plastic IC products)
5. inspection
6. rework

Since the computer simulation model replicates the live simulation model, as long as the process time of the live simulation game are plugged into the computer simulation model as processing time, the outcome of computer simulation matches with the live simulation outcome.

The computer simulation program consists of two major programming components:

1. VB interface includes these functional windows: Welcome, Setup, Help, and Report windows (Appendix II).

- Two pre-built ProModel® simulation shells. One model shell used for push system. The other one is for pull system (Appendix III & IV).

The VB interface setup window allows the trainees to set up their virtual shop. In the Manufacturing System tab, trainees are presented with these setting options:

Lean Ed - Interchange

Step One: Setup an IC packaging operation base on following options:

Manufacturing System | Floor Layout

Production Type:

☒ push: Batch-size 6 Plastic 4 Metal ☐ pull, one-piece flow

	kitting	die-attach	wire-bond	trimming	molding	inspection	re-work
Number of operators	1	1	1	1	1	1	1
Process times expressed in normal distribution (mean, standard deviation):							
Plastic	2, 1	2, 1	2, 1	2, 1	2, 1	2, 1	2, 1
Metal	2, 1	2, 1	2, 1	6, 2	0	2, 1	2, 1
Input queue capacity	inf	inf	inf	inf	inf	inf	inf
Output queue capacity	inf	inf	inf	inf	inf	inf	inf

Misc. option: ☐ TPM ☐ Poka-Yoke ☐ Cross Training ☐ TQM ☐ Eliminate Kitting Station ☐ Eliminate Scheduling Office ☐ Takt Time Balance

Benefit	Cost
Reduce Std. Dev. by 2/3	\$50
Reduce mean operation time 50%	\$30
Increase productivity by 15%	\$35
Decrease re-work rate from 10% to 1%	\$45
Saves \$50	\$0
Save \$50	\$0
Balance through-put rate between stations	\$30

StepTwo: Pick following action options

promodel

Figure 1 Manufacturing System Tab Window

- Production mode selection between pull and push – selects corresponding ProModel shell.
- If in push mode – push.mod file is selected, and the model uses batch quantity or queue capacity between the work-stations. User can change the batch quantity for both products in batch quantity drop down windows.

3. If in pull mode – pull.mod file is selected, and the model uses Kanban quantities between the work-stations. User determines the Kanban size in the Kanban quantity drop down windows.
4. Queue capacity / kanban quantity is linked to .mod file's queue location macro.
5. Station capacity is linked to either .mod file's location construct's capacity field.
6. TPM uses a macro controls process standard deviation. The reduction of standard deviation time results represents production overall-equipment-effectiveness improvement due to the implementation of TPM. This element is not present in live simulation.
7. TQM option activates the rework rate macro in .mod file. The rework rate is reduced from 10% to 1% when the TQM option selected. This element is not present in live simulation.
8. Poke-Yoke option controls work stations' mean process rate macro in the .mod file. This element is not present in live simulation.
9. Cross-training option uses a VB built-in algorithm to factor in the cost and gain within the total profit.
10. Takt-time balance option manipulates a process macro in the .mod file to alter its processing characteristics.
11. Eliminate kitting and/or scheduling option uses icon index change to hide the location icon(s) and to attach dummy locations to other location in the .mod file.

The VB code also links changes of certain settings (macros) with other setting selections. This automated safeguard feature prevents conflicting choices and ensures the constructed simulation model will always be meaningful. Example: when pull mode is

selected, Cross Training, Eliminate Kitting & Schedule option boxes are selected and grayed out as these options are in default as part of a pull production system.

Trainees can also setup a desired layout of the production system by clicking and dragging the location icons in the Floor Layout tab. The changed icon XY axis coordinates are used to update location and path network XY table for the computer models.

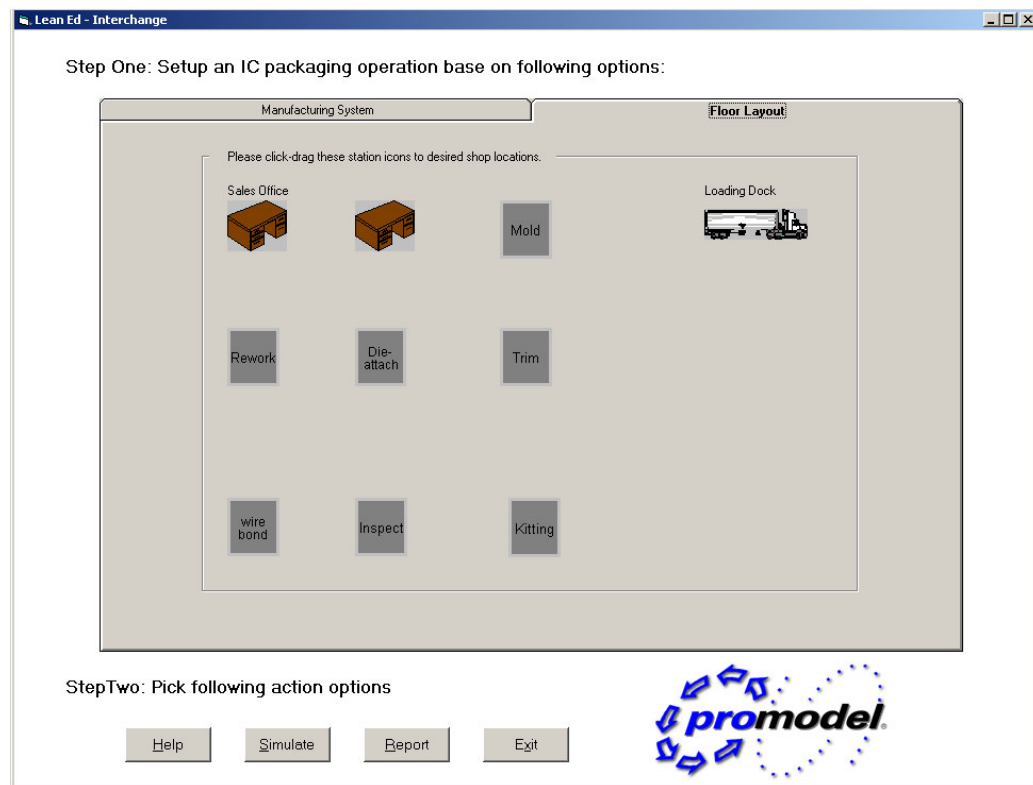


Figure 2 Floor Layout Tab Window

Four operation option buttons are available at the bottom of the setup window:

- Help: In case of trainee need additional information, he or she can click this button to displays stored information in help window.

- **Simulate:** Clicking this button launches VB ActiveX to load a populated ProModel® production model designed by trainee and starts simulation automatically.
- **Report:** The button uses VB ActiveX to derive the simulation results from simulation RDB output file, then to format and to display main operation results from the computer simulation. A default output file is installed along with the program to prevent crashing while trying to open an empty output file before any simulation has been performed on the computer.
- **Exit:** It closes the entire application. A confirmation window will appear to ask for confirmation to avoid unintended exit.

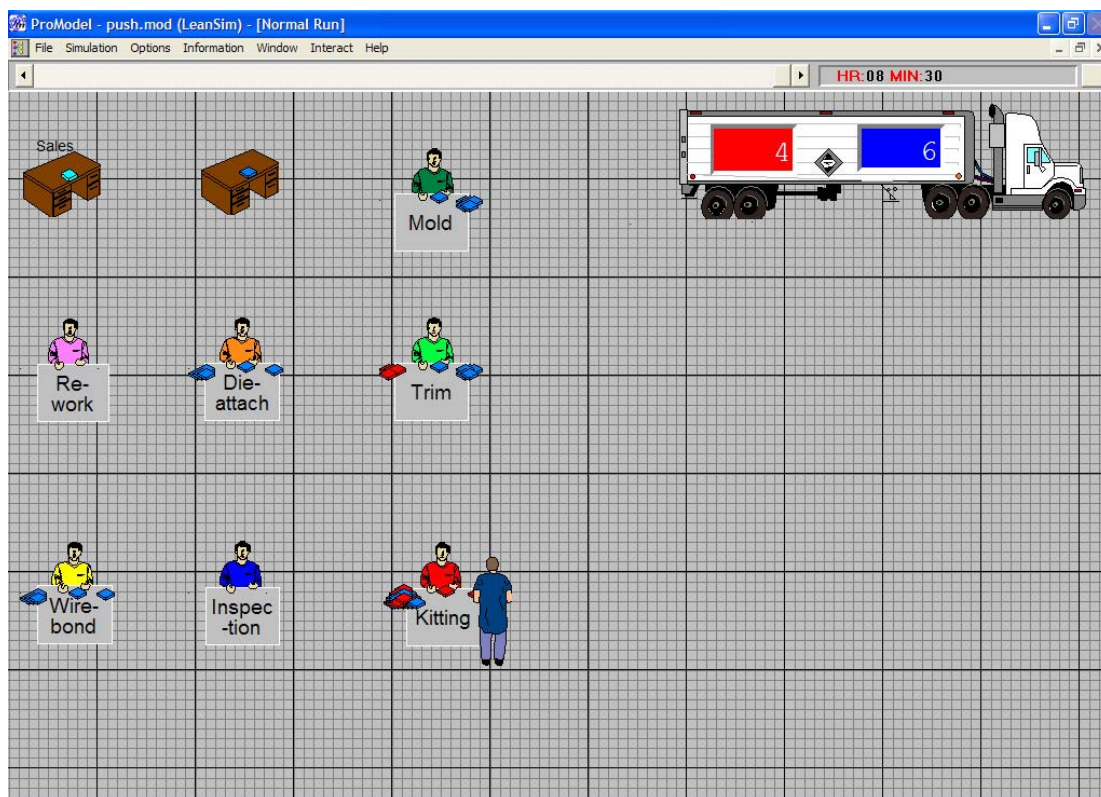


Figure 3 A Sample ProModel Model Layout

The trainee can run the simulation at the desired animation speed by adjusting the speed bar in the ProModel[®] application. The simulations are programmed to run for 8 hours after a 1-hour warm-up period. Since the simulation software compresses the process time, it can take as little as 30 seconds to finish a simulated 9-hr time span. The application closes itself at the end of the simulation. A message window appears and asks trainees whether want to view the results report or return to the setup window.

Clicking the Report button brings up the results window with these key results from previous simulation run:

- the quantity completed for each product
- WIP (Work-In-Process)
- average cycle time
- on-time rate
- overall profit

These numbers are derived from variable in the ProModel[®] RDB file or calculated in the VB code. In the report window, the trainee has the option to print the results or close the report window.

