PH.D. THESIS

Design for Production: Using Manufacturing Cycle Time Information to Improve Product Development

by Mandar Chincholkar
Advisor: Dr. Jeffrey W. Herrmann

PhD 2002-10

ABSTRACT

Title of Dissertation: DESIGN FOR PRODUCTION: USING MANUFACTURING CYCLE TIME INFORMATION TO IMPROVE PRODUCT DEVELOPMENT

Mandar M. Chincholkar, Doctor of Philosophy, 2002

Dissertation directed by: Associate Professor Jeffrey W. Herrmann
Department of Mechanical Engineering
and
Institute for Systems Research

Product development teams employ many methods and tools as they design, test, and manufacture a new (or improved) product. It is important that the product development team understand how their design decisions affect manufacturing system performance. Having this feedback early in the design process avoids rework loops needed to solve problems of manufacturing capacity or cycle time. The team can incorporate this information and associated costs into a design decision problem aimed at choosing the best possible product design.

It is clear that the product design, which requires a specific set of manufacturing operations, has a huge impact on the manufacturing cycle time. Reducing *manufacturing cycle time* has many benefits, including but not limited to lower inventory, reduced costs, improved product quality, faster response to customer orders, increased flexibility and a reduced time-to-market.

Design For Production (DFP) refers to methods that evaluate a product design by comparing its manufacturing requirements to available capacity and estimating manufacturing cycle time. DFP can be used to design the product in a way that decreases required capacity, reduces the manufacturing cycle time, or otherwise simplifies production.

To understand how a product design impacts manufacturing system performance, this research develops analytical (not simulation) models to quantify how introducing a new product increases congestion in the manufacturing system. It presents approaches that use this information intelligently and make suggestions on product redesign and manufacturing system improvements. Similar models are also developed for manufacturing systems with process drift, a condition causing a process to deviate from expected processing parameters resulting in a reduced yield at that station. This work presents models for evaluating how embedding passives into a printed circuit board affects not only the processing times at each step in the manufacturing process but also the overall manufacturing system behavior. Finally, this dissertation demonstrates the importance of the DFP approach by presenting a comprehensive perspective on the economic impacts of reducing manufacturing cycle time. Through these models and relationships, this research aims to understand the issues and impacts associated with the design for production approach and provide better tools that improve product development.

Keywords : design for manufacture, design for production, queuing, product design, product development, manufacturing cycle time, economic impact, embedded passives, process drift.

DESIGN FOR PRODUCTION: USING MANUFACTURING CYCLE TIME INFORMATION TO IMPROVE PRODUCT DEVELOPMENT

by

Mandar M. Chincholkar

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2002

Advisory Committee:

Associate Professor Jeffrey W. Herrmann, Chairman/Advisor
Professor Michael O. Ball
Associate Professor Satyandra K. Gupta
Professor Edward B. Magrab
Associate Professor Peter A. Sandborn

DEDICATION

To my parents

ACKNOWLEDGEMENTS

Through the course of this research, a number of people have contributed to and helped in various ways. This doctoral work has taught me a whole lot of invaluable lessons which though not a part of this dissertation will continue to guide me throughout my life. This work would be incomplete without acknowledging various people associated with my work, their efforts and contributions.

Firstly, I would like to whole-heartedly thank my advisor, Associate Professor Jeffrey W.Herrmann, for his guidance, invaluable help, ideas and patience. Our often-held and long drawn discussions of various minute aspects of the problem helped me better understand the problem and associated intricacies. I really appreciate Dr.Herrmann for his patience during these at times frustrating discussions and tolerating my temperamental behavior. I have imbibed a number of imitable qualities from Dr.Herrmann including the ability to analyze a problem without losing sight of the big picture of which the problem is a small part, the ability to scrutinize and critique both self written works as well as those of other researchers, to persevere to solve a tough problem at times changing the viewpoint if needed, to name a few. I hope to emulate these qualities in my professional career.

Next I wish to express my gratitude towards my dissertation committee, Dr.Michael Ball, Dr.Satyandra Gupta, Dr.Edward Magrab and Dr.Peter Sandborn for their ideas during the proposal presentation and problem definition effort. Thanks are also due to the committee for their guidance and evaluation of my work.

I am grateful for the superb facilities of the Computer Integrated Manufacturing Laboratory which served as a second home during the time I was working on my research. I wish to thank Dr.Edward Lin for his help with the modalities associated with the computing and other facilities in the lab. I also appreciate his help with my work and for the suggestions and ideas to improve my research. I also take this opportunity to thank all the students and researchers in the CIM Lab, both past and present, for their support and help in making my stay here an enjoyable and memorable one.

Finally I acknowledge the efforts of my parents who always stood by me during the good times and most importantly during the not so good times. I am forever indebted to them for their love and support throughout my professional and personal academic career. As I stand here, close to the end of my academic career, I look back and realize the sheer magnitude of their support over the years. My mother has always been there when I needed her, coaxing and egging me when the chips were down and deriving great happiness from my minutest achievements. She has always taught me to believe in myself and my dreams. My father has been a strong support who I have always looked to for guidance. He has taught me to look at the positive side of life and

to always aim for the stars. My brother also deserves special mention here for his support and help over the years. It is impossible to fully express my gratitude and appreciation for all my parents have done for me, and this acknowledgement is but an insignificant attempt.

The material in this dissertation is based upon work supported by the National Science Foundation under Grant No. DMI 97 - 13718. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author and do not necessarily reflect the views of the National Science Foundation. I wish to thank the National Science Foundation for their support.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1

Introduction

Product variety is continuously increasing in today's world market. In such an environment, the required philosophy for a company's survival, is the constant replacement of old products with new ones, improved variations of current products and completely new products. Product development has thus become a very crucial aspect of corporate competition. The design and development of a product is a complex process involving numerous considerations such as market analysis, requirements definition, conceptual design, detailed design, materials and process selection, optimization, process control, testing and evaluation, costing, manufacturing and production, and marketing [1, 5, 14, 18].

Successful product development requires the definition of various measures of performance for different phases of the product life cycle and methods to predict these performance measures. Accurately predicting these metrics enables the product development team to develop the product "first time right" thereby avoiding or at least minimizing development costs and product redesign. Such performance measures may be numerous and influencing various aspects of the development cycle, from concept generation to product delivery.

Traditionally, while designing a new product, each phase in product development is completed before the product passes to the next phase. This is known as “over the wall” design. For example, the designer finalizes the detailed designs before passing them to the manufacturing phase. There are, however, potential problems associated with this way of designing a product. Among these are the following [131]:

1. There is a loss of abstract and implied information as the product passes from one phase to another. Each phase receives a different interpretation of the customer requirements. Thus there exists some risk that the final product will not completely satisfy the customer requirements.

2. There is significant loss of time and effort in returning the designs to the design phase from the post-design phases to correct any mistakes or shortcomings discovered in these phases.

3. By finalizing the designs in the design phase, the designer utilizes only his knowledge of the design scenario. The knowledge of the post-design operations such as manufacturing cannot be incorporated into the designs. As a result, opportunities for product optimization over all the processes are missed due to the lack of effective communication between the two operations.

The *Design for X (DFX)* approach to designing a product aims to alleviate some of these problems by designing the product while keeping under consideration the performance of the design during other phases of its life cycle [47, 68, 69]. The DFX methodology evaluates product designs along with associated life cycle requirements such as those associated with manufacturability, schedulability, recyclability, dis-assemblability, producability and so on, to determine the product performance during these phases. It attempts to identify possible problems and shortcomings in the designs. DFX searches for solutions to these problems, proposing changes to the product or processes and prioritizes these ideas based on evaluation of the effects of these suggestions on the anticipated performance of the product, thereby avoiding redesigns later in the process.

1.1 Motivation

Associated with each phase in the product development process are a time and a cost which may be attributed to the requirements of that phase of the process as well as possible rework cycles, constituting total product development time and cost. In order to design a product so as to maximize product profitability, it is necessary to understand the economic implications of the product design during its life cycle. To achieve this goal, it is first essential to model this economic impact along with quantifiable metrics.

This dissertation studies the association between product design and manufacturing system performance. Models developed as a part of this research along with suitable performance metrics delineate the product design - manufacturing system relationship. Such metrics include manufacturing cycle time, WIP and throughput. These key factory-level performance measures affect financial measures such as cost, revenue and profitability.

Manufacturing cycle time may be defined as the total time spent by the product in the manufacturing system. Manufacturing cycle time is the interval that elapses as the manufacturing system performs all of the operations necessary to complete a work order. This manufacturing cycle time has many components, including move, queue, setup, and processing times. The terms *Throughput Time* or *Flow Time* are also used to describe the time spent by a product in a manufacturing system. The terms *Manufacturing Cycle Time* and *Throughput Time* are used interchangeably throughout this dissertation.

An examination of the impact of the product design would be incomplete without assessing the importance of manufacturing cycle time to product profitability, evaluating the impact of the manufacturing cycle time on the product life cycle and modeling these relationships. Such an analysis requires developing maps and models that describe how modifying the manufacturing cycle time affects costs and revenues for a product. Such maps and models can then be used as part of a more comprehensive product profitability assessment schema.

In summation, this research is motivated by the need to understand the impact of a product design on manufacturing system performance through certain production metrics and to translate these metrics into quantities that contribute to overall product profitability.

1.2 Design for Production

This dissertation introduces the term *Design for Production* (DFP) to describe methods that determine if a manufacturing system has sufficient capacity to achieve the desired throughput and methods that estimate the manufacturing cycle time. These methods require information about a product's design, process plan, and production quantity along with information about the manufacturing system that will manufacture the product. Knowing the capabilities of the manufacturing system can help a designer evaluate the feasibility of alternative product designs and use the information to either choose from proposed alternatives or modify the existing product design. This will remove the need to actually put the product through the post-design phases before discovering any infeasibility or prohibitive costs associated with making the product in the given manufacturing system.

Design for Manufacture (DFM) evaluates the materials, the required manufacturing processes, and the ease of assembly for the product. (This discussion will use the term manufacturing to describe both fabrication and assembly, and will include design for assembly as part of design for manufacturing.) Therefore, both DFM and DFP are related to the product's manufacture. DFM evaluates manufacturing capability and measures the manufacturing cost. It focuses on the individual operations that manufacturing requires. On the other hand, DFP evaluates quantity and rate of parts that the manufacturing system can output and how long each order will take. That is, it evaluates manufacturing capacity and measures the manufacturing cycle time. Moreover, this approach requires information about the manufacturing system as a whole. Like DFM, DFP can lead a product development team to consider changing the product design. In addition, DFP can provoke suggestions to improve the manufacturing system. DFP is likely to find greater application to new product introduction into an existing manufacturing system already producing certain products.

DFM approaches that generate process plans and estimate processing times can be the first DFP step, since DFP methods may use this information. Traditional DFM approaches can also improve manufacturing cycle time since they minimize the number of parts and reduce the processing time of each operation. DFP approaches may be distinguished by their focus on evaluating manufacturing capacity and manufacturing cycle time. Different research works have used various names to describe DFP approaches, including design for existing environment [132], design for time-to-market [50], design for localization [81], design for speed [97], design for schedulability [79], and design for manufacturing system performance [122]. Some researchers have reported case studies in which product designs were modified to improve production.

This research studies situations where a new product, which may be an improvement of an existing design or a completely different product, will be introduced into a given manufacturing system already processing a set of products. This dissertation presents models for

- understanding how introducing a new product into an existing manufacturing system affects the performance of the manufacturing system,
- understanding how changes in the product design affect manufacturing system performance, and
- understanding the economic implications of reducing manufacturing cycle time.

Specifically, this dissertation presents models and tools for estimating the manufacturing cycle time and throughput of a manufacturing system. It also discusses and models how manufacturing cycle time affects costs, revenue and profitability.

Thus, through the DFP approach, this research aims to provide the product development team with methods to evaluate the performance of the manufacturing system before production begins. Tools based on the approach may be applied during the conceptual or embodiment or detailed design phases of the design development process. Based on the requirements of the approach, however, it may be best suited for the embodiment phase, when the design team has a reasonable idea of various design instances and before all aspects of the design have been finalized.

Having said this, the utility of the approach in the chronology of the development process depends on the specific product development process. Also, it must be noted that the DFP approach addresses only one aspect of the product life cycle. While designing a product, information obtained using this approach must be combined with information from other phases of the life cycle. This dissertation presents applications that reflect the use of the DFP approach and associated tools for a typical product development process.

1.3 Organization

This dissertation is organized as follows: Chapter 2 reviews previous work that researches various product development concepts. It surveys literature documenting the importance of product design to post-design product development processes. Chapter 3 presents manufacturing system models based on queuing network analysis. It demonstrates how these models play a significant role in the DFP approach. Chapter 4 applies the models and algorithms developed to understanding the impact of embedding passive components into the substrate of a printed circuit board. Chapter 5 presents algorithms to analyze manufacturing processes and systems with process drift, where defective parts processed at a workstation are detected at a subsequent inspection station. Chapter 6 describes the economic benefits of reducing manufacturing cycle time. Chapter 7 summarizes the work done as part of this research. Further, it lists the research contributions from this work and presents potential ideas for future work that would extend the approaches and models presented as a part of this research.