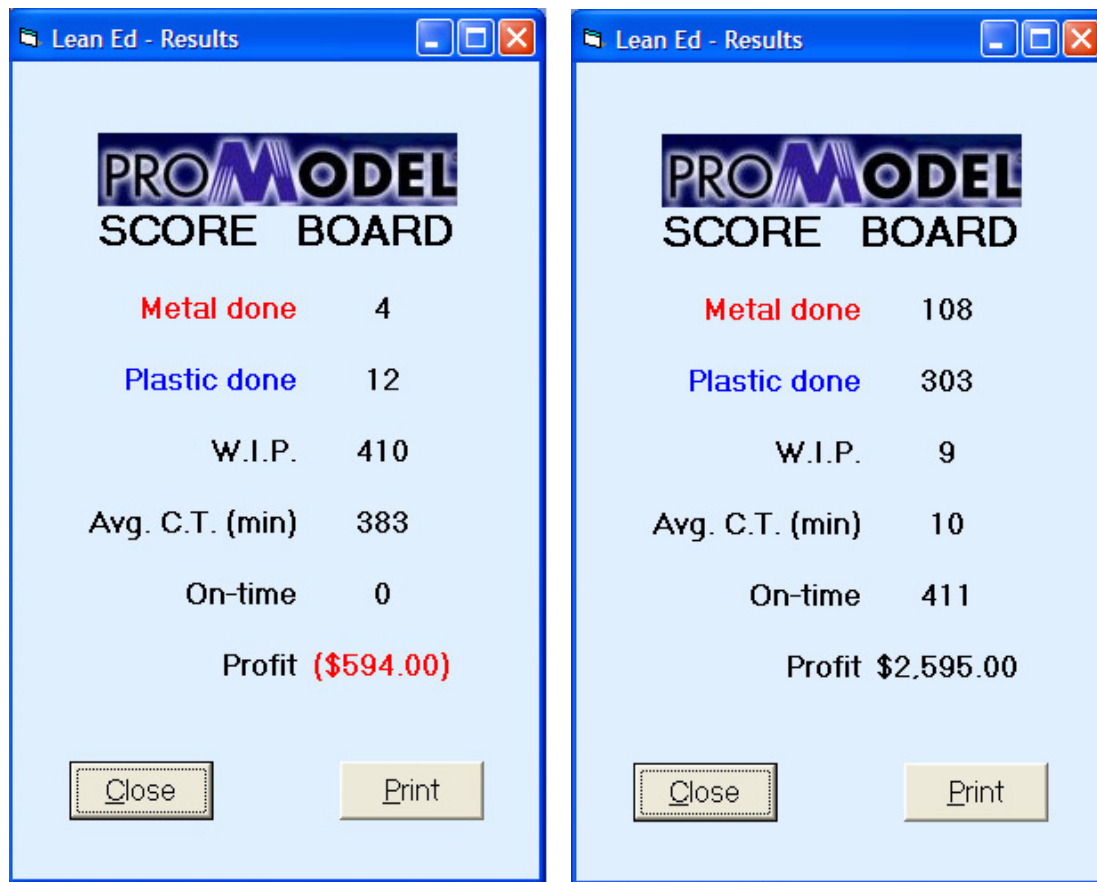


Figure 4 Samples of Report Window

Through the iteration of the computer simulation with different setups, the trainees will learn results from various lean manufacturing implementations. The “profit” value relates to different Kaizan measurements such as, TPM and Poke-Yoke, and will be calculated and displayed in the report window. These quickly derived profit values relating to different manufacturing scenarios provide trainees valuable insight into the sequencing or combining of different lean manufacturing implementations.

The investigation found that, in comparison to the live simulation, the computer simulation had these advantages:

1. Trainers need not spend more time on set-up and breakdown the physical hardware as in live simulation.

2. The time constraint is alleviated by the compressed time frame of computer simulation. This gives trainees time to run simulation iterations with small incremental changes. Trainees have the opportunity to fine-tune the operations parameters to optimize the results. Also, trainees are able to experiment and realize the importance of sequencing the implementation.
3. The computer simulation includes these important lean manufacturing components: TPM, Poke-Yoke, and TQM. Also, these action items are assigned price tags to make trainees conscious of the cost-benefit justification of these lean measures.
4. The model's individual work station process characteristics are consistent throughout the simulation as the model used standardized process distributions. This avoids the decreasing of the process time of individual work station throughout the live simulation as the trainees usually pick up speed on the manual assembly in later sessions of operations due to the learning curve.
5. The models have built-in warm-up periods to ensure the fairness of comparison. The computer model warms up the system by running the simulation for 1 hour without collecting the statistics for all the runs. Then it starts to collect simulation statistics data for the next 8 hours.
6. Since the trainee has control over the entire production system, he/she has a bird's eye view of the process and system as opposed to the limited impact and visibility of live simulations.
7. Since the computer simulation requires adequate computer terminals to be available, and software can be transferred via the Internet or CDs, only one

instructor is now required for both lecturing and preparation for the simulation sessions. Reasonable setup time is about 2 minutes for each new terminal to download the self-installing programs. And there is no needed to spend time for the terminals that have installed with the application. Since PC computer terminals are popular now, the availability in any sizable institution is common. Material requirements and setup/disassembly time could be reduced. This also removes the limitation on the class size – live simulation limits the number of trainees to be 15 to 25 in order to perform the manual assembly simulation.

Specific functions and features of this constructed computer simulation will be compared with live simulation compare in chapter four to answer the first three research question.

Survey Method and Form

To answer the fourth research question, whether a comparable educational objective was achieved by the computer simulation in comparison to live simulation, a survey study was constructed and administrated to a group of participants attending a Lean 101[©] class taught by a UMEP instructor. For the simulation session, some attending the live simulation as the control group; the rest went to a computer lab to do computer simulations as the treatment group. Each computer simulation attendee was assigned to one computer terminal which had the software loaded prior to the session. The author of this thesis provided instruction to the trainees on how to use the software, and resolved any questions or issues raised by trainees.

After the experimental training course was over, survey forms (Appendix IV) were used to collect responses submitted by all the trainees from both live and computer

simulations. The survey consisted of the thirteen multiple choice questions and two comment style questions. The first multiple choice question identified which simulation the trainee attended – live or computer. Multiple choice questions 2 to 4 attempted to identify the trainee's demographics such as education level, related work experience, and prior exposure to lean manufacturing. These questions intended to establish whether there was a significant difference among the treatment group and the control group.

The rest of the survey questions were developed based on the UMEP Lean 101[®] class standard feedback form. Question 5 asked the trainees their opinions on the lectures and the lecturer. Since the subjects of this question were the same, this question was used to test the opinion base of the survey participants.

Questions 6 to 13 covered different aspects of the live or the computer simulation the trainees attended. These rating questions intended to gauge the overall effectiveness of the simulation base on trainees' subjective responses to the provided statements. These statements were designed to signify these qualities of the simulation:

- Whether the simulation is helpful to understand the principles and the benefits of lean manufacturing? Questions 6, 7, 12, 13.
- Whether the simulation is interesting and engaging? Question 8.
- Whether the procedure for doing the simulation is clear? Questions 9, 10, 11, 12.

Trainees rated these statements with reactions based on their subjective impressions from six gradient choices: 1 - strongly disagree, 2 - disagree, 3 - somewhat disagree, 4 – somewhat agree, 5 – agree, 6 – strongly agree. Where 1 through 3 were negative – unfavorable responses to the statement, 4 through 6 were favorable. Since all the

statements in the survey about the program were positive, 4 indicated a favorable opinion about the program, 5 indicated a more favorable opinion, 6 indicated a most favorable opinion. Average score on the rating of statements is calculated to reflect the overall effectiveness.

At the end of survey form, attendees were prompted to express any additional opinion about their experience by elaborating on “particular likes” and “particular dislikes”. The responses for these two open-ended questions offered additional insights to compare the simulations.

Chapter Four

Results and Analysis

Introduction

This chapter presents the findings in response to each of the research questions listed in Chapter One. The first three research questions are grouped together to answer as they can be mostly obtained through quantitative observation or feature comparison. The results of the fourth research question related to the effectiveness of the computer simulation are presented using survey data and statistical analysis.

Answers to First Three Quantifiable Research Questions

Based on personal observation and discussions with the lean trainer, the following time-savings were realized and simulation enhancements made using computer simulation:

1. 80 minutes of teacher setup time was eliminated (80min → none). And only one instructor was required for the entire training program compared to two for live simulation.
2. Trainees saved 50 minutes in their simulation games (80 minutes → average 30 minutes).
3. Additional lean techniques (TPM, Poke-Yoke, and TQM) were included.

Additionally, the computer simulation resulted in other improvements over live simulation as follows:

1. All scenarios in simulation models were warmed up to provide a fair comparison of the output results between the runs.
2. Trainees had wider scope of observation in the computer simulation.
3. The computer simulation had the potential to be distributed as a review tool on lean concepts.
4. The simulation models offered the possibility to be modified by the instructor on-site to reflect the shop environment the trainees were familiar with in order to minimize possible resistance from trainees.

Also, survey feedback constructed for the fourth question reaffirmed the computer simulation's time compression value: 7 out of 8 computer simulation attendees used words such "fast," "speed," "quick" to describe their experience while one live simulation attendee characterized live simulation as "very long". Another live simulation attendee mentioned the "limited visibility" of the live simulation in the questionnaire. "Lacking of personal interaction" and "lacking of team building experience" were also mentioned as pitfalls of the computer-simulations-only approach.

Answer to Fourth Qualitative Research Question

Twenty-three Brigham Young University manufacturing engineering and technology undergraduate students participated in this survey study which evaluated the effectiveness of the computer simulation in comparison to the traditional live simulation. Of the twenty-three participants, fifteen volunteered to attend the live simulations as control group (to fill the minimum required of slots in the live simulation), and eight volunteered to attend the computer simulations as treatment group. The treatment group spent an average of 30 minutes with the computer model while the control group spent 80 minutes

in total, both as predicted. After the entire training session was over, all participants filled out and returned survey forms. All the score results were transferred from the survey forms to an Excel spreadsheet. The tallies and their averages are summarized in the tables below:

Table 1 Survey Results and Grouped Averages for Control Group

id	q1	q2	q3	q4	q5	q6	q7	q8	q9	Q10	q11	q12	q13	mean
1	control	3	2	2	3	6	5	5	5	5	5	5	5	5.13
2	control	4	2	2	3	5	5	5	5	5	5	5	4	4.88
3	control	3	2	2	4	6	5	6	5	6	5	5	5	5.38
4	control	3	2	2	3	6	6	6	5	6	6	6	5	5.75
5	control	3	1	3	3	6	6	6	6	6	5	6	5	5.75
6	control	3	1	2	4	5	5	5	5	5	5	5	5	5
7	control	3	1	2	4	5	5	5	6	6	5	6	5	5.38
8	control	3	1	3	3	6	6	6	6	6	6	6	6	6
9	control	3	1	2	5	6	6	6	6	6	6	6	6	6
10	control	3	2	2	5	6	6	6	5	5	5	5	5	5.38
11	control	3	1	3	5	5	5	4	4	4	5	5	6	4.75
12	control	3	1	3	4	5	6	6	5	6	5	6	6	5.63
13	control	3	1	3	5	6	5	5	6	5	5	6	6	5.5
14	control	3	1	3	3	4	4	4	3	4	4	4	4	3.88
15	control	4	1	3	4	4	6	6	6	6	6	6	5	5.63
Control Avg.		3.13	1.33	2.47	3.87	5.4	5.4	5.4	5.2	5.4	5.2	5.47	5.2	5.33

Table 2 Survey Results and Grouped Averages for Treatment Group

id	q1	q2	q3	q4	q5	q6	q7	q8	q9	q10	q11	Q12	q13	mean
16	treatment	4	2	2	3	4	3	4	3	3	3	3	6	3.63
17	treatment	4	4	2	3	4	3	4	5	4	4	3	4	3.88
18	treatment	3	1	2	4	5	4	5	4	5	4	5	5	4.63
19	treatment	4	1	2	3	3	4	4	5	3	5	2	4	3.75
20	treatment	4	2	2	3	6	5	5	5	5	5	5	5	5.13
21	treatment	4	1	2	4	5	4	4	5	4	5	5	5	4.63
22	treatment	3	1	2	4	4	4	5	4	3	3	3	3	3.63
23	treatment	4	2	2	3	5	5	6	6	5	5	5	5	5.25
Treatment Avg.		3.75	1.75	2	3.38	4.5	4	4.63	4.63	4	4.25	3.88	4.63	4.31

The demographic results from questions number two to number four indicated that the two groups were similar based on the group averages and their differences. Two average scores for question number two indicated that all participants were either in the college or very close to graduating. The difference between the two groups' average in their education level was .62. For question number three, working experience, the scores from two groups indicated all participants but one had less than 3 years working experience. The difference between two groups' average score was .42. For question number four, prior exposure to lean manufacturing, the difference between two groups' average was .47. The two groups' overall scores indicated they all had some or limited exposure to lean manufacturing. These comparisons indicated that the two groups were comparable despite their slightly differences.