Chapter 2

Literature Review

2.1 Introduction

Product development is a complicated process starting with a detailed target assessment, comprising an extensive research of the current market scenario, available products and understanding product customer requirements. This is followed by various steps from defining product specifications based on these requirements to packaging and dispatching the final product to its final destination. An effective product development process has a number of benefits [88] such as:

1. Increased Revenue
   (a) Increased product life-cycle revenue
   (b) Increased market penetration as a result of being first to market
   (c) Success in time-sensitive markets
   (d) More successful products
2. Improved product development productivity

    (a) Shorter development cycle times

    (b) Less development waste

    (c) Better resource utilization

    (d) Better ability to attract and retain technical talent

3. Operational Efficiencies

    (a) Design for manufacturability, serviceability and other characteristics

    (b) Higher-quality products

    (c) Lower engineering change order costs

    (d) Improved predictability of launch

New product introduction involves a long sequence of operations. The number of steps and the steps themselves depend on the type of product being developed. The schema for a product realization process may be outlined as follows:

1. The company analyzes feedback from the market about the current product performance and uses the information to identify need for modifications or new product launch.

2. Depending on the feedback, the research and product development teams discuss implications of the proposed modifications or the new product design. The costs involved, complexities (technical and other), and time required influence the decision about the changes or new product introduction.

3. Depending on discussions with the various constituents of a product development team such as manufacturing, reliability, a “new product concept” is formulated, which is then presented to the management for approval and authorization for further design work.

4. During the design process, in an ideal team environment, other development processes such as cost estimation may proceed simultaneously. Design of long lead time facilities, equipment and tooling may also commence.

5. Once a firm product concept has been finalized, resources are allocated towards tooling, equipment, testing, advertising, service training and allied activities. Design work continues while these decisions are being made.

6. The production equipment and facilities expansions are ordered and their construction proceeds. The long lead time tooling work is also begun. The product and process design continues during this phase.

7. Prototypes are built once the design work nears completion, and performance and reliability testing of the prototypes is undertaken. Designs are evaluated for conformance to original product plan and adherence of the project to its time schedule is examined.

8. Designs are finalized and sent for manufacturing and the remaining tooling, gauging and other equipment is ordered.

9. The field and life testing of the product referred to as *beta testing* proceeds. Potential customers are given the product for feedback and evaluation.

10. The production facility installations are completed. Complete detailed drawings and finalized bill-of-materials are released to the production department. A pilot production run is made and quality control features are adopted. Engineering changes are made depending on customer feedback and pilot production results.

11. Regular production is commenced and the products are shipped to the market.

Figure 2.1 shows one such combination of steps that represents the product realization process [14]. This is not an ideal sequence but only a schema to illustrate the basic approach and activities involved. The sequence would differ in different organizations.

Although the earlier paragraphs detail the product development process as a series of steps, in reality it hardly proceeds in such a regular manner. In practice, the process usually comprises of a network of paths that the product follows including a number of feedback loops between the various stages of the product development cycle. These feedback loops help ensure feasibility of the product design from perspective of post-design processes. However, these feedback loops, though useful in helping the product development team converge on a feasible design solution, nevertheless increase the time required and costs involved from demand recognition to the actual product launch and often contribute a great deal to the costs involved in making the product. Figure 2.2 depicts the relation between the project cost and the product development cycle time.

In order to make the product quickly and with minimum costs, it is prudent to minimize the number of feedback loops in the development process. This will streamline product development, taking it closer to the goal of achieving a serial procedure as outlined earlier.

For any corporate organization, short delivery times, periodic product innovations, and shorter time-to-market are very important attributes. Those organizations which can achieve and maintain such values usually outperform all competition.

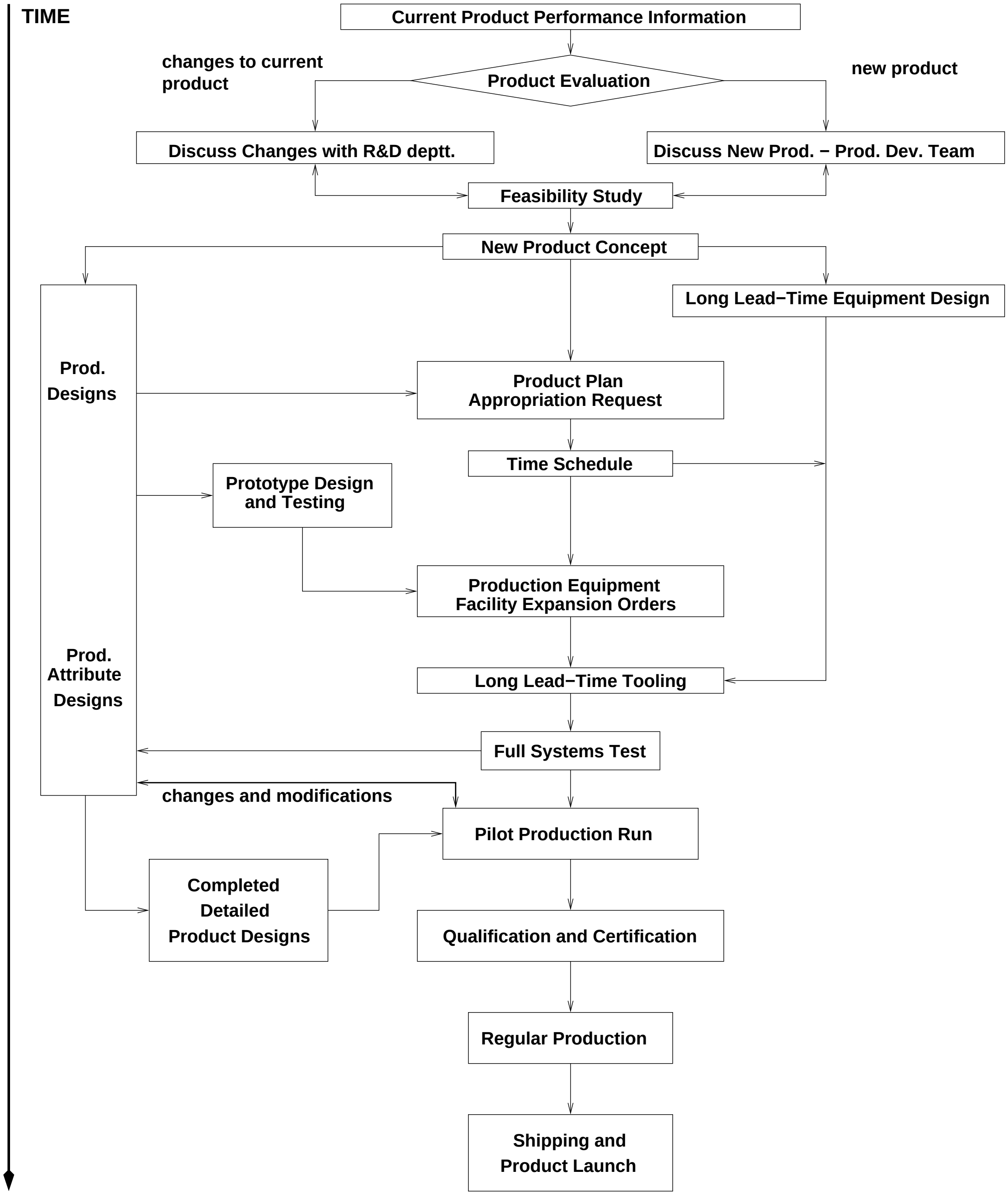
TIME
Current Product Performance Information
changes to current product
Product Evaluation
new product
Discuss Changes with R&D deptt.
Discuss New Prod. − Prod. Dev. Team
Feasibility Study
New Product Concept
Long Lead−Time Equipment Design
Prod. Designs
Product Plan Appropriation Request
Time Schedule
Prototype Design and Testing
Production Equipment Facility Expansion Orders
Prod. Attribute Designs
Long Lead−Time Tooling
Full Systems Test
changes and modifications
Pilot Production Run
Completed Detailed Product Designs
Qualification and Certification
Regular Production
Shipping and Product Launch

Figure 2.1: Product Realization Process

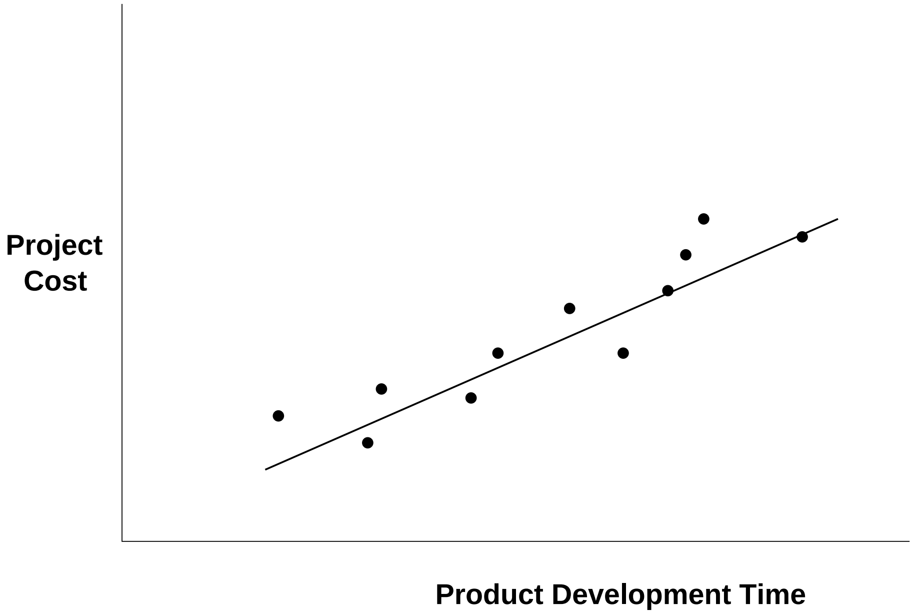

Figure 2.2: Project cost increases with increase in development time [88]

2.2 Concurrent Engineering

Syan [127] defines concurrent engineering as: *"Concurrent engineering is a systematic approach to the integrated, concurrent design of products and their related processes, including manufacture and support. This approach is intended to cause the developers, from the onset, to consider all elements of the product life cycle from concept through disposal, including quality, cost, schedule and user requirements"* . It follows from this definition that the development team in a concurrent engineering environment in aided in establishing a degree of clairvoyance in analyzing problems, that the product may to encounter during various stages of product development, aided by knowledge of development processes and tools like optimization or graph theory. The aim is to alleviate these problems at an early stage of development by making suitable development decisions. Magrab [86] presents a set of techniques under the IP^{2}D^2 methodology which could also be applied to facilitate concurrent engineering towards better product development. The proposed IP^{2}D^2 method broadly indicates the overlapping, interacting, and iterative nature of all the aspects that impact the product realization process. It is a continuous process by which a product's cost, performance, features, and values lead to a company's increased profitability and market share.

The benefits of concurrent engineering [71] are numerous and wide-spread throughout the product development cycle, including but not limited to reductions in time to market, reduced design changes and design iterations, improvements in the manufacturability, assembly, serviceability, recyclability and overall quality of the products. Hauptman and Hirji [54] survey the applicability of concurrent engineering to product development in great detail based on the study of a multitude of product development projects using concurrent cross-functional teams.