In question 5, trainees rated their lectures and lecturer – the common elements in their lean training program. Despite the fact that two groups attended the same lecture with the same lecturer at the same time, the average from two groups showed some discrepancy. The live simulation control group scored higher than the computer simulation treatment group did (3.87 vs. 3.38). This indicated that the treatment group had a slightly lower opinion base.

Statistical software R[®] (free statistical software) plotted questions 6 through 13 with density plot and averaged score dot plot to help us understand the data we collected:

- The density plot showed a relatively larger spread for the treatment group – visually, the bottom opening of the curve of the treatment group was wider than the control group's. This indicated a wider range of opinions was given by the treatment group than the control group. This difference indicated that the control

group's opinions were more consistent within the group, while the treatment group tended to have more disparity of opinions among themselves.

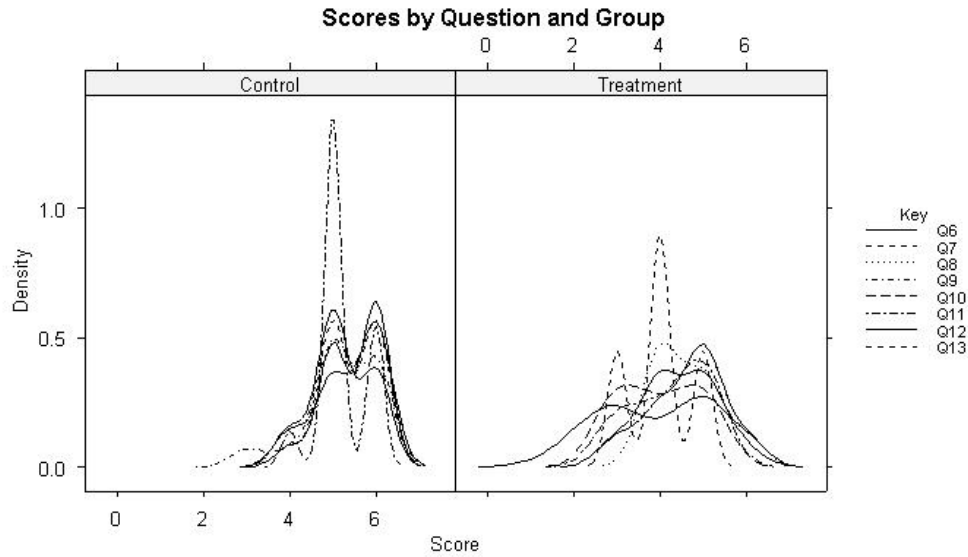


Figure 5 Density Plot

- Average score dot plot showed that the distributions were normal even though the sample sizes were small. Also the calculation indicated the treatment group had a lower average score of 4.31 while 5.33 for the control group.

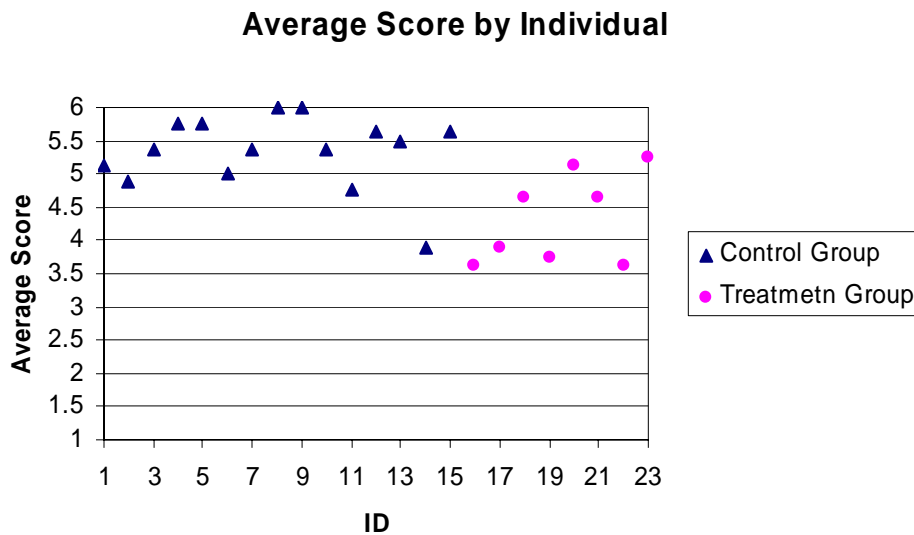


Figure 6 Average Score Dot Plot

Both groups were in the positive feedback range. The treatment group was 4.31 while the control group was 5.33. The sample mean difference was 1.02. The result was then verified with StatCrunch[®] online calculator to find that the 95% confidence interval and 2-Sample t- test. An affirmative 95% confidence level and a low 0.0031 P value indicated the difference between the two sets of results was statistically significant (Appendix V).

The possible cause of this weakness of computer simulation was that some elements of the live simulation were difficult to be duplicated in the computer simulation such as:

- Frustration and chaos in first round's "pre-lean" unprofitable operation even with intensive manual assembly,
- Team work experience,
- Real-time communication.

For the fourth question, the survey results showed that stand-alone computer simulation was not as effective as live simulation even though it scored within the positive feedback range.

Chapter Five

Conclusions and Recommendations

Conclusions

By testing computer simulation in lean manufacturing training, the four research questions proposed in the thesis statement were answered.

For the first three quantitative research questions analyzing computer simulation in comparison to live simulation, the answers were all positive:

1. Computer simulation was able to incorporate lean manufacturing elements not covered in live simulations.
2. The use of computer simulation reduced the overall training time for trainees in live simulations from 80 minutes to 30 minutes.
3. Using computer simulation reduced the physical setup time of live simulations for trainers from 80 minutes to none.

For the fourth research question, whether computer simulation achieved the same educational objectives as live simulation, the results from two groups of trainees with similar demographic background revealed that the stand-alone computer simulation was not as effective as live simulation.

Based on the collective findings for these four research questions, it would be inadvisable to completely change over from live simulation to computer simulation. An alternative, however, can be extrapolated from these results which would reap the

benefits from the best of both approaches. This would be to implement some combination of live and computer simulations in tandem.

Implementation Recommendations

To take the advantages from both types of simulations, trainees can attend lean manufacturing training program with this recommended sequence of events:

1. Attend a brief lecture for introduction and orientation.
2. Attend the first live-mode simulation session to experience the chaos and frustration in the pre-lean manufacturing situation.
3. Return to the lecture where the lecturer reports the results of the live simulation.
4. The group will then be taught about the concepts of lean manufacturing.
5. Instead of going back to a second session of live simulation, the group will use the operations data from the previous live simulation session to recreate the results using the computer interface. The experience of live simulation assists the trainees in grasping the computer simulation model.
6. After validating the computer model with data from the live simulation, the trainees will then use the computers to simulate proposed improvements and implementations just learned from the lecture. Also, the instructor can promote the team effort by forming the teams among the trainees to work together on how to implement the lean.
7. Continue with additional lecture and computer simulation sessions until all important lean topics are covered.

The approach described above still saves 60 minutes for the teacher instead of 80 minutes as in the computer-simulation-only method. It also still saves 40 minutes for the

trainees as opposed to 50 minutes. Though this dual-simulation approach does not reduce the time as much as stand-alone computer simulation does, other intended benefits from computer simulation are retained.

Recommendations for Additional Features

The following feature or functionality changes are also recommended for computer simulation based on the feedback and lessons learned from the trial run:

1. Make the animation more realistic by displaying the operator motion at individual workstations during the simulation. Show the operator switching workstations if the cross-training option is selected.
2. The adding or removing of workstations could be reflected with a counter display or image changes during the simulation to show the physical increment or decrement of the workstation capacity.
3. Though setup reduction was not mentioned in Lean 101[®] live simulation, this lean concept could be included in the computer simulation. This possible addition emphasizes the adverse effect of batch reduction without reducing setup time.
4. The inventory/queue can be visually changed according to the setup choice and the accumulation of WIP.
5. A more structured and standardized instruction should be prepared and presented to ensure that trainees understand the interface and computer modeling more quickly.
6. Add an algorithm in the code to track the movement of the location and assign a cost for the move as the current version cost nothing to relocate work stations.

7. Create a website to allow online access of the interface and models. This avoids the need for ProModel® installation and licensing on individual computers at remote training sites. The web-interface will pass on the data and animation between host server and individual computers. Trainees can even review the lean concepts by revisiting the computer simulation model with the log-in privilege after the class. This also allows trainees to demonstrate the lean concepts and benefits to their peers from any location with web access.
8. Build in an option to allow distributed interactive simulation. A distributed environment would allow multiple persons to work within one simulation game at the same time. The interaction promotes team spirit during the continuous improvement phase of lean journey.

With these further refinements and considerations, the addition of an interactive computer simulation approach has the potential to improve the effectiveness of existing lean manufacturing teaching programs.

Recommendations for Further Study and Research

The above recommendations of implementation should be carried out and verified with additional experiments. Any insight or observation gained from these experiments should be integrated into the successive studies to perfect the lean training program.

Bibliography

Czarnecki, Hank, Nicholas Loyd “Simulation of Lean Assembly Line for High Volume Manufacturing.” <<http://www.atnlean.com/publications/Simulation%20of%20Lean%20Assembly%20Line%20for%20High%20Volume%20Manufacturing.pdf>>, 2002.

Egan, Mary E., “The Perfect Patient”, Forbes, June 21, 2004.

Eskicioglu, Ahmet, Danny Kopec, “The Ideal Multimedia-Enabled Classroom: Perspectives from Psychology, Education, and Information Science”, Journal. of Educational Multimedia and Hypermedia, (2003) 12(2), 199-221.

Harrell, Charles R., Biman K. Ghosh, and Royce Bowden, Simulation Using ProModel, Boston: McGraw-Hill, 2002.

Imai, Masaaki, Kaizen: The Key to Japanese Competitive Success, New York: Random House, 1986.

Law, Averill M., W. David Kelton, Simulation Modeling & Analysis, Boston: McGraw-Hill, 1991.

Leon, Jorge V., Mark R Miller, “Interactive Computer Simulation in the Quality Assurance Laboratory”, Journal of Engineering Technology, (pp. 6-9), Fall 1993.

Loyd, Nicholas, Simulation of Information System In a Lean Factory, Huntsville simulation conference proceeding, 2002.

Maul, Gary P. Spotts, David, S, “A comparison of Computer –Based Training and Classroom Instruction”, Industrial Engineering, pp. 25-27, February 1993.

MEP-MSI, “MEP MSI Time Wise Training”, November 17, 2004
<http://www.mepmsi.org/index.php?page=supplychain_matrixarea1>, 2002.

MEP-NIST, “MEP Source for Manufacturers” November 17, 2004,
<<http://www.mep.nist.gov/lean/lean.html>>, 2002.

Monden, Yashuhiro, Toyota Production System: An Integrated Approach to Just-In-Time, Norcross: Industrial Engineering and Management Press, 1993.

National Science Foundation, “Information technology: Its impact on undergraduate education in science, mathematics, engineering and technology”, NSF (pp. 98-82), 1996.

NIST/MEP, Principles of Lean Manufacturing with Live Simulation, Gaithersburg: National Institute of Standards and Technology Manufacturing Extension Partnership, 1998.

Ohno, Taiichi, Toyota Production System, Portland: Productivity Press, 1988.

Robertson, David, William Johnson, "Virtual Frog Dissection: Interactive 3D Graphics Via the Web" November 8, 2004 <<http://dsd.lbl.gov/vfrog/WWW.94/paper.html>> 1994.

Rouse, William B., “Creating Successful Products, System, and Organizations”, Strategies for Innovation, New York: John Wiley & Sons, Inc., 1992.

Shewmaker Center for Workforce Technologies, “Lean Mfg”, November 17, 2004 <<http://www.nwacc.edu/scwt/MFG/LeanMfg.php>>, 2002.

Stier, Kenneth W, “Teaching Lean Manufacturing Concepts through Project-Based Learning and Simulation”, Journal of Industrial Technology (19.4), 2003.

Topham Greg C., Interactive Simulation-Based Training for Quality Management, Brigham Young University Master Thesis 1997.

UMEP.ORG, November 8, 2004, <www.MEP.org> 2000.

US Senate Committee on Armed Services, Subcommittee on Tactical Warfare.
“Development and use of training simulators : hearings before the Subcommittee on Tactical Warfare of the Committee on Armed Services, United States Senate, Ninety-eighth Congress, second session”, September 26, 27, 1984.

Verma, Alok K., Simulation Tools and Training Programs in Lean Manufacturing – Current Status, Final Report Submitted to NSRP-ASE Program, October 19, 2003.

Womack, James P, Daniel T. Jones, Lean Thinking: Banish Waste and Create Wealth in Your Corporation, New York: Simon & Schuster, 1996.

WWW.LEAN.ORG Jim Womack's E-letters 10/11/02 The “Right Sequence” for Implementing Lean,
<<http://www.lean.org/Community/Registered/ShowEmail.cfm?JimsEmailId=2>> 2000.

Appendix I.

Excerpts from Lean 101 Manual on Live Simulation

Orientation

to

Buzz Electronics Enterprises
(BEE)

Buzz Electronics Enterprises

The Blue Avenger

Basic model
for
every day use

Buzz Electronics Enterprises

The Red Devil

Industrial model
for
commercial use

BEE The Bottom Line

The Blue Avenger

Sells for \$20 Materials cost \$5.00

The Red Devil

Sells for \$30 Materials cost \$7.50

Labor cost: \$7.50 / person / shift
Facilities cost: \$10.00 / table / shift

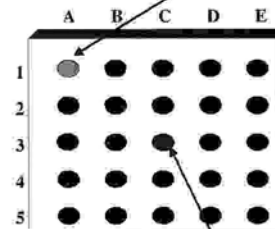
BEE Production Process Orientation

- | | | |
|------------------------------|----------------------------------------------|-------|
| • Sales Representative | Processes "Customer Order" | |
| • Production Scheduler | Generates "Factory Order" from forecast | |
| • Kitter(s) | Organizes raw materials for "Factory Orders" | |
| • Material Handler | Moves product between ALL workstations | |
| • Spring Assembler | Inserts Springs | ■ ■ ■ |
| • Resistor Assembler | Inserts Resistors | ■ ■ ■ |
| • LED Assembler | Inserts LEDs | ■ ■ ■ |
| • Diode Assembler | Inserts Diodes | ■ ■ ■ |
| • Inspector | Conducts functional tests | |
| • Reworker | Repairs failed boards | |
| • Warehouse/Ship Clerk | Matches boards to "Customer Orders" | |
| • Instruction Crib Attendant | Controls work instructions | |
| • Production Supervisor | Supervises production | |
| • Industrial Engineer | Monitors production process | |
| • Trucker | Ships products to the customer | |

Assembly

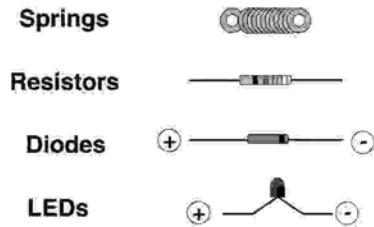
BEE Circuit Board Orientation

Blue, and Red, Boards

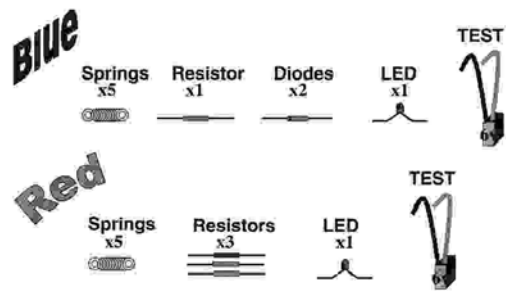


EXAMPLE:
1) Insert spring into 'C3'

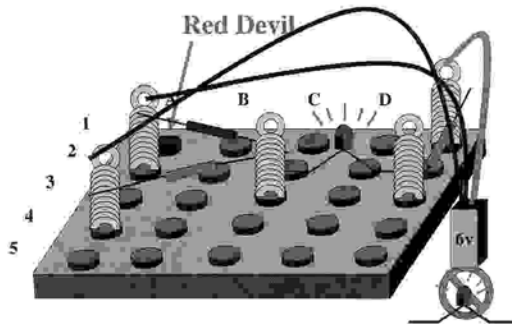
BEE Product Components Orientation



BEE Product Routing



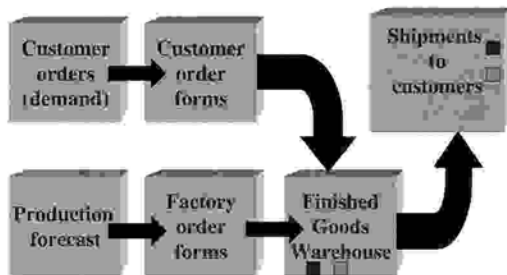
BEE Circuit Board Assembly Example



BEE Production Batching

The Blue Avenger	6 per batch	■ ■ ■ ■ ■ ■
The Red Devil	4 per batch	■ ■ ■ ■

BEE Production Scheduling Process



BEE Customer Service Targets

Promised shipments to customers

Promised shipments to customers	■	4 minutes after order
Promised shipments to customers	■	5 minutes after order

All orders are filled "first-in, first-out" (FIFO)

Buzz Electronics Enterprises

Company Policies

- All shifts are 20 minutes
- Keep busy at all times
- Yell if you need parts
- Handle all parts first in - first out (FIFO)
- Only the material handler can move parts
- Stay at your workstation
- The boss is *always* right!

Buzz Electronics Financial Report

	Round 1	Round 2	Round 3	Round 4
Sales Revenue	\$	\$	\$	\$
Operating Costs	\$	\$	\$	\$
Facility	\$	\$	\$	\$
Labor	\$	\$	\$	\$
Materials	\$	\$	\$	\$
(Total Operating Costs)	\$	\$	\$	\$
Net Income	\$	\$	\$	\$

A Summary of How BUZZ ELECTRONICS Evolved

Changes from Round 1 to Round 2

- 1) Change the layout - better flow between stations
- 2) Softer boss, more sensitive management style
- 3) Instructions for all products are released to workstation
- 4) No storage room for work in process - stored at workstation
- 5) Better work instructions to describe jobs
- 6) Better visual aids with the use of color coded templates

Changes from Round 2 to Round 3

- 1) Raw materials stored at point of use
- 2) Stations arranged to remove the need for a material handler
- 3) No need to complete production reports, except inspectors tally of pass/fail/failed products
- 4) Cut batch size in half
- 5) Increased team work/crosstraining and improve quality

Changes from Round 3 to Round 4

- 1) Put Pull System in place by building to customer order and not to a production forecast
- 2) Use of kanbans to signal production activities
- 3) Calculations and awareness of takt time to have a feel for whether customer orders are filled effectively
- 4) Overall, implementation of cellular manufacturing with multiskilled employees and strong team work
- 5) Eliminate the need for Production Scheduler responsibilities and Factory Order Forms

Appendix II.

Visual Basic Code for Setup Window

```

' In order to make this program work properly,
' please make sure these VB's reference libraries are included:
'   Visual Basic For Applications
'   Visual Basic run time objects and procedures
'   Visual Basic objects and procedures
'   OLE Automation
'   Microsoft ActiveX Data Objects 2.1 Library
'   ProModel Events1.0 Type Library
'   ProModel Type Library
'   PMVBObj
'
' If you have multiple ProModel products (such as MedModel, ServiceModel or eModel)
' intalled on your hard drive, please make sure ProModel is your default program.

' Please include these files on the C:\temp\LeanEd -
'   PullShell.mod; PushShell.mod

```

Option Explicit

```

Dim intSel As Integer      'variable to indicate selection

Dim intI As Integer

Dim intJ As Integer

Dim intK As Integer

Dim intX(21) As Integer    'location coordinantes of the location
Dim intY(21) As Integer

Dim intCost(10) As Integer 'cost accounting variable

Dim sngX(8) As Single      'X, Y coordinance of stations images
Dim sngY(8) As Single      'X, Y coordinance of stations images
Dim xPos As Single         'auxiliary X, Y coordinance for Icon loacations
Dim yPos As Single         'auxiliary X, Y coordinance for Icon loacations
Dim i As Integer           'utility variable

Dim intMean As Single      'process capability mean

```

```

Dim intSpread As Single      'process capability sigma
Dim dblRjctRt As Double     'process rejection rate

Dim strModel As String      'model name
Dim PromodelString As String

Dim ProMod As Object        'Object use to be ProModel.CProModel
Dim ProModDATA As Object    'Object use to be ProModel.CProModelData

Private Sub Form_Load()

    Dim intI As Integer, intJ As Integer, intK As Integer

    For intI = 0 To 6                'fillout the ComboBoxes for Q/Kanban and
        For intJ = 1 To 20 Step 1
            cboBef(intI).AddItem intJ
            cboAft(intI).AddItem intJ
        Next intJ
        For intJ = 1 To 4 Step 1
            cboWk(intI).AddItem intJ
        Next intJ
        cboBef(intI).AddItem "inf"
        cboAft(intI).AddItem "inf"
    Next intI

    For intK = 1 To 25 Step 1        'default batch-size
        cboBatch(0).AddItem intK
        cboBatch(1).AddItem intK
    Next intK

    For intI = 0 To 6                'default: "push" system queues
        intMean = 2
        intSpread = 1
        lblRed(intI).Caption = intMean & ", " & intSpread
        lblBlue(intI).Caption = intMean & ", " & intSpread
        cboBef(intI).Text = "inf"    'location capacity definition
    Next intI