One of the methodologies for achieving this concurrency in the product and process engineering [6, 91] is based on the formulation of an optimization problem with constraints drawn from various aspects of the product life-cycle. Tan *et al.* [130, 131] suggest a model which brings together different phases of the product development process using an intelligent agent framework. The approach begins from representing customer requirements and iteratively generates the final designs based on cost evaluation of the initial designs. Initial designs are provided to the system and the set of constraints between the product and the outside systems do not change during the iterations. The system aims to create a final design while taking into consideration most aspects of the product development process. All information exchange occurs between agents governing different stages of the development life such as process-planning agent, simulation agent, design agent, and critiquing agent. Cutkosky and Tenenbaum [31] present a system based on

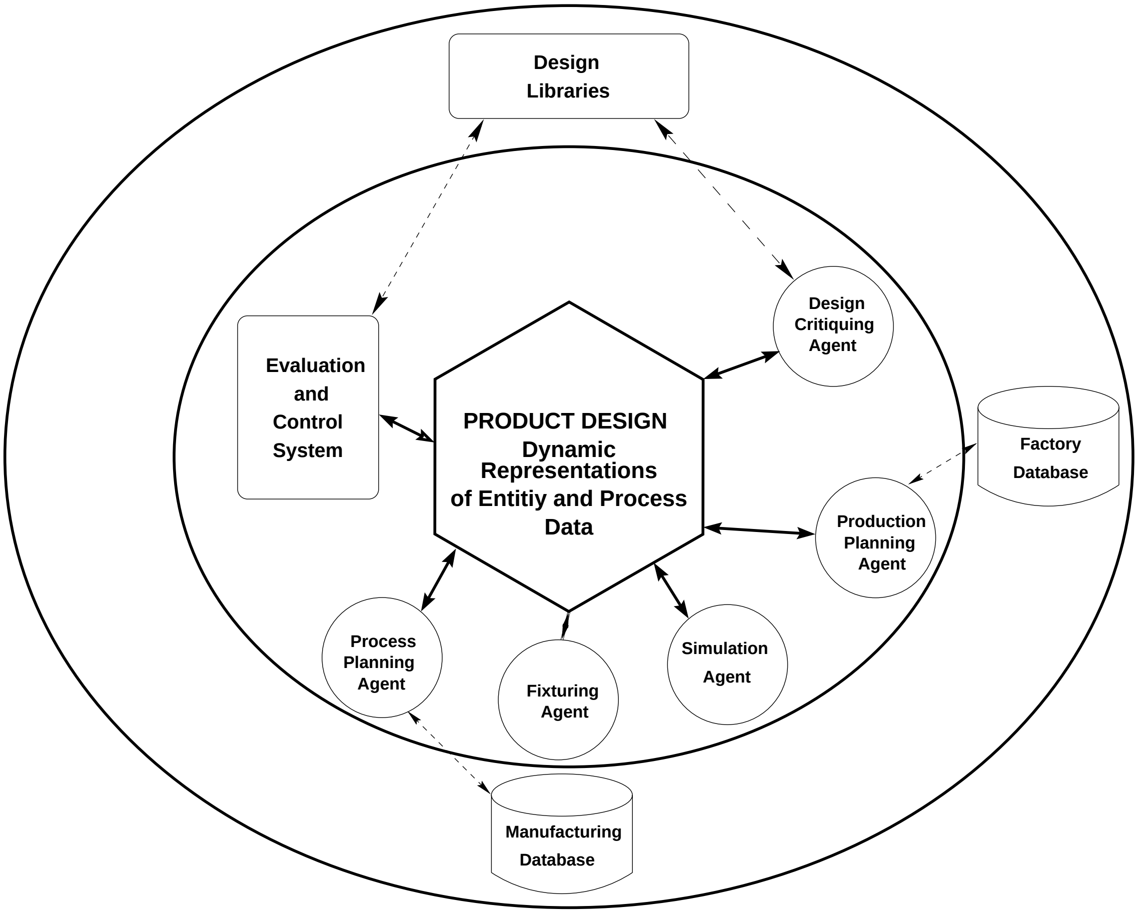

Figure 2.3: The intelligent agent framework proposed by Cutkosky and Tenenbaum

a similar architecture. Their paper provides details of the system architecture, representations and the software module design. Figure 2.3 represents the general framework proposed. Brookes *et al.* [16] list some relevant case studies of implementation of the concurrent engineering methodology.

The optimization modules developed for finding the best product design tend to use the product cost [32, 40, 99, 122, 142] as the objective function. Soundar and Bao [122] suggest a formulation for the problem with cost as the objective and the constraints based on

1. the critical design attributes as surface finish and weight
2. the critical manufacturing system performance attributes as WIP and queue time

and formulating a utility function to find an optimal solution. Wei and Egbelu [142] propose algorithms to model various manufacturing costs. Once these are decided, the minimum cost manufacturing sequence selection is modeled as a mixed integer problem with minimizing the summation of the costs as the objective and the operating sequence parameters as the constraints. The aim is to minimize costs while preserving product functionality.

Ball *et al.* [6] take a somewhat different approach to formulating the optimization problem. They propose that the cost be treated as an independent variable. Thus, now the cost may be used as a constraint in lieu of an objective as adopted by the earlier approaches. Often the design problem is split into constituent subproblems and optimization with cost as the objective is performed on individual sub-problems. The disadvantage of this method is that though the cost may be minimized for the local sub-problem, in the global scenario the cost may not always be a minimum. Using the cost as a constraint and solving the overall problem based on tradeoff analysis, ensures that the cost always remains constrained below the specified value. However, it is important to mention that in this case, the cost constraint decision is a designer prerogative and (s)he must be provided with adequate knowledge to help make a well-advised decision.

Thus the problem is a tradeoff problem between the costs and/or time involved in making the product and the product performance. With the optimization formulation it is hoped that the costs and time involved in the post design problems may be optimized against a slightly increased design cost and performance.

2.3 Importance of Design

In Section 2.1, the steps in a typical product development cycle were enumerated. Embedded in the chronology of events that formulate the development cycle, is the allocation and approval of finances without which the progress of the product development is difficult. The "Westinghouse Curve" [14] illustrates how the life cycle cost of a typical product is greatly affected by the decisions made during the early stages of the product design phase. Figure 2.4 shows this curve. From the curve, it may be seen that by the time a product concept is validated, well before the development is completed, 70% of the total budget for development has already been allocated. This underscores the importance of providing the designer with adequate knowledge of the post-design processes [72]. Using this knowledge, the designer can make the "best" design decisions in order that the costs may be minimized during the concept stage itself, well before the product goes into production.

In addition, any required redesign may be implemented in different stages of the product development process. Depending on the stage at which it is implemented, a redesign system behaves in different capabilities. In accordance, during the initial stages of the design process it serves to guide the designer in taking important design decisions [50] like materials and process selection, joint definitions and so on, based on knowledge of post design processes and customer requirements, stored in libraries. Thus, at this stage the approach complements the concurrent engineering philosophy as applied to the design stage, improving the designs by equipping the designer with knowledge from other processes. When the redesign schema is implemented at a later stage, as during the detailed design phase after the initial designs have been formalized, it serves as an evaluator of the preliminary designs.

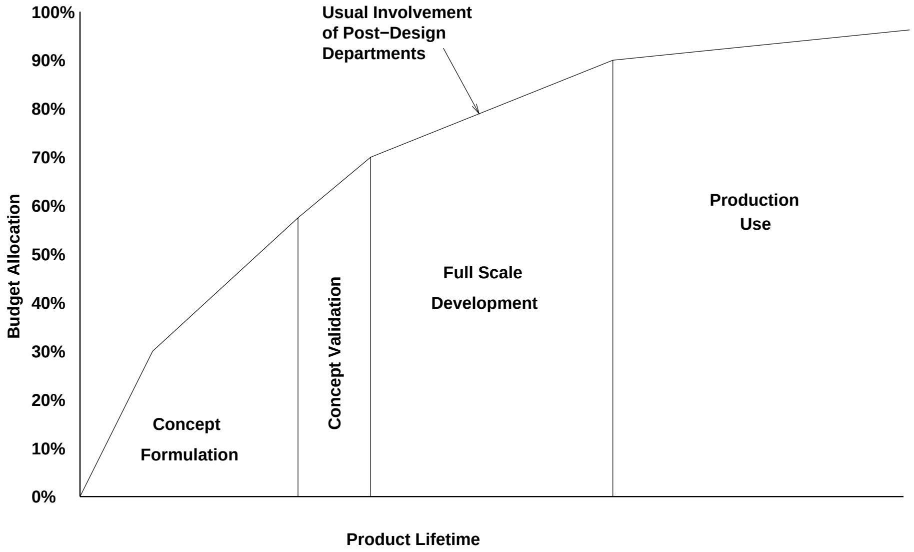

Figure 2.4: Westinghouse Curve [14]

The system attempts to characterize the designs according to the functionality and designer intent.

2.4 Design for X

Magrab [86] defines DFX as: *"Design for X (DFX) may be defined as a knowledge-based approach that attempts to design products that maximize all desirable characteristics such as high quality, reliability, serviceability, safety, user friendliness, environmental friendliness, and short time-to-market in a product design while, at the same time, minimizing lifetime costs, including manufacturing costs"*. The methodology spawns from the idea of designing products while taking into consideration the downstream processes in the product life-cycle and their effects on the product designs [47, 68, 69, 136].

The previous section explained the importance of design in the product development cycle towards overall financial management of the product development project. The DFX methodology comprises analyzing the product and processes involved and their performance characteristics, identifying the problems and shortcomings in these and highlighting them, searching for solutions to these problems, proposing changes to the product and/or processes (redesign advice) and prioritizing these ideas based on the evaluation of the effects of these suggestions on the performance of the product. The effects of a number of post-design processes such as manufacture, assembly, schedulability, recyclability, disassembly, production, fabrication, reduced time-to-market [12, 41, 50, 62, 67, 70, 98, 128, 129, 136] on the product design have been studied in detail. Magrab [86] includes a comprehensive list of broad DFX areas that may be considered while designing the product. The elements of these broad DFX areas may be considered as overlapping evaluation criteria in the IP^{2}D^2 methodology.

A number of expert systems have been developed based on the DFX philosophy [24, 73, 83, 120]. An example is Wu and O'Grady's [144] research on the correlation between the concurrent engineering and design for assembly methodologies. The approach explained uses a variant of the Petri Nets concept to abstract and model the information needed for design for assembly during the design process. The effects of changes in the designs on the assembly process in terms of the cost and lead time, as the designer incorporates these changes in the design, serve as evaluation measures for incorporating the changes into the designs.

Thus, when neither the product designs nor the processes are fixed, or for the introduction of new products with new process technologies, there exists large scope for concurrency in development for which DFX tools and techniques can play a key role.

2.5 Design for Manufacturing

Design for manufacturing methodologies are used to improve a product's manufacturability. Three important issues dominate the discussion of design for manufacturing (DFM), also called design for manufacturability.

- Can the manufacturing process feasibly fabricate the specified product design?
- How much time does the manufacturing operation require?
- How much does the operation cost?

(For discussion, this body of work uses the term manufacturing to describe both fabrication and assembly, and includes design for assembly as part of design for manufacturing.)

DFM guidelines help a product development team design a product that is easy to manufacture, while other DFM approaches evaluate the manufacturability (feasibility, time, and cost) of a given product design with respect to a specific manufacturing process. Some manufacturability evaluation approaches give the product development team feedback on what aspects of the design make it infeasible or difficult to manufacture.

DFM compares a product's manufacturing requirements to existing manufacturing capabilities and measures the processing time and cost. DFM approaches can be used during the conceptual design and the detailed design steps. Generally, DFM approaches focus on the individual manufacturing operations, for example Boothroyd *et al.* [13], Bralla [15], and Kalpakjian [74].

In an attempt to increase the awareness of manufacturing considerations among designers, leading professional societies and some manufacturing firms have published a number of manufacturability guidelines for a variety of manufacturing processes [4, 11, 14, 102, 135]. Researchers have developed several different approaches to evaluate manufacturability of a given design. Existing approaches can be classified roughly as follows:

1. Direct or rule-based approaches [71, 73, 110] evaluate manufacturability from direct inspection of the design description; design characteristics that improve or degrade the manufacturability are represented as rules, which are applied to a given design in order to estimate its manufacturability. Most existing approaches are of this type. Direct approaches do not involve planning, estimation, or simulation of the manufacturing processes involved in the realization of the design.

2. Indirect or plan-based approaches [56, 58, 63, 66, 92] do a much more detailed analysis; they proceed by generating a manufacturing plan and examine the plan according to criteria such as cost and processing time. If there is more than one possible plan, then the most promising plan may be used for analyzing manufacturability, and some plan-based systems generate and evaluate multiple plans [52, 53]. The plan-based approach involves reasoning about the processes involved in the product's manufacture.

The direct approach appears to be more useful in domains such as near-net shape manufacturing, and less suitable for machined or electro-mechanical components, where interactions among manufacturing operations make it difficult to determine the manufacturability of a design directly from the design description. In order to calculate realistic manufacturability ratings for these latter cases, most of the rule-based approaches would require large sets of rules.

DFM has been very useful for reducing the unit manufacturing cost of many products, and successful product development processes require tools like DFM [115].

2.6 Design for Time-to-Market

Time-to-Market is the time from product conceptualization to market introduction. Short time-to-market means that a product reaches the market early, which in turn provides the corporate organization with the opportunity to enter the market during the growth phase of the product life-cycle, when the profit margins and potential for growth are higher and the product has a longer market life. Bralla [14] enumerates some of the advantages of achieving the objective as:

1. Reduced time-to-market implies reduced product development time, which in turn reduces the development costs since less funds are allocated to late engineering changes, rework and delays due to bureaucratic tarries.

2. The design related cost reductions are applied early in the development cycle.

3. As a consequence of being the first to introduce the product into the market, the company can be assured of an increased market share and the distribution and retail network confidence. This in turn increases the life-cycle of the product. Figure 2.5 compares the life cycles of products under normal and reduced development times.

4. Typical product delays that are associated with the introduction of a new product are reduced due to the reduced introduction time of the product into the market. These may include unforeseen changes in the market conditions necessitating design changes and changes in the development team members

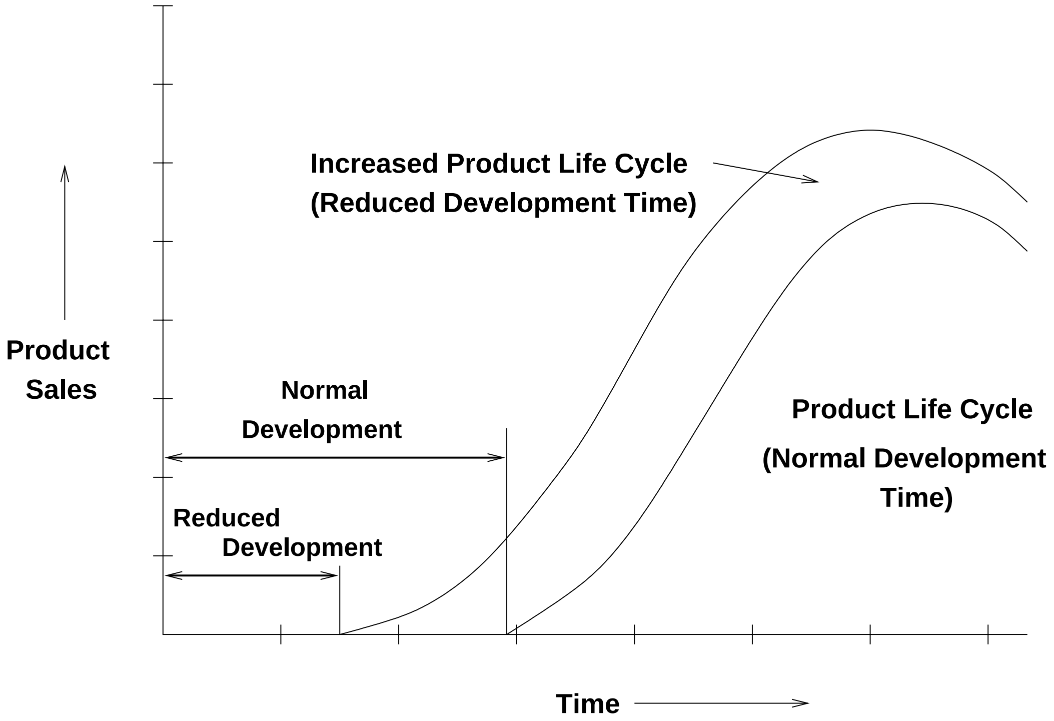

Figure 2.5: Product life-cycle curves with normal and faster time-to-market [88]

and the competence of these members.

Depending on the complexity, technical or otherwise, the process of introducing the new product may be very complex. Balachandra [5] identifies the following target areas for strategic initiatives by the product introduction project management in order that this complexity may be minimized as much as possible:

1. *Technology:* Technological information about new developments should be available to the product designers, manufacturing and marketing teams through free flow of such information and interactions between the teams.

2. *Market:* There should be a clear definition of the key attributes of the competitive strategy for the product within which all development should be attempted without aiming at perfection. The focus should be on incremen-

tal improvements in the product line.

3. *Organization:* All functions in the development should be treated with equal importance and there should be adequate communication [9] within and between different functional teams with suitable conflict resolution mechanisms.

4. *Vendors:* Vendors should be involved form the beginning in the product development.

Brookes and Backhouse [17] discuss the importance of evaluating the performance of product introduction and the difficulties faced within the task, as the product life cycle reduces and more products need to be introduced more often into the market. Some of the performance measures that can be used to evaluate the time-performance include time-to-market measures, average concept-to-launch time, time for each phase of development, average over-run and percent of products over-running, average time between product re-designs, product performance measures, product cost, technical performance, quality, return on sales, market share, design performance, manufacturing cost, manufacturability and testability. They provide case studies [17], evaluating product performance in different corporate organizations.

Govil [50] presents an approach to combining product design and production in an attempt to reducing the time-to-market of the product. The strategy, which is employed in the conceptual design stage of product design, has the following stages. First the designer inputs the product and process information into a tree structure. The system performs computations to find the production rate in order to be able to launch the product at the desired time-to-market. Next, the product and production system components that are critical to the desired production rate are identified. Alternatives for the product and production system that contribute towards improving the production rate are identified and the best set of alternatives is selected and output to the designer as improvements.

The system guides the user in creating a functionally decomposed representation of the product design based on Suh's axiomatic functional decomposition approach [124]. The user is allowed to assign materials for the parts of the product and choose compatible manufacturing processes. It also allows choice of assembly processes for various sub-assemblies leading to the final assembly. The system then calculates various processing system parameters, identifies the critical parameters and suggests improvements to the system and the processing logistics in order to reduce/eliminate the criticality of these resources.

2.7 Design Refinement

One of the important components of the concurrent engineering and DFX philosophies is design modification or design refinement. The aim is to modify the designs during the design phase itself before indulging in more expensive and resource intensive processes like manufacturing, while anticipating the problems that may be encountered in these stages. In order to achieve this, the redesign systems may act in a feed-forward capacity (design guidance) or in a feedback mode involving completed product designs. It is difficult to distinguish between the two mechanisms rigorously and a good redesign system combines the benefits of both.

2.7.1 Classification and Representation

Dixon *et al.* [39] propose a classification for mechanical engineering design based on a combination of the actual design, the designer and the design environment. They emphasize the need to represent functionality in product design. The objective of the classification is to enable the problems themselves to indicate possible solution processes. Bacon and Brown [3] propose using such classifications and repositories of devices derived therefrom to help discover the behavior of a device given some formal description of the structure. Hayes and Gaines[57] discuss a similar approach applying similar principles, termed *near misses*, to suggest redesigns in order to improve the manufacturability of the part. The suggestions are related to the part designs, the size and shape of the stock and the manufacturing equipment. Crawford and Anderson [29] propose a different architecture for preliminary mechanical design i.e. a stage where each component of the solution can be modeled by a number of parameters, variables, constraints and goals, using network representations and graph algorithms to model problems and plan solution procedures. Rinderlie [109] proposes product representative designs as combinations of three descriptions *function, form,* and *fabrication* which then formalizes the designer's task as specifying form in order to satisfy any constraints on product function and fabrication. The paper thus lays the foundation for representation of functionality of product designs and its relationship with the form and product fabrication. Mckay *et al.* [89] extend the functional modeling idea and describe the use of advanced product modeling techniques to represent product families without data redundancy. Two domain models are used to depict a model of a variant in a product family - a product variety data model and a framework-based product data model. In a similar work, Kimura and Suzuki [76] outline a framework for a more efficient product design system through the representation of design intent. Their emphasis is on separating design constraints as either *well-formed* or *ill-formed* and then adapting solution methods suitably.

Aldakhilallah and Ramesh [2] propose an architecture for a self-contained product design, process planning and control system which is well suited for a practicable and comprehensive concurrent engineering approach as explained above. They propose the decomposition of the designs into the constituent elemental features, which are prioritized by functionality and represented in a graph structure. The system suggests changes, as may be required, to the product designs for suitability to the manufacturing operations. During production, the system monitors the machine breakdowns, capacity changes and other anomalies and modifies the schedule depending on the severity of these anomalies.

Thus considerable efforts have been made to represent product information as cross-functional models which can be used to represent data required for various product development stages. These unified models are then evaluated and design modifications put forth. As mentioned before, it is very difficult to decouple concurrent engineering and DFX, and a system which aims to achieve better performance should aim to use a combination of these redesign schemas (in the advisory and evaluative capacity) during the development process.

2.7.2 Representing Design Advisory Rules

During the conceptual design stage the product development team hopes to take into consideration the entire development life cycle of the product while making design decisions. Hence, it is essential that as much knowledge as possible be made available to the design team about all aspects of the product life cycle.

Function plays a strategic role in the conceptual design stage. Deng *et al.* [38] describe strategies and methods for developing such a functional modeling design environment to guide designers during the conceptual design stage. They have developed a model based on the function, environment, behavior and structure of the product. Bardhan *et al.* [8] discuss an approach towards development of a multi-digit code system for each feature to be added to a design, that identifies its major attributes, feature essentials and unique identification. Based on this code system, the applicable design rules are checked for possible violation and modifications are suggested.

Govil's [50] advisory system guides the designer during the conceptual design phase by providing information about the post-design processes. Libraries of materials and properties are provided to the designer who is required to input the designs in the form of functional requirements (FR) of the parts and the design parameters (DP) corresponding to these functional requirements, using the axiomatic design theory [124]. Once the embodiments are finalized from these FRs, the designer is advised on the materials-processes options and the design-material-process model is created in the form of a tree structure. This tree structure may then be used for further analysis. Schmidt and Cagan [116] discuss the ability of *grammars* to generate a space of machine designs, providing a platform for a designer assistance tool. The central idea is to generate designs from a library using a grammar, the details of the representation of which are explained by Flasinski [45].

Often products exist over multiple domains and a product model defined in one domain needs to be valid and usable in another domain. Especially relevant to this situation is a product requiring construction of a prototype, since the materials and processes for the product and the prototype are normally different. Various researchers [49, 78, 146] explain problem solving architectures that cover multiple domains by managing information during the transformation between the two domains. Krishnan and Magrab [78] propose the use of interchangeable process specific entities to model the product in multiple domains. The use of such process specific entities is advantageous since the limitations of the manufacturing process are implicitly integrated into the design and the entities are representative of these limitations. Thus the geometric information and the manufacturability information are coupled into the product designs as they are created.

2.7.3 Product Redesign

Redesign may be included in the design process by incorporating design modifications after the detailed design phase. In this phase, the geometry of the parts is defined and the associated information needed to manufacture the part as dimensions, tolerances and related parameters is formalized. After this stage is complete, the designs may be evaluated in the context of post-design processes and relevant suggestions to improve product design performance during these post-design processes may be put forth.

In order for the system to evaluate designs efficiently and put forth viable and useful design modification options, it is imperative that the product models that act as input to these evaluation schemas be succinct and representative of all aspects of the product design. To attain this objective and to capture the functionality (what the design does) and design intent (justification of the underlying rationale behind design decisions), one well-researched approach is the creation of an *intermediate model* [60, 61, 133].

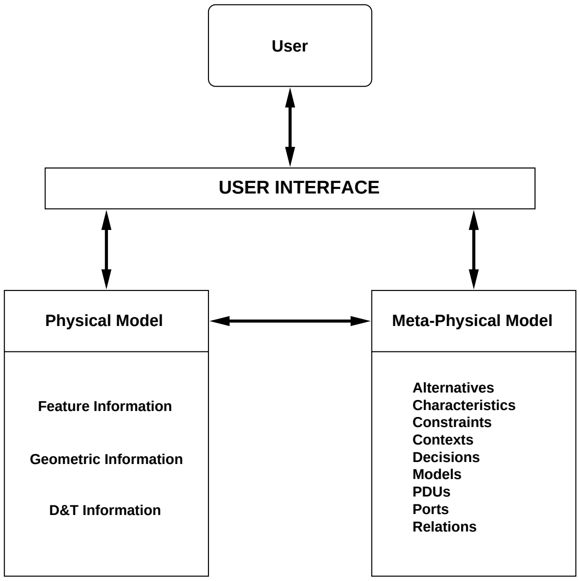

Figure 2.6: Product Model Schematic

Figure 2.6 shows the schematic for such a product model comprising two parts: a physical model and a meta-physical model along with their inter-relationships. The overall model is based on the creation of an abstract Product Definition Unit (PDU) as the basic element, which can represent anything from a system to a feature. The PDUs are given types, characteristics and links with other PDUs and physical entities. The PDU is merely a shell to encapsulate information. Features, which are natural collections of items that are used in a particular context, form the links between the physical and meta-physical entities. The information in the meta-physical realm pertains to nature, structure and behavior of objects in the physical realm.

DeMartino *et al.* [34] propose the creation of an *Intermediate Model* which is a multiple-view, feature-based representation of the product. To integrate design with other engineering processes, it is necessary to have a product model which is representative of the information of the design and the relevant engineering process and both must be able to share this information. The proposed intermediate model (called *Feature Kernel Model* [35, 36, 37, 48]) is a hybrid model comprised essentially of an adjacency graph, where each node corresponds to a feature face and arcs connecting the nodes represent the relationship between them in the form of geometric constraints or topological relationships. To make it easier to extract features [48] for use in post-design engineering processes, the shape features are represented [37] by their boundary entities and the topological adjacency relations between feature pairs. The linguistic properties of the features (semantic representation) [35] are modeled by a set of algebraic expressions representing the relations between the shapes. The model is capable of handling and representing multiple views of the designs. These multiple views adapt to the modeling requirements of other engineering processes. Consequently, the model serves as an intermediary between different processes supporting the philosophy of concurrent engineering in collaborative product design.

Hayes [55] presents a *Design Adviser* system for providing design evaluation to the designer concurrently during the design process. Part and processing system details are input to the system and it suggests changes to the design in order to be able to make the part. The modifications are aimed at reducing the manufacturing cost and keeping the alterations to the original designs to a minimum. Murayama *et al.* [93] take the geometric model of the part as the input and suggest modifications to the designs in order to improve the recyclability of the designs. Pnueli and Zussman [104] suggest an algorithmic approach for evaluating the end-of-life and recyclability and improving it through redesigns. The evaluation schema is based on the rules for optimal recyclability. The product designs, in all these approaches, are represented by AND/OR graphs [75, 134]. The system so formulated also attempts to automate the redesign optimization stage. Towards this end, some rules and guidelines for redesign generation based on *design for recycling* principles are ingrained into the system.

2.8 Manufacturing System Analysis

Manufacturing system analysis plays a very important role in the development of the new product. The designs are construed into physical products through the manufacturing system. Hence it is imperative that the performance of the manufacturing system be evaluated as a part of the product development process. The following sections aim to understand previous research in the field.