```

```

        cboBef(intI).Enabled = True

        cboWk(intI).Text = "1"

        cboWk(intI).Enabled = True

        cboAft(intI).Text = "inf"

        cboAft(intI).Enabled = True

    Next intI

'individually change un-common process times
lblRed(4).Caption = "0"

lblRed(3).Caption = intMean * 3 & ", " & intSpread * 2


cboBatch(0).Text = 6
cboBatch(0).Enabled = True
cboBatch(1).Text = 4
cboBatch(1).Enabled = True


dblRjctRt = 0.05                                'default rejection rate

End Sub

Private Sub optPull_Click()

    'one cap @ kanban/Q
    'arrival one piece order

        cboBatch(0).Text = 1
        cboBatch(0).Enabled = False
        cboBatch(1).Text = 1
        cboBatch(1).Enabled = False

    For intI = 1 To 6                                'pre-defined Kanban size

        cboBef(intI).Text = 1
        cboBef(intI).Enabled = True
        cboBef(intI).BackColor = &HFF8080
        cboAft(intI).Text = 1
        cboAft(intI).Enabled = True
        cboAft(intI).BackColor = &H8080FF
    
```



```

Next intI

'cboBef(6).Text = 8                                'recommended Kanban size @ final stage
'cboAft(6).Text = 5

lblName3.Caption = "Plastic Kanban capacity"

lblName4.Caption = "Metal Kanban capacity"

'also "remove" kitting, scheduling graphiclly, and "attach" them to Sales
chkKit.Value = 1                                    'default selection
chkKit.Enabled = False
chkSchd.Value = 1
chkSchd.Enabled = False
chkCrstrn.Value = 1
chkCrstrn.Enabled = False

End Sub

Private Sub optPush_Click()

'no constrain at all, but arrival rate of orders equals capacity otherwise never
reach balance

'l@station, inf @other Q's

'arrival is one piece/batch

For intI = 0 To 6                                    'queue capacity definition
    cboBef(intI).Text = "inf"
    cboBef(intI).Enabled = True
    cboBef(intI).BackColor = &HFFFFFF
    cboAft(intI).Text = "inf"
    cboAft(intI).Enabled = True
    cboAft(intI).BackColor = &HFFFFFF
Next intI

    lblName3.Caption = "Input queue capacity"
    lblName4.Caption = "Output queue capacity"

cboBatch(0).Text = 6                                're-enble options "grayed" by optPull
cboBatch(0).Enabled = True

```

```

cboBatch(1).Text = 4
cboBatch(1).Enabled = True
chkKit.Value = 0
chkKit.Enabled = True
chkSchd.Value = 0
chkSchd.Enabled = True
chkCrSTrn.Value = 0
chkCrSTrn.Enabled = True

End Sub

Private Sub chkKit_Click()
    If chkKit.Value = 1 Then                                'if kitting is eliminated
        imgIcon(2).Visible = False
        lblKit.ForeColor = &H808080
        lblBlue(0).Caption = "0"
        lblRed(0).Caption = "0"
        cboBef(0).Enabled = False
        cboWk(0).Enabled = False
        cboWk(0).Text = "Inf"
        cboAft(0).Enabled = False
        imgIcon(2).Left = 777
        imgIcon(2).Top = 777
    Else
        imgIcon(2).Visible = True
        lblKit.ForeColor = &H404040
        lblBlue(0).Enabled = True
        lblRed(0).Enabled = True
        cboBef(0).Enabled = True
        cboWk(0).Enabled = True
        cboWk(0).Text = "1"
        cboAft(0).Enabled = True
        lblRed(0).Caption = intMean & ", " & intSpread
        lblBlue(0).Caption = intMean & ", " & intSpread
    'set kitting graphic to blank, "attach" it to the die-attach station, no overlapping
    'change process/wait time to zero, gray out the Kitting station on the form

```

```

End If

End Sub

Private Sub chkSchd_Click()

    If chkSchd.Value = 1 Then          'if schudling office is eliminated
        imgIcon(1).Visible = False    'set schedule graphic to blank,
        imgIcon(1).Left = 747        'attach location to sales, no overlapping
        imgIcon(1).Top = 747

    Else
        imgIcon(1).Visible = True
    End If

End Sub

Private Sub chkPkyk_Click()

    If chkPkyk.Value = 1 Then
        intCost(1) = 30
        intMean = 1
    Else
        intCost(1) = 0
        intMean = 2
    End If

    For intI = 0 To 6
        lblRed(intI).Caption = intMean & ", " & intSpread
        lblBlue(intI).Caption = intMean & ", " & intSpread
    Next intI

    lblRed(4).Caption = "0"

    If chkTakt.Value = 0 Then
        lblRed(3).Caption = intMean * 3 & ", " & intSpread * 2
    End If

    If chkKit.Value = 1 Then

```

```

        lblRed(0).Caption = "0"

        lblBlue(0).Caption = "0"
    End If
End Sub

Private Sub chkTakt_Click()
    If chkTakt.Value = 0 Then
        lblRed(3).Caption = intMean * 3 & ", " & intSpread * 2
        intCost(2) = 30
    Else
        lblRed(3).Caption = intMean * 1 & ", " & intSpread * 1
        intCost(2) = 0
    End If
End Sub

Private Sub chkTpm_Click()
    If chkTpm.Value = 1 Then
        intCost(3) = 50
        intSpread = 0.2
    Else
        intCost(3) = 0
        intSpread = 1
    End If

    For intI = 0 To 6
        lblRed(intI).Caption = intMean & ", " & intSpread
        lblBlue(intI).Caption = intMean & ", " & intSpread
    Next intI

    lblRed(4).Caption = "0"

    If chkTakt.Value = 0 Then
        lblRed(3).Caption = intMean * 3 & ", " & intSpread * 2
    End If

```

```

    If chkKit.Value = 1 Then
        lblRed(0).Caption = "0"
        lblBlue(0).Caption = "0"
    End If
End Sub

Private Sub chkTqm_Click()
    If chkTqm.Value = 0 Then
        dblRjctRt = 0.1
        intCost(4) = 45
    Else: dblRjctRt = 0.01
        intCost(4) = 0
    End If
End Sub

Private Sub cmdBack_Click(Index As Integer)
    intSel = MsgBox(" Are you sure want to quit? ", vbOKCancel + vbQuestion +
vbDefaultButton2, "Lean Ed")
    If intSel = 1 Then
        End
    End If
End Sub

Private Sub cmdHlp_Click()
    frmHelp.Show
End Sub

Private Sub frmLayout_DragDrop(Image As Control, X As Single, Y As Single)
    Image.Move X - Image.Width / 2, Y - Image.Height / 2
    i = 0
    While i <> Image.Index
        i = i + 1
    Wend

```

```

If TypeOf Image Is VB.Image Then
    sngX(i) = X
    sngY(i) = Y
End If

End Sub

Private Sub cmdSmlt_Click(Index As Integer)
    Dim X As Integer
    Dim RecordIndex As Long
    Dim result(10) As Long
    Dim Xaxis As Integer
    Dim Yaxis As Integer
    Dim dblX As Double
    Dim dblY As Double
    Dim GraphicID As Integer
    'Dim strDummy() As String
    Dim strDummyCap(21) As String
    Dim strLoc As String
    Dim strCap As String
    Dim vStatus As Long
    Dim strTotalCost As String
    Dim strRealText As String

    'check for batch size whether larger than queue size while doing pull; batch size is
    not zero;

    For intI = 0 To 6
        If ((cboBef(intI).Text <> "inf") Or (cboAft(intI).Text <> "inf")) And
        ((Val((cboBatch(0).Text) > Val(cboBef(intI).Text)) Or (Val(cboBatch(1).Text) >
        Val(cboAft(intI).Text)) Or (cboBatch(0) = "0") Or (cboBatch(1) = "0"))) Then

            intSel = MsgBox("Error Batchsize!", vbOKOnly + vbExclamation, "Warning")

            Exit Sub

        End If
    Next intI

    intMean = 2
    intSpread = 0.5

```

```

'assign the cost for all the setting change
If chkTpm.Value = 1 Then
    intCost(0) = 50
End If
If chkPkyk.Value = 1 Then
    intCost(1) = 30 & intSpread = 0.2
End If
If chkCrstrn.Value = 1 Then
    intCost(2) = -20                                'can I?
End If
If chkTqm.Value = 1 Then
    intCost(4) = 45 & dblRjctRt = 0.01
End If
If chkTakt.Value = 1 Then
    intCost(7) = 30
End If

'Initiate ActiveX steps
Set ProMod = CreateObject("ProModel")
Set ProModDATA = CreateObject("ProModelData")

Select Case optPush.Value
    Case True: strModel = "push"
    Case False: strModel = "pull"
End Select

'capture layout initial location
For intI = 2 To 8
    sngX(intI) = imgIcon(intI).Left
    sngY(intI) = imgIcon(intI).Top
Next intI

ProMod.LoadModel ("C:\Temp\LeanEd\" & strModel & ".mod") 'base on mfg sys
selection, pick different template model

```

```

'set the location capacity, and XY coordiance

'make a size of 21 array to capture the stations and queue capacity, name, etc. to
"re-create" layout

For intI = 0 To 18 Step 3

    'strDummy(intI) = "Q_station" & (intI \ 3 + 1)

    strDummyCap(intI) = cboBef(intI \ 3).Text

    intX(intI) = CInt(sngX(intI \ 3 + 2) / Screen.TwipsPerPixelX) - 15 'location: X,
Y axis of (-15, 0 for befQ locaitons)

    intY(intI) = CInt(sngY(intI \ 3 + 2) / Screen.TwipsPerPixelY) + 25

Next intI

For intJ = 1 To 19 Step 3

    'strDummy(intJ) = "Station" & (intJ \ 3 + 1)

    strDummyCap(intJ) = cboWk(intJ \ 3).Text

    intX(intJ) = CInt(sngX(intJ \ 3 + 2) / Screen.TwipsPerPixelX)

    intY(intJ) = CInt(sngY(intJ \ 3 + 2) / Screen.TwipsPerPixelY)

Next intJ

For intK = 2 To 20 Step 3

    'strDummy(intK) = "Station_Q" & (intK \ 3 + 1)

    strDummyCap(intK) = cboAft(intK \ 3).Text

    intX(intK) = CInt(sngX(intK \ 3 + 2) / Screen.TwipsPerPixelX) + 40

    intY(intK) = CInt(sngY(intK \ 3 + 2) / Screen.TwipsPerPixelY) + 25

Next intK

ProModDATA.Populate          'initiate populate

'fix schedule office locationXY / icon if needed
Xaxis = CInt((imgIcon(1).Left) / Screen.TwipsPerPixelX)          'conversion
Yaxis = CInt((imgIcon(1).Top) / Screen.TwipsPerPixelX)

If chkSchd.Value = 1 Then

    Call FixLocIcon(26, 29)      'index schedule location icon to the 29th in glb

End If

Call FixLocXY(26, Xaxis, Yaxis)

Call FixPathNetworksNode(2, Xaxis, Yaxis)

```



```

'fix kitting locationXY / icon if needed

If chkKit.Value = 1 Then          'if kitting is removed
    Call FixLocIcon(2, 29)        'kitting location icon become 29th icon in glb --
blank
End If

RecordIndex = 0                  'build locations table: offices and 7 station
For intI = 0 To 20               'start the cycle for workstation locationXY placing
    RecordIndex = intI + 1        'off-set by 1
    strCap = strDummyCap(intI)    'Capacity
    GraphicID = RecordIndex       'location icon
    Xaxis = intX(intI)            'location X,Y coordinates
    Yaxis = intY(intI)
    Call FixLocXY(RecordIndex, Xaxis, Yaxis)
    Call FixPathNetworksNode(RecordIndex + 2, Xaxis, Yaxis)
    Call FixVariables(RecordIndex, strCap)
Next intI

'batch size variables
Call FixVariables(26, cboBatch(0).Text)
Call FixVariables(27, cboBatch(1).Text)

'build macro/variable table for process-time,
For intI = 0 To 6
    If lblBlue(intI).Caption = "0" Then
        strRealText = "0"
    Else
        strRealText = "N(" & lblBlue(intI).Caption & ")"
    End If
    Call FixMacro(intI + 1, strRealText)
Next intI

'anomaly -- for that red station
strRealText = "N(" & lblRed(3).Caption & ")"
Call FixMacro(8, strRealText)