2.8.1 Models

A critical piece of data for estimating manufacturing cycle times is the processing time of each step required to manufacture the given product design. There exist many models and techniques for estimating processing times. Many of the DFM approaches include this activity. Estimating the processing time of a manufacturing step, given a detailed design, is usually different from estimating the processing time, given a conceptual design. For a detailed design, highly detailed process planning, manufacturing process simulation, or time estimation models can be employed [62, 92]. For existing products, the processing and setup times should be available from existing process plans. For a conceptual design, however, less detailed models must depend upon a more limited set of critical design information [50].

Most of the models presented here are descriptive (or evaluative) models that predict system behavior. Only the production scheduling models are optimization (or generative) models, since they attempt to find the best sequence of activities that minimizes the manufacturing cycle time. However none of these models attempts to select the best product design or manufacturing system configuration.

Types of Manufacturing Cycle Times

At this point two types of manufacturing cycle times are of interest to DFP approaches. First, consider a manufacturing system that will complete a large number of work orders of the new product. The size of these work orders may be fixed or have some variability. In this setting, the product development team will need to estimate the *average manufacturing cycle time* of these work orders. Second, consider a manufacturing system that will complete a small number of work orders of the new product. The product development team needs to determine the *total manufacturing cycle time* from the time the first work order starts to the time the last work order finishes. This will apply to an engineer-to-order or make-to-order manufacturer that wants to respond to a particular customer request and needs to estimate when the complete customer order (which may be one or more work orders) will be done. Note that this is similar to due date determination methodologies.

Models of Steady State Performance

This section describes types of models that can be used to estimate average manufacturing cycle time in a manufacturing system in steady-state. That is, the product mix, including the desired throughput of the new product, does not change, and the key resources of the manufacturing system are given and fixed. Most of the works referenced here consider the cycle time of a product with a simple routing, that is, the product requires a given sequence of operations.

Fixed lead times. In this model, completing a work order requires a fixed amount of time. This time does not depend upon the system's throughput or the available capacity. This model is the one used by material requirements planning (MRP) systems. A version of this model specifies a fixed lead time (based on past performance) for each workstation in the facility. This model is most appropriate for a facility where parts and assemblies are all very similar, and the product mix does not change very much.

Conveyor model. This model, described by Hopp and Spearman [65], estimates the manufacturing cycle time W for a job released to a CONWIP line that already has n jobs waiting to start processing. T_P is the minimum practical lead time, and r_P is the practical production rate:

$$W = \frac{n}{r_P} + T_p \tag{2.1}$$

This can be applied for estimating the manufacturing cycle time W of a job with n parts that requires processing on a line that processes one part at a time. If the line produces r_P parts per time unit, and each part takes T_P time units on average to move down the line, then W is approximated as follows:

This model is also useful for estimating the total manufacturing cycle time T of a set of s jobs. If W is the average manufacturing cycle time of a job and the release rate is one job every t time units, then

$$T = (s - 1)t + W \tag{2.2}$$

Queuing system models and approximations. Queuing models can represent a wide variety of manufacturing systems. Often, the model is a network of queues, where each node represents a different manufacturing resource or workstation. Given information about the probability distributions of job arrivals and job processing times at each node, one can determine the average time in system for a job. In general, the processing time distribution at one resource affects the interarrival time distribution at the resource that departing jobs visit next.

Papadopoulos *et al.* [103] review a large set of queuing system models for transfer lines, production lines, and flexible manufacturing systems. Many researchers have studied open queuing networks, like Buzacott and Shanthikumar [19], who present queuing network models for manufacturing systems and Connors *et al.* [26], who modeled semiconductor wafer fabrication facilities. Their goal is to analyze these facilities quickly by avoiding the effort and time needed to create and run simulation models. They present numerical results that show how the queuing network model yields results similar to those of a simulation model. Queuing network models are also the mathematical foundation of manufacturing system analysis software like rapid modeling [125]. Koo *et al.* [77] describe software that integrates a capacity planning model and queuing network approximations. They report that the approximations are reasonable when variability is moderate. However, few researchers have described how to apply this body of work to product design and manufacturability evaluation.

Cyclic production scheduling models. If the manufacturing system produces the same set of items repeatedly, then production scheduling models can be used to determine the period length and the time during each period that each manufacturing operation occurs. This information can be used to determine the manufacturing cycle time of each job. See, for example, Lee and Posner [82]. One can use cyclic production scheduling to model mass manufacturing systems that use hoists, robots, or other material handling machinery to move material between resources.

Discrete event simulation models. Simulation models can estimate manufacturing cycle time in almost any manufacturing system. There are a large number of simulation software packages available [126], and many good resources on simulation [7, 80]. By running multiple replications of a simulation model, one can estimate the mean manufacturing cycle time for each product. These are also useful for verifying analytical models.

Hybrid models. In some cases, simulation or queuing models may be most appropriate for only the more critical, heavily utilized resources, while fixed lead time models are sufficient for low utilization resources. A hybrid model uses different models for different workstations.

Models of Evolving Systems

Evolving systems refer to manufacturing systems where the product mix or the resource availability changes significantly over the time horizon of interest. This might include the desired production rate of the new product itself. Of course, it may be possible to divide the time horizon into two or more periods where the system reaches steady-state. In this case, the models mentioned above can be used for each time period. Alternatively, one can neglect the aspects of the system that are evolving and use a steady state model to approximate the system.

Production scheduling models. These models include, for all jobs to be processed by the manufacturing system, each scheduled activity on each key resource. Typically, there is a set of previously scheduled jobs for the existing products. The goal is to schedule the required jobs for the new product and determine when they will be completed. If the product development team is interested in the average manufacturing cycle time, W will be the average flow time of the jobs. If the product development team is interested in the total manufacturing cycle time, T will be the difference between the maximum completion time (of the relevant jobs) and the first release time.

Li and Cheng [84] present a version of the due date setting problem that has both new jobs that need due dates and old jobs that already have due dates. The objective is to schedule the jobs and assign due dates to minimize the total cost of delayed due dates and the maximum job tardiness cost. The approach is to solve a deterministic machine scheduling problem.

Discrete event simulation models. As stated above, these models can evaluate the performance of almost any manufacturing system. More sophisticated simulation softwares allow the user to model systems where production rates, resource availability, and other factors change during the simulation run.

Model Comparison

The models presented above vary widely. Some are quite simple, while others are very complex. This section compares these models on the following criteria: data requirements, computational effort, descriptive power, approximation accuracy, and ability to do sensitivity analysis.

Data requirements. The fixed lead times and conveyor models require the least data. Although some queuing network models require a large amount of data, the approximations need just a few statistics for each workstation. The production scheduling models require times for each activity. The simulation models require the most data, though the amount required depends on the level of detail involved. Of course, acquiring a small set of data may require summarizing a much larger set of data, but estimating and maintaining this smaller set requires less effort.

Computational effort. Computational time limits the number of runs that can be done and thus limits the amount of analysis that can be done. The fixed lead times and conveyor models require little computation. Again, some queuing network models require much computation, but the approximations are straightforward. Production scheduling models can require large computational effort, since some production scheduling problems are NP-complete. Heuristic approaches, however, may require less effort. Detailed simulations of large facilities can require hours to run, and one must perform multiple replications if the model includes random events.

Descriptive power. Some of these models can provide much more information than the average manufacturing cycle time. Queuing networks, production schedules, and simulation can provide information about resource utilization as well. Simulation is especially good for determining the range of system performance over different time periods, whereas queuing models usually provide estimates of means.

Approximation accuracy. The accuracy of any model depends upon the quality of the data provided. Timely, accurate data is essential. In general, the fixed lead time and conveyor models are the least accurate. Queuing network models vary widely. More sophisticated models will give more accurate estimates than the approximations. Simulation models, if used correctly, can provide accurate estimates and are useful for manufacturing systems with more complex interactions between resources or non-standard probability distributions.

Ability to do sensitivity analysis. Sensitivity analysis is important when the product development team wants to determine how changing the product design or manufacturing system will change the manufacturing cycle time. Because they use very little data, the fixed lead time and conveyor models are not useful. The production scheduling models have limited capabilities. The queuing network approximations are the most useful, since manufacturing cycle time is a function of the processing times and other parameters, and one can use derivatives to describe the sensitivity. The simulation models are less useful, though researchers are currently developing techniques like perturbation analysis for estimating gradients.

2.8.2 Capacity Analysis

Capacity analysis compares the manufacturing system's capacity to the product design requirements. The manufacturing system's capacity depends upon the time available at each required resource and the time already allocated to fabricating other products. The product design requirements depend upon the setup and processing time at each operation and the desired production rate. Capacity analysis can determine if sufficient capacity exists, estimate the maximum feasible production level, suggest alternative release dates, and suggest changes that would increase the manufacturing system capacity. Of course, the available capacity is not the same for each resource, since some resources are busier than others and sometimes there exist multiple, identical resources that can share the workload. Further, the capacity requirements are not the same for each resource since setup and processing times can vary greatly from one operation to the next. In addition, the available capacity may change from one time period to the next as the product mix changes.

Taylor *et al.* [132] use a capacity analysis model to determine the maximum production quantity that an electronics assembly facility can achieve. The analysis is done for a set of existing products and the detailed design of a new product. If the maximum production quantity is insufficient, the product design is changed so that its manufacture avoids a bottleneck resource, which increases the achievable production quantity to an acceptable level. This work does not estimate manufacturing cycle time.

Bermon *et al.* [10] present a capacity analysis model for a manufacturing line that produces multiple products. Their approach is not focused on product design but it is oriented towards decision support and quick analysis. They define available capacity as the number of operations that a piece of equipment can perform each day. Given information about the equipment available, the products, and the operations required, their approach allocates equipment capacity to satisfy the required throughput and availability constraints. They incorporate cycle time by constraining allocated capacity (utilization) to a level strictly below the available capacity. The difference is the contingency factor. Instead of setting this contingency factor in some ad hoc manner, as some manufacturers do, they describe a method to calculate a contingency factor for each tool group. The ideal contingency factor prevents the average queue time at that tool group from exceeding a predetermined multiple of the processing time. To model the relationship between utilization and queue time, their approach uses a queuing model approximation. Thus, their approach can determine if the manufacturing line has sufficient capacity to meet the required production and achieve reasonable manufacturing cycle times.

Many authors have described capacity planning methods that are part of traditional manufacturing planning and control systems [65, 139]. These methods determine how much, when, what type, and where a manufacturing system should add capacity to meet throughput requirements. Typical objectives include minimizing equipment costs, inventory, and cycle time. Different capacity planning models vary, and the more accurate methods require more data and more computational effort. These approaches do not consider how the product design affects the manufacturing system performance.

2.9 Estimating Manufacturing Cycle Time

Previous approaches estimate manufacturing cycle time either by modeling the steady-state performance of the manufacturing system or by scheduling or simulating manufacturing systems that are evolving as the product mix changes over time.

Previous work on manufacturability evaluation and partner selection for agile manufacturing developed two approaches for estimating manufacturing cycle time of microwave modules and flat mechanical products. Given a detailed product design, the variant approach [20, 21] first calculates Group Technology codes that concisely describe the product attributes. Then, this approach searches a set of existing products manufactured by potential partners and identifies the ones that have the most similar codes. The manufacturing cycle time of the most similar existing products gives the product development team an estimate of the new product's manufacturing cycle time.

The generative approach [62, 92], however, creates a set of feasible partner-specific process plans for the given product design and calculates the cycle time at each step in each plan. Given a production quantity, the approach calculates the required processing time for an order of that size and adds the processing time to historical averages for the setup and queue times at that resource in that manufacturing facility. The approach then sums these times over all the steps in each process plan, which gives the product development team an opportunity to see how choosing different partners affects the manufacturing cycle time. This approach does not consider the available capacity that the manufacturing resources have or adjust the queue times as utilization increases.

Singh [119] calculates the time at a manufacturing operation as the sum of the setup time and the run time (the part processing time multiplied by the lot size). This approach ignores any time due to queuing or moving. Seepersad *et al.* [118] present a manufacturing cycle time analysis of a heat-exchanger tube manufacturing facility and an approach for optimal design of these tubes using a product platform based approach.

Govil [50] assumes that the cycle time at each manufacturing operation is one time period. The lead time for purchased parts may be multiple periods. This approach uses the assembly structure described earlier to create a tree of purchasing and manufacturing operations, and the manufacturing cycle time is the length of the longest path through this tree.

Meyer *et al.* [90] describe an approach for comparing microwave module designs. Each different design uses a different set of electronic components. The approach generates process plans that are feasible with respect to the characteristics of the selected components. They evaluate each design and process plan based on the cost, the system reliability, and the maximum lead time required to procure any of the selected components.

Veeramani *et al.* [137, 138] describe a system that allows a manufacturer to respond quickly to requests for quotation (RFQs). They apply the approach to companies that sell modified versions of standard products that have complex subassemblies (like overhead cranes). Based on customer specifications for product performance, the system generates a product configuration, a three-dimensional solid model, a price quotation, a delivery schedule, the bill of materials, and a list of potential design and manufacturing problems. The system verifies whether the design can be feasibly manufactured by the shop. The authors claim that, to generate the delivery schedule for that order, the system uses data about shop floor status, current orders, and alternative process plans to determine the time needed to produce the new order. Although no details are given, it appears that the system does some shop floor scheduling to determine the completion date.

Elhafsi and Rolland [43] study a make-to-order manufacturing system and build a model that can determine the delivery date of a single customer order. The model takes into account the production lines' existing workloads and allocates portions of the order to different lines to minimize the cost and estimate the expected delivery date. Each line is modeled as a single-server queuing system.

The U.S. Air Force is developing the Simulation Assessment Validation Environment (SAVE), which integrates a set of virtual manufacturing tools. The SAVE program will help product development teams develop affordable weapon systems (like fighter aircraft) by giving them the ability to evaluate cost, manufacturing cycle time, inventory levels, rework, and other manufacturing metrics. The SAVE approach uses detailed factory simulation models to estimate manufacturing cycle time.

Soundar and Bao [122] describe a plan to address the question of determining how the product design affects the manufacturing system. They propose using mathematical and simulation models to estimate a variety of different performance measures, including manufacturing cycle time. Though the approach is quite general, the paper does not describe any examples or results.

2.10 Process Yield, Manufacturing Cycle Time and Throughput

The throughput for a manufacturing system is the rate at which parts are processed in the system. An associated aim, in addition to estimating manufacturing cycle time, is studying the relationship between the manufacturing cycle time and throughput for a manufacturing system subject to process drift.

As a station processes a batch of parts, some parts will become defective due to the variability of the process. These bad parts must be detected and discarded.

In general, yield is the ratio of the number of good parts produced to the number of parts processed. This ratio is between 0 and 1. The bad parts may be detected at the current station or at an inspection station that the parts visit at a later step in the processing sequence.

In particular, the case where some bad parts are detected and discarded at the current station while other bad parts are detected and discarded at the next inspection station is important. Some types of flaws are obvious and can be detected immediately, while others require careful examination by trained inspectors using special equipment or procedures. This research uses the term *scrap yield* to describe the fraction of parts that do not have obvious flaws. These parts continue to the next step. However, some of these parts have undetected flaws (which will be found at the inspection station). Moreover, the size of the fraction with undetected flaws depends upon whether the process is operating within its specifications. *Normal yield* occurs when this is the case. The *reduced yield* (which is lower than the normal yield) occurs when the process behavior has drifted beyond its specifications. It must be noted that this assumption of a two state yield, as explained here, is a binary simplification for the process drift which is in reality a continuously decreasing function.

Process drift is a common occurrence in many manufacturing processes where machines become dirty (leading to more contamination) or other aspects change, leading to degraded performance. Statistical process control tracks process quality to determine when the process has gone out of control (has drifted beyond its specifications). The time that the process remains out of control depends upon how long it takes a batch of parts to move from that station to the inspection station. This time is called the detection time. When the batch is inspected, the drift is noticed (through a statistical process control method), the process is fixed and the process resumes operating within its specifications. The fraction of parts with undetected flaws now equals the normal yield.

Clearly, a larger detection time implies that the process will operate out of control (at the reduced yield) for a longer period of time, which reduces the throughput (the number of good parts produced). The detection time depends on the position of the inspection station in the processing sequence and the manufacturing cycle times at the stations that follow the process that is out of control.

These issues are important considerations in design for production. First, the product design may affect the yield of a process, since some designs are easier to make than others (eg. a product with tight tolerances may have a lower yield on a machining step than a product with broader tolerances). Second, the product design may affect the processing time and manufacturing cycle time at various stations. Thus, changes to the design may increase (or decrease) the detection time, which affects the throughput. Since design for production seeks to understand how design changes affect the performance of the manufacturing system, it is important to have models that can estimate manufacturing cycle time, yield, and throughput, which are important performance measures.

Srinivasan *et al.* [123] enumerate benefits of reducing cycle time towards improving system yield for semiconductor manufacture. The paper presents graphs relating the process yields to deviation of cycle time from its nominal value along with a simulation model to quantify the relationship. Narhari and Khan [95] analyze inspection results as reentrant flows into the queuing network. They also address the problem of alternate ways of locating inspection stations in a processing sequence. This gains relevance from the fact that increasing the number of inspection stations in a processing sequence increases the overall cycle time but leads to early detection of defects. Cunningham and Shanthikumar [30] present analyses of effects of reducing cycle time on improving die yield of semiconductor wafers. They present two conjectures on how reducing cycle time improves yield, an informational conjecture which states that the completed batches can be studied for defects and improved and a physical conjecture which states that a reduced cycle time means lower contamination of completed batches.

Since detecting a process drift depends on the time that elapses before the first defective product arrives at the closest inspection station, the cycle times for the resources in the processing sequence for the product from the resource where the drift occurs and the next inspection station contribute to the detection time for the process drift. This research focuses on understanding the relationship between the manufacturing cycle time and the defect detection time. Since process drift affects not only the product batch during whose processing it actually occurs, but also every batch of every product thence until detection, there are two implications:

1. The yield losses at various resources in the processing sequence result in a decrease in the batch size along the processing sequence. Chapter 3 provides relations to calculate the initial release rate based on the desired throughput and given input batch size. Due to the decreased batch size, the final throughput is much lower than the release rate.

2. Inspection stations increase the processing time for the product without adding value to it. Hence, optimal placement of these stations contributes towards decreasing the total time spent by the product in the system. These may include placing the inspection stations near *high-risk* processing stations. The models presented here for manufacturing systems with process

drift may be extended to identify such processing stations and figuring out inspection station positioning.

2.11 Printed Circuit Boards and Embedded Passives

In applications such as consumer and industrial electronic products, a printed circuit board, PCB (or printed wiring board, PWB) forms the backbone of the device. The PCB substrate supports the discrete components that form part of the circuit along with the wiring requirements for these components. The components may be passive devices (such as resistors and capacitors) or active devices (such as diodes, integrated circuits, and transistors). These components may be mounted on one side of the substrate or on both sides depending on the circuit's requirements and the size of the board. Usually, the substrate is constructed by laminating copper to one or more surfaces of a sheet of plastic reinforced by paper or glass fiber. Single layer, single sided boards have only one circuit layer. Single layer, double sided boards have two circuit layers, one on each side of the board. Multilayer boards have three or more circuit layers made by bonding (or laminating) layers of patterned, pre-etched, undrilled copper-clad laminate together. Layer interconnections are then made by drilling and plating through holes in the non-conducting plastic.

As circuits get more complicated, they require more discrete passives. Coupled with progressive reduction in the size of the electronic device, which in turn means reduced "real estate" atop the PCB, this intensifies the need for shrinking the size of the passive components and developing alternative technologies to accommodate the large numbers needed for device functionality. *Embedded* (or *integrated*) *passive components*, which are part of or buried in the PCB substrate and are fabricated along with the substrate [114], may be one way of realizing these goals.

The *National Electronics Manufacturing Initiative (NEMI)* defines embedded passives as functional elements either buried or incorporated on the surface of an interconnecting substrate. For more information see, for example, [44, 105, 107, 108, 111, 147].

2.11.1 Definitions

This subsection explains some terminology associated with embedding passives components in the substrate of a PCB.

Passive Device is simply a single passive element (capacitor, resistor or inductor) in a leaded or SMT (surface mount technology) case [87].

Discrete Passives are the simplest form of passive devices, composed of individual parts with two I/Os per unit or element.

Panelization is the process of combining many smaller printed circuit boards into a large one for processing.

Embedded (or Integral) Passives are passive devices that are buried into the substrate material of the PCB itself. The passive elements are then considered to be an integral part of the substrate.

2.11.2 Pros and Cons of Embedding

There are several potential advantages and some shortcomings to embedding passives into the PCB substrate, studied in literature [44, 105, 107, 111, 108, 147]. These include:

1. Increase in the number of embedded components, particularly resistors, decreases the total board area.

2. Due to the embedding of some resistors and bypass capacitors into the board itself, the wiring requirements in the form of tracks on the outer layer of the board, for connecting the components to one another are reduced. However, due to the reduction in size of the board, the wiring density may increase since less area is now available for laying the tracks. A greater number of boards can now be fabricated from the same panel due to this reduction in board area. Capacitors can be embedded directly under the active component they support, thereby reducing the number of layers and interconnecting vias.

3. Since a number of passives are now embedded into the PCB itself, the number of passives to be surface-mounted reduces resulting in reduced assembly time and costs. Fallouts of this reduction are an increase in the assembly level yield and reduction in the rework needed at that stage. Thus, increasing the number of passives embedded into the substrate increases the density of the circuit but saves space on the surface of the substrate decreasing the product weight.

4. Area processes tend to make all components bad when they fail as opposed to assembly which can be corrected for the failed component. It is therefore advantageous to include the passive layers near the bottom of the substrate

(during initial manufacturing cycle) so that the low yield steps are included early and defective ones discarded/reworked early. This somewhat constrains the processing sequence for a PCB with embedded passive components. Due to an increase in the board complexity, the yield could decrease as also the board throughput. On the whole, embedding passive components could reduce engineering and manufacturing flexibility.

5. Using embedded passives allows for increased active circuit density, improved electrical performance and improved reliability. A capacitor dielectric placed between the power and ground plane would lower noise and provide blocking capacitors for filtering applications. This would simplify board construction, thereby reducing costs and lowering parasitic inductance and cross-talk. Additionally, electrical connections are shortened and electrical properties of each device are improved through additional termination and filtering opportunities. Product quality is improved due to reduction in incorrectly soldered passive devices to the PCB and reliability is increased.

6. Increase in cost due to embedding passives remains constant upto a certain number of passive components (since it is easy to form one as it is to form many in an area process). Thus, embedded passives become cost effective when a large number of components within a system can be fabricated in a single run. Cost savings are associated with eliminating discrete resistors, rework reduction, board densification and/or reduction, more streamlined assembly process. The cost of the device may be further reduced by introducing manufacturing automation in making PCBs with embedded passives.

In conclusion, though embedding requires additional processing steps, removing discrete passive components can result in streamlined assembly, less rework and greater overall circuit design flexibility.

2.11.3 Buried Passive Parameters

There are different techniques for embedding passive components into the PCB substrate. One of the technologies widely employed is known as *Ohmega-Ply*©[1]. The following explanation of the technology is described by Signer [117] and the *Ohmega-Ply*© technical manual [100].

The advantage of *Ohmega-Ply*© is that the construction is compatible with existing printed board processing and the material system has fewer dimensional constraints. The material system consists of copper foil to one side of which a metal alloy film is applied. This is then laminated to a polymer substrate such that the alloy film contacts the substrate. The materials come with resistive sheet values of different ohm-per-square and cover a large number of applications.

Now, during processing to create the resistor, the etching is controlled in different passes so that in certain portions, both the foil and the resistive metal alloy film are etched while in others, the foil but not the resistive film is etched. Two foil areas connected by only the resistive material form a resistor connecting the two areas.

Figure 2.7 depicts the cross-section of a PCB with an embedded resistor and capacitor.

The resistance, R, of a buried resistor is given by

[1] Ohmega-Ply is a registered trademark of Ohmega Technologies, Culver City, California

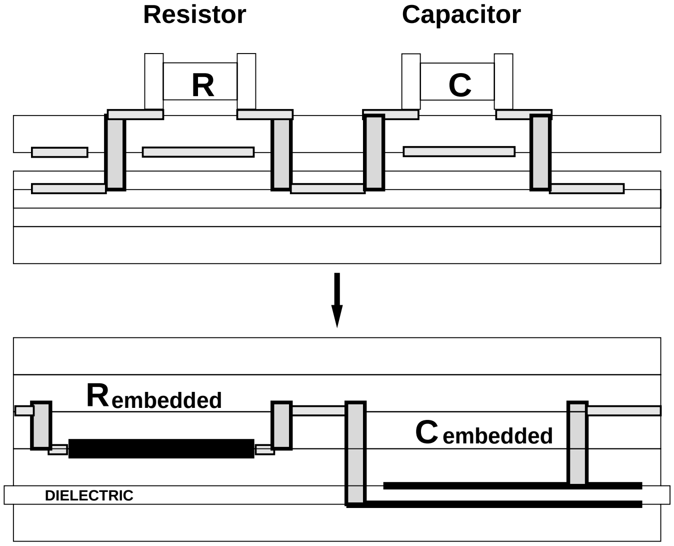

Figure 2.7: Cross-section showing embedded passives [87]

$$R = \rho_s \, N \tag{2.3}$$

where

$$\begin{aligned}
\rho_s &= \text{sheet resistance of material, } \Omega/\text{square} \\
N &= \text{number of squares} \\
&= \frac{L}{W} \\
L &= \text{length of resistor element} \\
W &= \text{width of resistor element}
\end{aligned}$$

Figure 2.8 shows these parameters. The shaded squares correspond to the number of squares in the embedded resistor. The resistance of the resistor shown in the figure is $5\rho_s \, \Omega$.

An advantage of this system is that all manufactured units fall within accept-

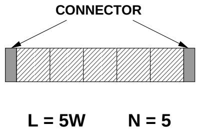

Figure 2.8: Resistance of an embedded resistor

able tolerance limits of the PCB production processes employed. The limitations of the technique are that since it is a polymer based technology, the units cannot be used in high temperature applications and the initial resistivity tolerances are tighter since the parameter depends on the material sheet resistance tolerance and element dimensions.