```

```
strTotalCost = CStr(intCost(0) + intCost(1) + intCost(2) + intCost(3) + intCost(4) +
intCost(5) + intCost(6) + intCost(7) + intCost(8) + intCost(9))
```

```
Call FixMacro(9, strTotalCost)
```

```
'fix processing probability variables in Processing_Routing_Probability subtable for
both products
```

```
ProModDATA.SelectMainRecordByIndex 19, 1
```

```
ProModDATA.SelectMainRecordByIndex 20, 1
```

```
ProModDATA.SetRealFieldValue 20, 13, dblRjctRt
```

```
ProModDATA.SelectMainRecordByIndex 20, 2
```

```
ProModDATA.SetRealFieldValue 20, 13, 1 - dblRjctRt
```

```
ProModDATA.SelectMainRecordByIndex 19, 2
```

```
ProModDATA.SelectMainRecordByIndex 20, 1
```

```
ProModDATA.SetRealFieldValue 20, 13, dblRjctRt
```

```
ProModDATA.SelectMainRecordByIndex 20, 2
```

```
ProModDATA.SetRealFieldValue 20, 13, 1 - dblRjctRt
```

```
ProMod.RedrawLayout
```

```
ProMod.RedrawTables
```

```
ProMod.ShowTranslationDlg True
```

```
ProMod.Simulate
```

```
'start simulation
```

```
'ProMod.SetMessageMode 0
```

```
Do
```

```
DoEvents
```

```
'wait for end of simulation
```

```
vStatus = ProMod.GetStatus
```

```
Loop Until vStatus = 8
```

```
'8 is the "end" value
```

```
SendKeys "(Tab)"
box,
```

```
'clear end message prompt
```

```
SendKeys "(Enter)"
```

```
Do
```

```
' "quit-loop"
```

```
DoEvents
```

```
ProMod.Quit
```

```
Loop Until X = 0
```

```

Set ProModDATA = Nothing                                'clear/reset data object
Set ProMod = Nothing                                    'clear/reset model object

intSel = MsgBox("Would you like to see the results?", vbYesNo, "Simulation Complete")
'ending msg

If intSel = 6 Then
    cmdRpt_click (1)
End If

End Sub

Private Sub cmdRpt_click(Index As Integer)

    Dim iReplication As Integer
    Dim iScenario As Integer
    Dim iPeriod As Integer
    Dim iField As Integer
    Dim iRecord As Integer
    Dim iTable As Integer
    Dim iNumFields As Integer
    Dim iNumRecords As Integer

    Dim RDBObj As Object                                'RDB Object (a ProModel OLE
Function)

    Dim sName As String, sTableWanted As String
    Dim sngOne As Single
    Dim sngTwo As Single
    Dim sngThree As Single
    Dim sngFour As Single
    Dim sngFive As Single
    Dim sngSix As Single

    Dim X As Integer
    Dim varX As Variant

    'MousePointer.Value = 11

    frmReport.Show                                     'bring up the report window

```

```

'Extract data, table and path we are using is pre-defined here
sTableWanted = "Variables"

'Open the RDB Data Server to allow access to the stats module
Set RDBObj = CreateObject("RDBDataServer")
X = RDBObj.OpenFile("C:\program files\ProModel\output\" & strModel & ".rdb")

'Set the initial values to look up in the stats module
iScenario = 1
iReplication = 1
iPeriod = 1
iTable = 1
iField = 1
iRecord = 1

'Finds the right table
For iTable = 1 To 20
    X = RDBObj.SelectData(iScenario, iReplication, iPeriod, iTable, iField, iRecord)
    sName = RDBObj.TableName
    If InStr(1, sName, sTableWanted, 1) > 0 Then Exit For
Next iTable

iNumFields = 5

X = RDBObj.SelectData(iScenario, iReplication, iPeriod, iTable, iNumFields, 1)
sngOne = RDBObj.Getvalue
X = RDBObj.SelectData(iScenario, iReplication, iPeriod, iTable, iNumFields, 2)
sngTwo = RDBObj.Getvalue
X = RDBObj.SelectData(iScenario, iReplication, iPeriod, iTable, iNumFields, 3)
sngThree = RDBObj.Getvalue
X = RDBObj.SelectData(iScenario, iReplication, iPeriod, iTable, iNumFields, 4)
sngFour = RDBObj.Getvalue
X = RDBObj.SelectData(iScenario, iReplication, iPeriod, iTable, iNumFields, 5)
sngFive = RDBObj.Getvalue
X = RDBObj.SelectData(iScenario, iReplication, iPeriod, iTable, iNumFields, 6)

```

```

sngSix = RDBObj.Getvalue

'after extract data such as cost/revenue, the number be crunched to find the net-gain
frmReport.lbl1.Caption = FormatNumber(sngOne, 0)
frmReport.lbl2.Caption = FormatNumber(sngTwo, 0)
frmReport.lbl3.Caption = FormatNumber(sngThree, 0)
frmReport.lbl4.Caption = FormatNumber(sngFour, 0)
frmReport.lbl5.Caption = FormatNumber(sngFive, 0)

If sngSix < 0 Then
    frmReport.lbl6.ForeColor = &HFF&
Else
    frmReport.lbl6.ForeColor = &H0&
End If

frmReport.lbl6.Caption = FormatCurrency(sngSix)

X = RDBObj.CloseFile
Set RDBObj = Nothing

End Sub

Function FixLocXY(RecIndex As Long, X As Integer, Y As Integer)

    ProModDATA.SelectMainRecordByIndex 1, RecIndex
    ProModDATA.SelectMainRecordByIndex 45, 1
    ProModDATA.SetIntFieldValue 45, 10, X           'icon location X
    ProModDATA.SetIntFieldValue 45, 11, Y           'icon location Y

End Function

Function FixLocIcon(RecIndex As Long, Graphic As Integer)

    ProModDATA.SelectMainRecordByIndex 1, RecIndex
    ProModDATA.SelectMainRecordByIndex 45, 1

```

```

ProModDATA.SetIntFieldValue 45, 7, Graphic      'Icon Graphic Lib. ID

End Function

Function FixPathNetworksNode(RecIndex As Long, intX As Integer, intY As Integer)

    Dim lngTableIndex As Long
    Dim lngSubTableIndex As Long
    Dim lngFieldIndex1 As Long
    Dim lngFieldIndex2 As Long

    lngTableIndex = 3          'Path Networks table #
    lngSubTableIndex = 54      'Nodes sub-table #
    lngFieldIndex1 = 3         'Node X axis field #
    lngFieldIndex2 = 4         'Node Y axis field #

    ProModDATA.SelectMainRecordByIndex lngTableIndex, 1
    ProModDATA.SelectMainRecordByIndex lngSubTableIndex, RecIndex
    ProModDATA.SetIntFieldValue lngSubTableIndex, lngFieldIndex1, intX + 30
'node X
    ProModDATA.SetIntFieldValue lngSubTableIndex, lngFieldIndex2, intY + 30
'node Y

End Function

Function FixVariables(RecIndex As Long, strInitValue As String)

    Dim lngTableIndex As Long
    Dim lngFieldIndex As Long

    lngTableIndex = 9          'Variable table
    lngFieldIndex = 4          'Initial value field

    ProModDATA.SelectMainRecordByIndex lngTableIndex, RecIndex
    ProModDATA.SetStringFieldValue lngTableIndex, lngFieldIndex, strInitValue

End Function

```

```

Function FixMacro(RecIndex As Long, strText As String)

    Dim lngTableIndex As Long
    Dim lngFieldIndex As Long

    lngTableIndex = 11          'Macro table
    lngFieldIndex = 2           'Text field

    ProModDATA.SelectMainRecordByIndex lngTableIndex, RecIndex
    ProModDATA.SetStringFieldValue lngTableIndex, lngFieldIndex, strText

End Function

```


Appendix III

ProModel Code for Unpopulated Push Model Shell

```

*****
*
*
*           Formatted Listing of Model:
*
*           J:\Thesis\Project\push.MOD
*
*****

```

```

Time Units:           Minutes
Distance Units:       Feet
Initialization Logic:  animate 25
Termination Logic:    profit = blue_done*8+red_done*5-
(S1_cap+S2_cap+S3_cap+S4_cap+S5_cap+S6_cap)*50 - ExtCost - WIP*1
                      Cycle_time = Total_CT / (Blue_done + Red_done)

```

```

*****
*
*           Locations
*
*****

```

Name	Cap	Units	Stats	Rules	Cost
QS0	QS0_cap	1	Time Series Oldest	,	
#					
#kitting					
S0	S0_cap	1	Time Series Oldest	,	
SQ0	SQ0_cap	1	Time Series Oldest	,	
QS1	QS1_cap	1	Time Series Oldest	,	
#					
#die - attach					
S1	S1_cap	1	Time Series Oldest	,	
SQ1	SQ1_cap	1	Time Series Oldest	,	
QS2	QS2_cap	1	Time Series Oldest	,	
#					
#wire - bond					
Q2	S2_cap	1	Time Series Oldest	,	

```

SQ2      SQ2_cap 1      Time Series Oldest, ,
QS3      QS3_cap 1      Time Series Oldest, ,
#
#trimming
S3        S3_cap 1      Time Series Oldest, ,
SQ3      SQ3_cap 1      Time Series Oldest, ,
QS4      QS4_cap 1      Time Series Oldest, ,
#
#molding
S4        S4_cap 1      Time Series Oldest, ,
SQ4      SQ4_cap 1      Time Series Oldest, ,
QS5      QS5_cap 1      Time Series Oldest, ,
#
#inspection
S5        S5_cap 1      Time Series Oldest, ,
SQ5      SQ5_cap 1      Time Series Oldest, ,
QS6      QS6_cap 1      Time Series Oldest, ,
#
#rework
S6        S6_cap 1      Time Series Oldest, ,
SQ6      SQ6_cap 1      Time Series Oldest, ,
dummy1   QS7_cap 1      Time Series Oldest, ,
#
#warehouse
S7        S7_cap 1      Time Series Oldest, ,
dummy2   SQ7_cap 1      Time Series Oldest, ,
#
#sales
O1        inf      1      Time Series Oldest, ,
#
#scheduling
O2        inf      1      Time Series Oldest, ,
#
#delivery
O3        inf      1      Time Series Oldest, ,

```

```
*****
*                               Entities                               *
*****
```

Name	Speed (fpm)	Stats	Cost
Blue_order	1500	Time Series	
Red_order	1500	Time Series	
Blue	150	Time Series	
Red	150	Time Series	

```
*****
*                               Path Networks                           *
*****
```

Name	Type	T/S	From	To	BI	Dist/Time	Speed Factor
Net	Passing	Speed & Distance	N1	N2	Bi	7.07	1
			N2	N3	Bi	8.48	1
			N3	N4	Bi	3.00	1
			N4	N5	Bi	3.00	1
			N5	N6	Bi	4.00	1
			N6	N7	Bi	1.00	1
			N7	N8	Bi	2.00	1
			N8	N9	Bi	3.00	1
			N9	N10	Bi	2.00	1
			N10	N11	Bi	2.00	1
			N11	N12	Bi	2.82	1
			N12	N13	Bi	3.00	1
			N13	N14	Bi	3.00	1
			N14	N15	Bi	2.23	1
			N14	N18	Bi	9.05	1

N15	N16	Bi	2.23	1
N16	N17	Bi	2.82	1
N17	N18	Bi	3.16	1
N18	N19	Bi	2.00	1
N19	N20	Bi	4.00	1
N20	N21	Bi	2.00	1
N20	N24	Bi	12.00	1
N21	N22	Bi	2.23	1
N22	N23	Bi	4.00	1
N23	N24	Bi	4.12	1
N24	N25	Bi	8.00	1
N25	N26	Bi	2.00	1
N26	N27	Bi	8.24	1
N27	N1	Bi	62.03	1

```
*****
*                                     *
*                               Interfaces                               *
*                                     *
*****
```