2.12 Embedded and Discrete Passive Components

There are many implications of deciding the quantity and type of passives to be embedded to cost and manufacturing system performance. Some of the tradeoffs are related to the cost of manufacture of the substrate, the cost of surface mounting discrete components and overall area of the passives to be embedded and in effect the PCB itself. Also, reducing the number of discretes has the additional benefit of reducing the assembly time needed for the PCB, thereby reducing the manufacturing cycle time for the assembly station.

Embedding discrete passives into the PCB changes the manufacturing process for the PCB. The advantages and changes in the design, when the decision to embed discretes (the first time a passive component is embedded) is taken by the PCB manufacturer, are essentially on account of layers with embedded passives to be added to the PCB. These changes are listed in Subsection 2.12.1. Embedding additional passives into the substrate may not necessarily add layers to the PCB.

Subsection 2.12.2 enumerates the design changes that result.

2.12.1 For the Initial Embedded Passive

When any of the passive components in the design of the PCB are embedded in the substrate or each time an additional layer with embedded passives needs to be added to the PCB, the following points need to be kept under consideration while defining the design parameters for the PCB:

1. The overall drill time increases due to the addition of each new layer to the board. This is especially important for the case of laser drilling where an increase in thickness drastically increases the required drill time. It is of less importance for mechanical drilling. However, in case of mechanical drilling, the tool wear is also likely to increase due to the increase in thickness of the board. This in turn means that the tool needs to be changed more often meaning more tool change (and setup) time and costs for the same panel.

2. When the first passive is embedded, additional steps enumerated in Subsection 4.1.3 are added to the process plan for the PCB. Thus, the time taken for the these operations some of which may be directly proportional to the number of layer pairs increases.

3. Addition of a layer from embedding a passive device increases the time required for the kitting operation as well as the layup and lamination operations.

4. Embedding the passive devices imposes restrictions on the assembly sequence of the PCB. Since errors in the layers having embedded passives are not easily

rectifiable, it is necessary to complete and test such layers before making assemblies or sub-assemblies with these layers. Thus they influence the overall processing sequence for the PCB.

5. It may be possible to eliminate the surface mounted components or the pin-through hole components on one side of a double sided PCB through embedding passives into the substrate. If the present board has discrete passives on both sides of the board, with embedding some passives, there is a saving of space on the surface of the PCB. It may be possible to combine the discretes on both sides onto just one side of the PCB. There would be a tremendous saving on time as a result, since now the PCB needs to pass through the assembly process and hence the series of stations only once.

2.12.2 Changes Effective Each Time a Passive is Embedded

Each time an additional discrete component is embedded, design and processing modifications occur for the PCB. These, though not as dramatic as the ones listed above, nevertheless need to be kept under consideration while analyzing the PCB. Such modifications include:

1. Reduced number of components that are to be surface mounted or through-hole mounted. Lower number of components to be assembled on the PCB results in reduced assembly time for the PCB.

2. Reduced number of holes to be drilled (corresponding to the passive which has been embedded).

3. In conjunction, lower drilling and deburring process times.

4. If the passive embedded is a build capacitor requiring the addition of a new layer each time it is added, all steps required (see Subsection 4.1.3) for the addition of a passive layer need to be performed.

5. Due to a possible increase in board area due to the addition of each new passive, the time taken for printing the artwork on the board is likely to increase.

6. Embedding each passive has potential to reduce the surface area of the board meaning more boards can be fabricated from a panel. This in turn means that the setup time at each station per board is reduced since a setup is done for a panel which now makes more boards.

7. Each time a passive is embedded without the addition of a new layer, some of the area of the board is occupied by the passive with the result that this area is blocked and cannot be used for wiring i.e. laying tracks to connect the various components that the PCB supports. Section 4.3.1 presents a model for the number of layers in a PCB with embedded passives.

Please refer to [22] for more details.

2.13 Summary

This chapter discussed research into understanding the *concurrent engineering* and *design for X* concepts. It included some previous work done in manufacturing system analysis and design and some models developed for the purpose. Some sections were devoted to explaining the importance of design and redesign to the product development process along with the need to minimize redesign.

The chapter surveyed some of the techniques for facilitating redesign. These techniques include creating intermediate cross-domain product representations capturing functional information and formulating optimization problems for evaluating design performance. It also reported some systems developed that employ these techniques. These ideas and systems utilize information from various post-design processes such as manufacturing and assembly.

Two sections of the chapter were devoted to explaining the concept of process drift and its importance to the relationship between the manufacturing cycle time for a product, process yield and product throughput, and embedding passive components into the substrate of a printed circuit board respectively. The section on embedded passives reviewed the projections and expectations of the electronics community from the process of embedding passives and the expected benefits. These topics will be studied in more detail in this dissertation.

Thus, this chapter acknowledges that there is a plethora of issues in product development, in general and DFX and product profitability assessment, in particular. There is a definite need to address the issues related to manufacturing system performance for a system processing a set of product designs. A good deal of issues have already been studied by different researchers and models and methodologies proposed. There is yet significant potential to develop a more complete and comprehensive approach to design for production.

The following chapters detail manufacturing system models for different production situations, tools and techniques to understand the design-production relationship and contributions towards improving the product development process and models to quantify the economic impact of reducing manufacturing cycle time.

Chapter 3

Manufacturing System Model

In general, DFP refers to methods that study the impact of a product design on the manufacturing system performance and provide means to estimate this manufacturing system performance. For estimating the manufacturing system performance, this chapter models the manufacturing system as a queuing network. This model is then used to analyze the manufacturing system processing different product sets.

Section 3.1 explains the model in detail. Section 3.2 explains the working of the DFP tool developed based on this model. Section 3.3 presents an application of the tool to a product domain: microwave modules, along with sample results. Section 3.4 summarizes the chapter.

The models developed in Section 3.1 have been extensively used by Wei and Thornton [140, 141] for production system performance evaluation of Boeing's aircraft tube manufacturing plant.

3.1 Manufacturing System Model Explanation

This section explains the mathematical formulation that is used to model the manufacturing system into which the product set is introduced for production.

Assumptions. This manufacturing system model estimates the average manufacturing cycle time under the following conditions:

- the product mix and the resource availability are fixed,
- the manufacturing system has reached a steady state,
- no job visits any station more than once (i.e. there is no re-entrant flow),
- the yield at each processing station in the system does not change.

Steady state may be defined as a state where the demand does not change over the given time period and the manufacturing system processes a large number of batches.

3.1.1 Data Requirements

The manufacturing system model requires the following data:

- For each workstation:
  - the number of resources available,
  - the mean time to failure for a resource,and
  - the mean time to repair the resource.
- For each existing product and the new product:

  - the job size (number of parts),
  - the desired throughput (number of parts per hour of factory operation), and
  - the sequence of workstations that each job must visit.

- For each product-resource combination:
  - the mean setup time (per job) at each workstation and its variance,
  - the mean processing time (per part) at each workstation and its variance, and
  - the yield at each workstation that a job must visit (the ratio of good parts produced to parts that undergo processing).

Often, even in make to stock kind of situations, demand varies with time. The models presented in this chapter address the issues of varying demand by defining *production horizons*. The demand in each production horizon is assumed to be at a constant rate. The demand can, however, vary from one production horizon to another. Thus, the assumption of steady state demand now holds true for each production horizon. The release rate for each production horizon is calculated using the demand for that production horizon. Note that, as a result, the average manufacturing cycle time for the product may be different in each production horizon.

Mathematical Notation

SCV = squared coefficient of variation

I = the set of all products (existing and new)

T_i = desired throughput of product i (parts per hour)

B_i = job size of product i at release

c_i^r = SCV of job interarrival times for product i

J = the set of all stations

n_j = the number of resources at station j

m_j^f = mean time to failure for a resource at station j

m_j^r = mean time to repair for a resource at station j

R_i = the sequence of stations that product i must visit

R_{ij} = the subsequence that precedes station j

t_{ij} = mean part process time of product i at station j

c_{ij}^t = SCV of the part process time

s_{ij} = mean job setup time of product i at station j

c_{ij}^s = SCV of the setup time

y_{ij} = yield of product i at station j

Y_{ij} = cumulative yield of product i through R_{ij}

Y_i = cumulative yield of product i through R_i

x_i = release rate of product i (jobs per hour)

A_j = availability of a resource at station j

V_j = the set of products that visit station j

t^+_{ij} = total process time of product i at station j

c^+_{ij} = SCV of the total process time

t^+_j = aggregate process time at station j

c^+_j = SCV of the aggregate process time

$t^\star_j$ = modified aggregate process time at station j

$c^\star_j$ = SCV of the modified aggregate process time

Aggregation. Aggregation calculates, for each product, the processing time of each job at each station. It also calculates, for each station, the average processing time, weighted by each product's arrival rate. Finally, it modifies the aggregate processing times by adjusting for the resource availability.

The cumulative yield is the product of the operation yields. Note that in the model, the yield for a product at a station y_{ij} will have a value between zero and one. This is also referred to as the pass fraction for the station. y_{ij} will have a value less than one for each processing station where defective parts are discarded. If defective parts are identified at a test or inspection station that discards them, the processing station will have a yield of one and the test (inspection) station will have a yield less than one.

$$Y_{ij} = \prod_{k \in R_{ij}} y_{ik} \tag{3.1}$$

$$Y_i = \prod_{k \in R_i} y_{ik} \tag{3.2}$$

$$x_i = \frac{T_i}{(B_i Y_i)} \tag{3.3}$$

$$A_j = \frac{m_j^f}{m_j^f + m_j^r} \tag{3.4}$$

$$V_j = \{i \in I : j \in R_i\} \tag{3.5}$$

The time spent by a job at station j is the sum of the part processing time and the setup time. The job size depends on the cumulative yield of the preceding operations.

$$t_{ij}^{+} = B_i Y_{ij} t_{ij} + s_{ij} \tag{3.6}$$

$$(t_{ij}^{+})^2 c_{ij}^{+} = B_i Y_{ij} t_{ij}^2 c_{ij}^t + s_{ij}^2 c_{ij}^s \tag{3.7}$$

Equation 3.7, which is used to calculate c_{ij}^{+}, holds because the variance of the total process time is the sum of the variance of the part process times and the variance of the job setup time. The aggregate process time of jobs at station j is the weighted average of all the jobs that visit station j. Each product is weighted by its release rate, as shown in Equation 3.8. Equation 3.9 calculates the mean of the squared aggregate process time, which can be used to determine c_j^{+}, the SCV.

$$t_j^{+} = \frac{\sum_{i \in V_j} x_i t_{ij}^{+}}{\sum_{i \in V_j} x_i} \tag{3.8}$$

$$(t_j^{+})^2 (c_j^{+} + 1) = \frac{\sum_{i \in V_j} x_i (t_{ij}^{+})^2 (c_{ij}^{+} + 1)}{\sum_{i \in V_j} x_i} \tag{3.9}$$

Equations 3.10 and 3.11 modify the mean and SCV for the process times by including the effects of resource availability.

$$t_j^\star = \frac{t_j^+}{A_j} \tag{3.10}$$

$$c_j^\star = c_j^+ + 2A_j(1-A_j)\frac{m_j^r}{t_j^+} \tag{3.11}$$

Arrival and Departure Processes. The arrival process at each station depends upon the products that visit the station. Some products are released directly to the station, while others arrive from other stations. The departure process depends upon the arrival process and the service process.

V_{0j} = the set of products that visit station j first

V_{hj} = the set of products that visit station h immediately before j

λ_j = total job arrival rate at station j

λ_{hj} = arrival rate at station j of jobs from station h

q_{hj} = proportion of jobs from station h that next visit station j

c_j^a = SCV of interarrival times at station j

c_j^d = SCV of interdeparture times at station j

$$\lambda_j = \sum_{i \in V_j} x_i \tag{3.12}$$

$$\lambda_{hj} = \sum_{i \in V_{hj}} x_i \tag{3.13}$$

$$q_{hj} = \frac{\lambda_{hj}}{\lambda_h} \tag{3.14}$$

Equations 3.15 and 3.16 estimate the SCVs for the departure and arrival processes.

$$
\begin{aligned}
c_j^d &= 1 + \frac{u_j^2}{\sqrt{n_j}}(c_j^\star - 1) + (1 - u_j^2)(c_j^a - 1) && (3.15) \\
c_j^a &= \sum_{h \in J}((c_h^d - 1)q_{hj} + 1)\frac{\lambda_{hj}}{\lambda_j} + \sum_{i \in V_{0j}} c_i^r \frac{x_i}{\lambda_j} && (3.16)
\end{aligned}
$$

Solving the above set of equations yields the complete set of c_j^a and c_j^d for all stations.