Net	Node	Location
-----	-----	-----
Net	N1	O1
	N2	O2
	N3	QS0
	N4	S0
	N5	SQ0
	N6	QS1
	N7	S1
	N8	SQ1
	N9	QS2
	N10	Q2
	N11	SQ2
	N12	QS3
	N13	S3

N14	SQ3
N15	QS4
N16	S4
N17	SQ4
N18	QS5
N19	S5
N20	SQ5
N21	QS6
N22	S6
N23	SQ6
N24	dummy1
N25	S7
N26	dummy2
N27	O3

```
*****
*                                     *
*                               Mapping                               *
*                                     *
*****
```

Net	From	To	Dest
-----	-----	-----	-----
Net	N2	N1	
	N3	N2	
	N4	N3	
	N5	N4	
	N6	N5	
	N7	N6	
	N8	N7	
	N9	N8	
	N10	N11	
	N11	N12	
	N12	N13	
	N13	N14	
	N14	N18	

N15	N16
N16	N17
N17	N18
N18	N19
N19	N20
N20	N24
N21	N22
N22	N23
N23	N24
N24	N25
N25	N26
N26	N27
N27	N1
N1	N2
N10	N9
N2	N3
N11	N10
N3	N4
N12	N11
N13	N12
N14	N13
N4	N5
N15	N14
N16	N15
N18	N14
N5	N6
N19	N18
N20	N19
N21	N20
N24	N20
N25	N24
N26	N25
N27	N26
N6	N7
N7	N8

N8	N9
N1	N27
N9	N10
N14	N15
N17	N16
N18	N17
N20	N21
N22	N21
N23	N22
N24	N23

* Resources *

Name	Units	Stats	Res Search	Ent Search	Path	Motion	Cost
Operator 1		Summary	Closest	Oldest	Net	Empty: 7 fpm	
					Home: N1	Full: 5 fpm	
						Accel: 5 fpss	
						Decel: 5 fpss	
						Pickup: 60 Seconds	
						Deposit: 60 Seconds	

* Processing *

Process			Routing			
Entity Logic	Location	Operation	Blk	Output	Destination Rule	Move


```

-----
---
Blue      S5      //get Operator
                WAIT time_inspect MIN
                //free Operator

Net                                1   Blue   SQ5      0.050000 1   MOVE ON

Net                                Blue   S7      0.950000   MOVE ON

Red      S5      //get Operator
                WAIT time_inspect MIN
                //free Operator

Net                                1   Red    SQ5      0.200000 1   MOVE ON

Net                                Red    S7      0.800000   MOVE ON

Blue_order O1      wait 1
                order_time = clock()

Net                                1   Blue   O2      FIRST 1   MOVE ON

Red_order O1      wait 1 order_time = clock()

Net                                1   Red    O2      FIRST 1   MOVE ON

ALL      O2      IF ENTITY() = Blue THEN
                ACCUM blue_batch_size
                IF ENTITY() = Red THEN
                ACCUM red_batch_size

Net                                1   ALL    QS0      FIRST 1   MOVE ON

ALL      QS0      inc WIP      1   ALL    S0      FIRST 1   MOVE ON
Net

ALL      S0      get Operator
                WAIT time_kitting MIN
                free Operator      1   ALL    SQ0      FIRST 1   MOVE ON
Net

ALL      SQ0      IF ENTITY() = Blue THEN
                ACCUM blue_batch_size
                IF ENTITY() = Red      THEN
                ACCUM red_batch_size

Net                                1   ALL    QS1      FIRST 1   MOVE ON

```

ALL Net	QS1		1	ALL	S1	FIRST 1	MOVE ON
ALL	S1	get Operator					
		WAIT time_spring MIN					
Net		free Operator	1	ALL	SQ1	FIRST 1	MOVE ON
ALL	SQ1	IF ENTITY() = Blue THEN					
		ACCUM blue_batch_size					
		IF ENTITY() = Red THEN					
		ACCUM red_batch_size					
Net			1	ALL	QS2	FIRST 1	MOVE ON
ALL Net	QS2		1	ALL	Q2	FIRST 1	MOVE ON
Blue	Q2	get Operator					
		WAIT time_blue_resistor MIN					
		free Operator					
Net			1	Blue	SQ2	FIRST 1	MOVE ON
Red	Q2	get Operator					
		WAIT time_red_resistor MIN					
Net		free Operator	1	Red	SQ2	FIRST 1	MOVE ON
ALL	SQ2	IF ENTITY() = Blue THEN					
		ACCUM blue_batch_size					
		IF ENTITY() = Red THEN					
		ACCUM red_batch_size					
Net			1	ALL	QS3	FIRST 1	MOVE ON
ALL Net	QS3		1	ALL	S3	FIRST 1	MOVE ON
ALL	S3	get Operator					
		WAIT time_LED MIN					
Net		free Operator	1	ALL	SQ3	FIRST 1	MOVE ON
Blue	SQ3	ACCUM blue_batch_size					
Net			1	Blue	QS4	FIRST 1	MOVE ON
Red	SQ3	ACCUM red_batch_size					
Net			1	Red	QS5	FIRST 1	MOVE ON

ALL Net	QS4		1	ALL	S4	FIRST 1	MOVE ON
ALL	S4	get Operator					
		WAIT time_diode MIN					
Net		free Operator	1	ALL	SQ4	FIRST 1	MOVE ON
ALL	SQ4	IF ENTITY() = Blue THEN					
		ACCUM blue_batch_size					
		IF ENTITY() = Red THEN					
		ACCUM red_batch_size					
Net			1	ALL	QS5	FIRST 1	MOVE ON
ALL Net	QS5		1	ALL	S5	FIRST 1	MOVE ON
ALL Net	SQ5		1	ALL	QS6	FIRST 1	MOVE ON
ALL Net	QS6		1	ALL	S6	FIRST 1	MOVE ON
ALL	S6	get Operator					
		WAIT time_rework MIN					
Net		free Operator	1	ALL	SQ6	FIRST 1	MOVE ON
ALL Net	SQ6		1	ALL	S7	FIRST 1	MOVE ON
ALL	S7	IF ENTITY() = Blue THEN					
		INC Blue_done					
		IF ENTITY() = Red THEN					
		INC Red_done					
		DEC WIP					
		done_time = clock()					
		CT= done_time - order_time					
		IF done_time < order_time + 60 THEN					
		inc On_time					
		Total_CT= Total_CT + CT					
Net			1	ALL	O3	FIRST 1	MOVE ON

ALL O3 1 ALL EXIT FIRST 1

* Arrivals *

Entity	Location	Qty each	First Time	Occurrences	Frequency	Logic
Blue_order	O1	1	1	1000	1.666	
Red_order	O1	1	2	300	5	

* Attributes *

ID	Type	Classification
order_time	Integer	Entity
done_time	Integer	Entity
CT	Integer	Entity

* Variables (global) *

ID	Type	Initial value	Stats
QS0_cap	Integer	inf	None
S0_cap	Integer	1	None
SQ0_cap	Integer	inf	None
QS1_cap	Integer	inf	None
S1_cap	Integer	1	None

SQ1_cap	Integer	inf	None
QS2_cap	Integer	inf	None
S2_cap	Integer	1	None
SQ2_cap	Integer	inf	None
QS3_cap	Integer	inf	None
S3_cap	Integer	1	None
SQ3_cap	Integer	inf	None
QS4_cap	Integer	inf	None
S4_cap	Integer	1	None
SQ4_cap	Integer	inf	None
QS5_cap	Integer	inf	None
S5_cap	Integer	1	None
SQ5_cap	Integer	inf	None
QS6_cap	Integer	inf	None
S6_cap	Integer	1	None
SQ6_cap	Integer	inf	None
QS7_cap	Integer	inf	None
S7_cap	Integer	inf	None
SQ7_cap	Integer	inf	None
dummy_Var1	Integer	0	None
blue_batch_size	Integer	6	None
red_batch_size	Integer	4	None
red_done	Integer	0	Basic
blue_done	Integer	0	Basic
WIP	Integer	0	Time Series
Cycle_time	Integer	0	Time Series
On_time	Integer	0	Time Series
profit	Integer	0	Time Series
Total_CT	Integer	0	None

```

*****
*
*                               Macros
*
*****

```

ID	Text
-----	-----
time_kitting	N(2, 0.5)
time_spring	N(2, 0.5)
time_blue_resistor	N(2, 0.5)
time_LED	N(2, 0.5)
time_diode	N(2, 0.5)
time_inspect	N(2, 0.5)
time_rework	N(2, 0.5)
time_red_resistor	N(6, 1)
ExtCost	0

Appendix IV.

ProModel Code for Unpopulated Pull Model Shell

```

*****
*
*
*           Formatted Listing of Model:
*
*           J:\Thesis\Project\pull.MOD
*
*****

```

```

Time Units:           Minutes
Distance Units:       Feet
Initialization Logic:  ANIMATE 25

```

```

ORDER SQ7_cap blue TO SQ6
ORDER QS7_cap red TO QS6

```

```

ORDER SQ5_cap blue TO SQ5
ORDER QS5_cap red TO QS5

```

```

ORDER SQ4_cap blue to SQ4

```

```

ORDER SQ3_cap blue TO SQ3
ORDER QS3_cap red TO QS3

```

```

ORDER SQ2_cap blue TO SQ2
ORDER QS2_cap red TO QS2

```

```

Termination Logic:           WIP =
SQ7_cap+QS7_cap+SQ5_cap+QS5_cap+SQ4_cap+SQ3_cap+QS3_cap+SQ2_cap+QS2_cap

Profit = blue_done*8 + red_done*5 -
(S7_cap+S5_cap+S4_cap+S3_cap+S2_cap)*50 - ExtCost - WIP*1

Cycle_time = Total_CT / (red_done+blue_done)

```



```
*****
*                               Locations                               *
*****
```

Name	Cap	Units	Stats	Rules	Cost
-----	-----	-----	-----	-----	-----
QS0	QS0_cap	1	Time Series Oldest, ,		
#					
#Kitting					
S0	S0_cap	1	Time Series Oldest, ,		
SQ0	SQ0_cap	1	Time Series Oldest, ,		
QS1	QS1_cap	1	Time Series Oldest, ,		
#					
#Die-attach					
S1	S1_cap	1	Time Series Oldest, , First		
SQ1	SQ1_cap	1	Time Series Oldest, ,		
QS2	QS2_cap	1	Time Series Oldest, ,		
#					
#Wire bond					
S2	S2_cap	1	Time Series Oldest, , First		
SQ2	SQ2_cap	1	Time Series Oldest, ,		
QS3	QS3_cap	1	Time Series Oldest, ,		
#					
#Trimming					
S3	S3_cap	1	Time Series Oldest, , First		
SQ3	SQ3_cap	1	Time Series Oldest, ,		
QS4	QS4_cap	1	Time Series Oldest, ,		
#					
#Molding					
S4	S4_cap	1	Time Series Oldest, , First		
SQ4	SQ4_cap	1	Time Series Oldest, ,		
QS5	QS5_cap	1	Time Series Oldest, ,		
#					
#Inspect					

```

S5          S5_cap 1      Time Series Oldest, , First
SQ5         SQ5_cap 1     Time Series Oldest, ,
QS6         QS7_cap 1     Time Series Oldest, ,
#
#Rework
S6          S6_cap 1      Time Series Oldest, , First
SQ6         SQ7_cap 1     Time Series Oldest, ,
dummy4      SQ6_cap 1     Time Series Oldest, ,
#
#Warehouse
dummy5      S7_cap 1      Time Series Oldest, ,
dummy6      QS6_cap 1     Time Series Oldest, ,
O1          inf 1         Time Series Oldest, ,
O2          inf 1         Time Series Oldest, ,
O3          inf 1         Time Series Oldest, ,

```

```

*****
*                               *
*                               *
*****

```

Name	Speed (fpm)	Stats	Cost
Blue	150	Time Series	
Red	150	Time Series	
Blue_order	1500	Time Series	
Red_order	1500	Time Series	

```

*****
*                               *
*                               *
*****

```

Name	Type	T/S	From	To	BI	Dist/Time	Speed Factor

Net	Passing	Speed & Distance	N1	N2	Bi	2.23	1
			N2	N3	Bi	2.82	1
			N3	N4	Bi	1.00	1
			N4	N5	Bi	1.00	1
			N5	N6	Bi	8.54	1
			N6	N7	Bi	2.23	1
			N7	N8	Bi	4.24	1
			N8	N9	Bi	13.60	1
			N9	N10	Bi	4.00	1
			N10	N11	Bi	2.23	1
			N11	N12	Bi	17.11	1
			N12	N13	Bi	6.00	1
			N13	N14	Bi	3.00	1
			N14	N15	Bi	14.31	1
			N14	N18	Bi	20.09	1
			N15	N16	Bi	4.12	1
			N16	N17	Bi	2.00	1
			N17	N18	Bi	2.00	1
			N18	N19	Bi	1.41	1
			N19	N20	Bi	5.83	1
			N20	N21	Bi	15.13	1
			N20	N22	Bi	16.03	1
			N21	N26	Bi	7.81	1
			N26	N27	Bi	5.09	1
			N27	N22	Bi	12.20	1
			N22	N23	Bi	2.23	1
			N23	N24	Bi	4.47	1
			N24	N25	Bi	1.00	1
			N25	N1	Bi	63.00	1

* Interfaces *

Net	Node	Location

Net	N1	O1
	N2	O2
	N3	QS0
	N4	S0
	N5	SQ0
	N6	QS1
	N7	S1
	N8	SQ1
	N9	QS2
	N10	S2
	N11	SQ2
	N12	QS3
	N13	S3
	N14	SQ3
	N15	QS4
	N16	S4
	N17	SQ4
	N18	QS5
	N19	S5
	N20	SQ5
	N21	SQ6
	N22	S6
	N23	QS6
	N24	dummy4
	N25	dummy5
	N26	dummy6
	N27	O3

```

*****
*                                     *
*                               Mapping                               *
*                                     *
*****

```

Net	From	To	Dest

Net	N2	N1	
	N3	N2	
	N4	N3	
	N5	N4	
	N6	N5	
	N7	N6	
	N8	N7	
	N9	N8	
	N10	N11	
	N11	N12	
	N12	N13	
	N13	N14	
	N14	N18	
	N15	N16	
	N16	N17	
	N17	N18	
	N18	N19	
	N19	N20	
	N20	N22	
	N21	N20	
	N22	N23	
	N23	N24	
	N24	N25	
	N25	N1	
	N26	N27	
	N27	N22	
	N1	N2	
	N10	N9	
	N11	N10	
	N2	N3	
	N3	N4	
	N12	N11	
	N13	N12	

N14	N13
N4	N5
N5	N6
N15	N14
N16	N15
N6	N7
N18	N14
N7	N8
N19	N18
N20	N19
N8	N9
N1	N25
N9	N10
N22	N20
N23	N22
N24	N23
N25	N24
N14	N15
N17	N16
N18	N17
N20	N21
N26	N21
N27	N26
N21	N26
N22	N27

```

*****
*                                     *
*                               Processing                               *
*                                     *
*****

```

	Process	Routing	
Entity Logic	Location Operation	Blk Output	Destination Rule Move

```

-----
Blue      S5      WAIT time_inspect
Net
1 Blue      S6      0.050000 1 MOVE ON
Net
Blue      SQ6      0.950000 MOVE ON
Red      S5      WAIT time_inspect
Net
1 Red      S6      0.050000 1 MOVE ON
Net
Red      QS6      0.950000 MOVE ON
Blue_order O1      Order_time = clock()
wait .1
Net
1 Blue_order O3      FIRST 1 MOVE ON
Blue_order O3      JOIN 1 Blue
ORDER 1 Blue to SQ1
inc blue_done
Done_time = clock()
IF Done_time < Order_time + 120 THEN
inc On_time
CT= Done_time - Order_time
Total_CT = Total_CT + CT
Net
//INC WIP      1 Blue_order EXIT      FIRST 1 MOVE ON
Blue      SQ6      1 Blue      O3      JOIN 1 SEND 1
Blue TO S5
MOVE ON
Net
Blue      S6      WAIT time_rework      1 Blue      SQ6      FIRST 1 MOVE ON
Net
Blue      SQ5      1 Blue      S5      SEND 1 SEND 1
Blue TO S4

```

							MOVE ON
Net							
Blue	S4	WAIT time_diode	1	Blue	SQ5	FIRST 1	MOVE ON
Net							
Blue	SQ4		1	Blue	S4	SEND 1	send 1
Blue to S3							
							MOVE ON
Net							
Blue	S3	WAIT time_LED	1	Blue	SQ4	FIRST 1	MOVE ON
Net							
Blue	SQ3		1	Blue	S3	SEND 1	send 1
Blue to S2							
							MOVE ON
Net							
Blue	S2	WAIT time_blue_resistor					
			1	Blue	SQ3	FIRST 1	MOVE ON
Net							
Blue	SQ2		1	Blue	S2	SEND 1	send 1
blue to S1							
							MOVE ON
Net							
Blue	S1	WAIT time_spring	1	Blue	SQ2	FIRST 1	MOVE ON
Net							
Blue	SQ1		1	Blue	S1	SEND 1	MOVE ON
Net							
Red_order	O1	wait .1					
		order_time = clock()					
			1	Red_order	O3	FIRST 1	MOVE ON
Net							
Red_order	O3	JOIN 1 Red					
		ORDER 1 Red to QS1					
		inc red_done					
		done_time = clock()					
		IF done_time < order_time + 120 THEN					
		inc On_time					
		CT= done_time - order_time					

Total_CT = Total_CT + CT

```

Net
    1 Red_order EXIT FIRST 1 MOVE ON

Red red TO S5 QS6 1 Red O3 JOIN 1 SEND 1

Net
    MOVE ON

Red Net S6 WAIT time_rework 1 Red QS6 FIRST 1 MOVE ON

Red red TO S3 QS5 1 Red S5 SEND 1 SEND 1

Net
    MOVE ON

Red Net S3 WAIT time_LED 1 Red QS5 FIRST 1 MOVE ON

Red red TO S2 QS3 1 Red S3 SEND 1 SEND 1

Net
    MOVE ON

Red S2 WAIT time_red_resistor

Net
    1 Red QS3 FIRST 1 MOVE ON

Red Red TO S1 QS2 1 Red S2 SEND 1 SEND 1

Net
    MOVE ON

Red Net S1 WAIT time_spring 1 Red QS2 FIRST 1 MOVE ON

Red Net QS1 1 Red S1 SEND 1 MOVE ON

```

```

*****
* Arrivals *
*****

```

Entity	Location	Qty each	First	Time	Occurrences	Frequency	Logic
Blue_order	O1	1	1	inf	1.666		
Red_order	O1	1	0	inf	5		

```
*****
*                               Attributes                               *
*****
```

ID	Type	Classification
Order_time	Integer	Entity
Done_time	Integer	Entity
CT	Integer	Entity

```
*****
*                               Variables (global)                       *
*****
```

ID	Type	Initial value	Stats
QS0_cap	Integer	inf	None
#			
#Kitting			
S0_cap	Integer	Inf	None
SQ0_cap	Integer	inf	None
QS1_cap	Integer	1	None
#			
#Die-attach			
S1_cap	Integer	1	None
SQ1_cap	Integer	1	None
QS2_cap	Integer	1	None
#			
#wire-bond			
S2_cap	Integer	1	None
SQ2_cap	Integer	1	None
QS3_cap	Integer	1	None

```

#
#trimming
    S3_cap      Integer      1      None
    SQ3_cap      Integer      1      None
    QS4_cap      Integer      1      None
#
#molding
    S4_cap      Integer      1      None
    SQ4_cap      Integer      1      None
    QS5_cap      Integer      1      None
#
#inspection
    S5_cap      Integer      1      None
    SQ5_cap      Integer      1      None
    QS6_cap      Integer      1      None
#
#rework
    S6_cap      Integer      1      None
    SQ6_cap      Integer      1      None
    QS7_cap      Integer      1      None
#
#warehouse
    S7_cap      Integer      1      None
    SQ7_cap      Integer      1      None
    dummy_Var1   Integer      0      None
    Red_batch_size Integer      1      None
    blue_batch_size Integer      1      None
    Red_Done      Integer      0      Basic
    Blue_Done      Integer      0      Basic
    WIP            Integer      0      Time Series
    Cycle_time     Integer      0      Time Series
    On_time        Integer      0      Time Series
    Profit         Integer      0      Time Series
    Total_CT       Integer      0      Basic

```

```

*****
*                                     Macros                                     *
*****

```

ID	Text
-----	-----
time_kitting	0
time_spring	N(2, 0.5)
time_blue_resistor	N(2, 0.5)
time_LED	N(2, 0.5)
time_diode	N(2, 0.5)
time_inspect	N(2, 0.5)
time_rework	N(2, 0.5)
time_red_resistor	N(6, 1)
ExtCost	-20

Appendix V.
Lean Simulation Survey

Participants information:

1. Simulation attended: ☐ live, ☐ computer
2. Education: ☐ some high school, ☐ high school, ☐ some college, ☐ college, ☐ advanced degree
3. Mfg related working experience: ☐ 0-1years, ☐ 1-3years, ☐ 3-7years, ☐ over 7years
4. Exposure to Lean Mfg prior to this class: ☐ none, ☐ minimal, ☐ some, ☐ guru

Please express your opinion on following statements using scale from 1 to 6:

- 1 - strongly disagree
2 - disagree
3 - somewhat disagree
4 - somewhat agree
5 - agree
6 - strongly agree
5. 1 2 3 4 5 6 The lecture and lecturer offered excellent coverage on lean manufacturing.
6. 1 2 3 4 5 6 The simulation helped you better understand the benefits of lean manufacturing.
7. 1 2 3 4 5 6 The simulation effectively illustrated and reinforced all the important principles and concepts taught in the lecture.
8. 1 2 3 4 5 6 The simulation was interesting and engaging.
9. 1 2 3 4 5 6 The simulation was clearly worded or instructed, and easy to understand.
10. 1 2 3 4 5 6 The simulation presented a logical and gradual transition from push production to pull production.
11. 1 2 3 4 5 6 The simulation clearly demonstrated performance metrics resulting from the lean transformation.
12. 1 2 3 4 5 6 The simulation helped you clearly see the impact of interdependencies and variability on system performance.
13. 1 2 3 4 5 6 You now know how to apply lean principles and concepts to your work and even to life.
14. What did you particularly like about the simulation?
15. What did you particularly dislike about the simulation?

Appendix VI.

T-Test Statistic Analysis for Survey Results

Survey results and averages

T-test results

t-Test: Two-Sample Assuming Unequal Variances

	<i>Control</i>	<i>Treat</i>
Mean	5.333333	4.3125
Variance	0.30506	0.455357
Observations	15	8
Hypothesized Mean Difference	0	
df	12	
t Stat	3.672705	
P(T<=t) one-tail	0.001596	
t Critical one-tail	1.782287	
P(T<=t) two-tail	0.003191	
t Critical two-tail	2.178813	