If the shop is a flow shop, and all products visit the same sequence of stations, then the stations may be renumbered: $1, 2, ..., J$. $V_j = I$ and $V_{j-1,j} = I$ for all stations, and the last equation can be simplified as follows:

$$
\begin{aligned}
c_1^a &= \frac{\sum_{i \in I} c_i^r x_i}{\sum_{i \in I} x_i} && (3.17) \\
c_j^a &= c_{j-1}^d, 2 \leq j \leq J && (3.18)
\end{aligned}
$$

Performance Measures. The performance measures of interest are the average utilization of resources and the manufacturing cycle time. The average cycle time of a job depends upon the cycle time at each station it visits.

u_j = the average resource utilization at station j

$CT_j^\star$ = the average cycle time at station j

CT_i = the average cycle time of jobs of product i

$$\begin{aligned}
u_j &= \frac{t_j^\star}{n_j} \sum_{i \in V_j} x_i && (3.19) \\
CT_j^\star &= \frac{1}{2}(c_j^a + c_j^\star) \frac{u_j^{(\sqrt{2n_j+2}-1)}}{n_j(1-u_j)} t_j^\star + t_j^\star && (3.20) \\
CT_i &= \sum_{j \in R_i} CT_j^\star && (3.21)
\end{aligned}$$

Sensitivity. This model can indicate how the manufacturing cycle time of the new product is sensitive to its part processing time at any station. In the general case, calculating the derivative $\frac{d\ CT_j^\star}{d\ t_{ij}}$ is feasible but complex due to the equations that describe the arrival and departure processes.

$$\begin{aligned}
M_j &= \text{estimate for } \frac{d\ CT_j}{d\ t_j^\star} \\
S_{ij} &= \text{estimate for } \frac{d\ CT_j^\star}{d\ t_{ij}}
\end{aligned}$$

$$\begin{aligned}
M_j &= \frac{CT_j^\star}{t_j^\star} && (3.22) \\
S_{ij} &= \frac{CT_j^\star}{t_j^\star} \frac{x_i B_i Y_{ij}}{A_j \lambda_j} && (3.23) \\
&= M_j \frac{x_i B_i Y_{ij}}{A_j \lambda_j} && (3.24)
\end{aligned}$$

Discussion. The queuing network approximations used here offer some advantages and also have limitations [19]. Compared to simulation models or more sophisticated queuing network analysis techniques, these approximations are less accurate, especially for very complex systems, and cannot provide the same range of performance measures. However, they require less data and less computational effort than the simulation models and other analysis techniques. Therefore, they are more appropriate for situations where a decision-maker needs to compare many scenarios quickly.

3.2 DFP Tool

This research developed a DFP tool to help the product development team better evaluate product designs, based on the analytical model detailed in Section 3.1. This tool demonstrates the proposed evaluation mechanism based on manufacturing system performance. This section explains the working of the tool. The tool can be customized as needed for a specific product domain. Section 3.3 explains the application of the tool to the domain of microwave modules.

The aim of the tool is to help the designer make good design decisions based on knowledge of the performance of the design in the production phase. Towards this end, the design and production teams create a database of the existing production facility:

Factory Database : This is the database that needs to be created by the production team and be made available to the program before running the subroutines. It provides information about the resources available in the manufacturing system. For each of the resources, the data needed is:

1. name of machine,
2. mean set up time,
3. variance in set up time,
4. number of machines,
5. mean time to failure, and

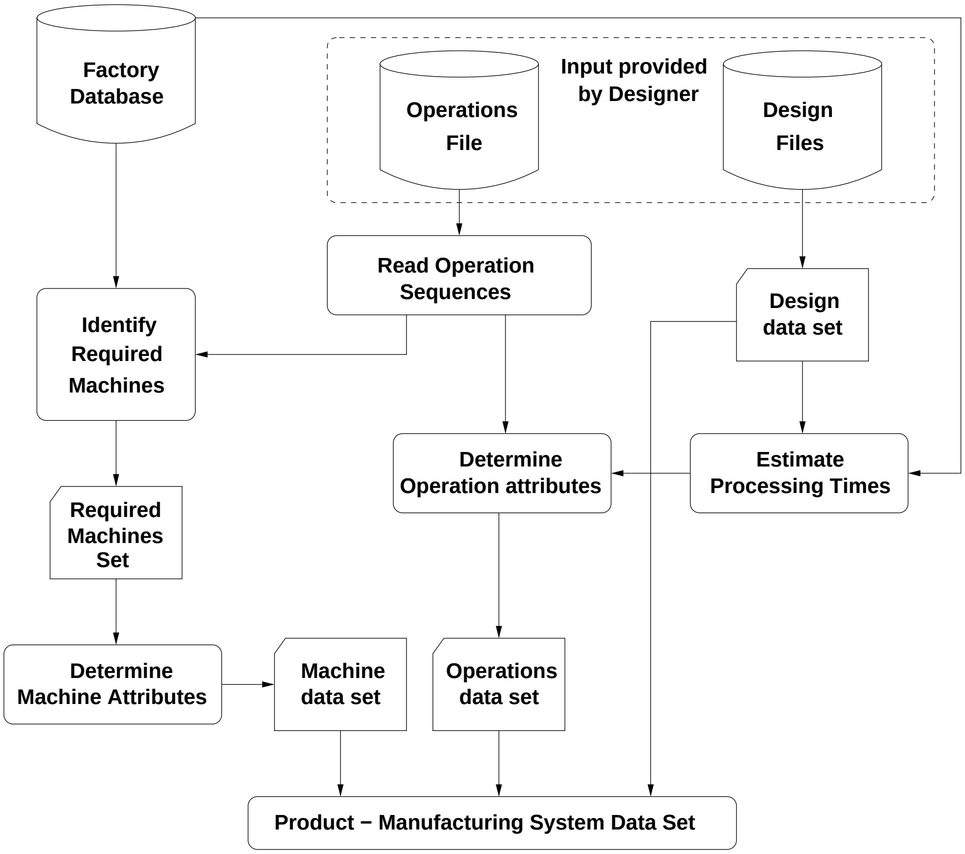
Factory Database
Operations File
Input provided by Designer
Design Files
Read Operation Sequences
Design data set
Identify Required Machines
Determine Operation attributes
Estimate Processing Times
Required Machines Set
Determine Machine Attributes
Machine data set
Operations data set
Product − Manufacturing System Data Set

Figure 3.1: Block diagram for DFP tool inputs

6. mean time to repair.

It is expected that the designer use this database as a guide, thereby avoiding making changes to the production system, which is often an expensive proposition. The user input to the tool is in the form of two files:

Operations File : This file contains a list of the products (in the product set) being manufactured. It should provide the following product information:

1. name of the product,

2. job size for the product (number of parts),

3. length of production horizon,
4. throughput requirements (parts per time unit),
5. operations sequence and associated operations parameters, and
6. SCV for the job interarrival times.

Design Input File : This is, in actuality, a set of files, one corresponding to each product in the product set. Each design file should list the relevant design information for the product. This design information will be used for identifying the resources needed from the factory and creating the necessary process plans for the products. Refer to Figure 3.1.

The tool then uses the product design data and factory information to create operations-input and machine-input data sets. Using these data sets, the process plans are created. The tool can be used to analyze the following two scenarios:

Case 1 : a completely new product is introduced into the system with different design parameters and different processing operation requirements, and

Case 2 : an existing product is modified to improve its performance which does not change the component set but merely modifies the process plans.

In either case, two instances of the model are created: one for the existing product set and another after the new product has been introduced or an existing product has been modified.

3.2.1 Algorithm:

The tool uses the following algorithm to estimate the manufacturing system performance for a given product set. Figure 3.1 shows a block diagram corresponding to the *Inputs* part of the algorithm.

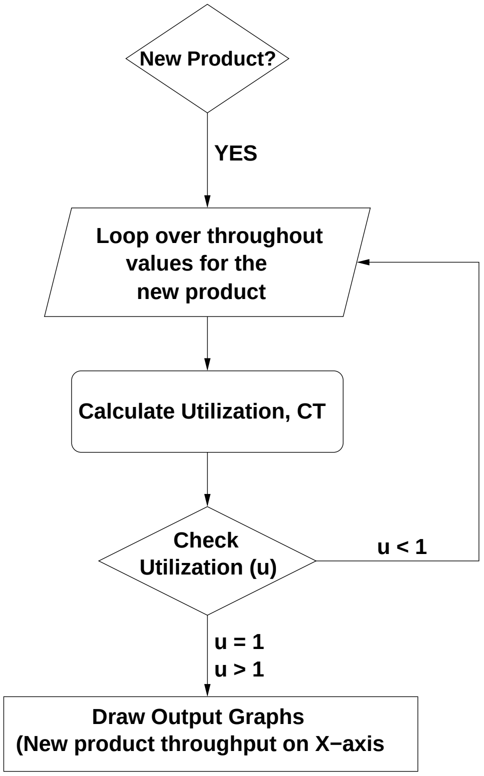
New Product?
YES
Loop over throughout values for the new product
Calculate Utilization, CT
Check Utilization (u)
u < 1
u = 1
u > 1
Draw Output Graphs (New product throughput on X−axis

Figure 3.2: Flow chart for Case 1

1. Inputs:

    (a) The factory file, which describes the resource set $\mathcal{J}$ present in the manufacturing system

    (b) For the set $\mathcal{P}_{old}$ of existing products, a design file containing the set $\mathcal{D}_{old}$ of critical design information and an operations file containing the set $\mathcal{O}_{old}$ of operations sequences

    (c) The set $\mathcal{P}_{new}$ contains the products in $\mathcal{P}_{old}$ and the new product x (if a new product is to be introduced, else $\mathcal{P}_{old}$ with the modified existing product). For $\mathcal{P}_{new}$, a design file containing the set $\mathcal{D}_{new}$ of critical design information and an operations file containing the set $\mathcal{O}_{new}$ of operations sequences

2. Program Execution:

    (a) For each product in $\mathcal{P}_{old}$, calculate the necessary processing times based on the critical design information in $\mathcal{D}_{old}$. Create the set $\mathcal{T}_{old}$ of processing times.

    (b) From $\mathcal{O}_{old}$ identify and create the set $\mathcal{R}_{old}$ of required resources.

    (c) If $\exists\, o \in \mathcal{O}_{old}$ such that o requires a resource r where $r \notin \mathcal{J}$, advise user. Exit

    (d) Create the set $\mathcal{Q}_{old}$ of process plans $\forall\, p \in \mathcal{P}_{old}$ using information from $\mathcal{R}_{old}$ and $\mathcal{T}_{old}$

    (e) Using $\mathcal{Q}_{old}$, calculate the utilization u_r, $\forall\, r \in \mathcal{R}_{old}$. If $u_r \geq 1$ for any $r \in \mathcal{R}_{old}$, advise user of insufficient capacity. Exit

(f) Calculate the cycle times $CT_r^{\star}$, $\forall\, r \in \mathcal{R}_{old}$ and CT_p, $\forall\, p \in \mathcal{P}_{old}$. Create output files for each resource and product

(g) Repeat Steps 2a through 2d for $\mathcal{P}_{new}$ using $\mathcal{D}_{new}$ and $\mathcal{O}_{new}$ to create $\mathcal{Q}_{new}$

(h) Using $\mathcal{Q}_{new}$, calculate the utilization u_r, $\forall\, r \in \mathcal{R}_{new}$.

(i) Calculate the cycle times $CT_r^{\star}$, $\forall\, r \in \mathcal{R}_{new}$ and CT_p, $\forall\, p \in \mathcal{P}_{new}$. Create output files for each resource and product

(j) If $u_r < 1 \;\forall\, r \in \mathcal{R}_{new}$, then increase the throughput of x, the new product, by a predetermined amount (or if one of the existing products is to be modified, increase the value of the counter) and return to Step 2h

(k) Plot results. Exit

Figure 3.2 shows the flow chart for Case 1 listed above (i.e. when a new product is introduced into a system already processing a set of existing products) corresponding to Steps 1a, 1c, and 2g through 2k while Figure 3.3 illustrates the flow chart for Case 2 (i.e. when an existing product is modified).

The tool dispenses design improvement advice to the designer based on which resource is over-utilized. The system recognizes that the resource has become overutilized for new product throughput higher than that specified by the user. It apprises the user of the situation and indicates the allowable increase in new product throughput before a resource becomes overutilized. It then advises the user to add capacity in the form of machines if the resource is an automated station or personnel for a manual resource.

NOTES: The manufacturing system for the tool is a multiple resource multiple product system. The processing time distributions for each of the parts on each resource are known. These parts follow the FIFO (first in first out) rule for processing on a resource. This means that irrespective of the priority or importance of the parts due for processing on a resource, parts are given preference for processing depending on the relative times of arrival of the parts at the machine. If a batch is being processed, the entire batch is completed before any other job is undertaken. Between the resources queues may form and the stations are assumed to have infinite buffers. As the algorithm explained, the tool requires creating two product-manufacturing system instances for each run, one corresponding to the existing products and one with the new product (or with the modified existing product corresponding to Case 2 above).

3.2.2 Outputs

The outputs for the DFP tool are in the form of:

- advice to the user for adding workstations that the design needs but are not present in the current manufacturing system,
- advice to the user regards adding resources to overutilized workstations,
- graphs plotting an output variable (such as utilization or manufacturing cycle time) for the processing resource under consideration versus an input parameter (such as throughput),
- graphs plotting an output variable (such as manufacturing cycle time) for the product under consideration versus an input parameter (such as throughput), and

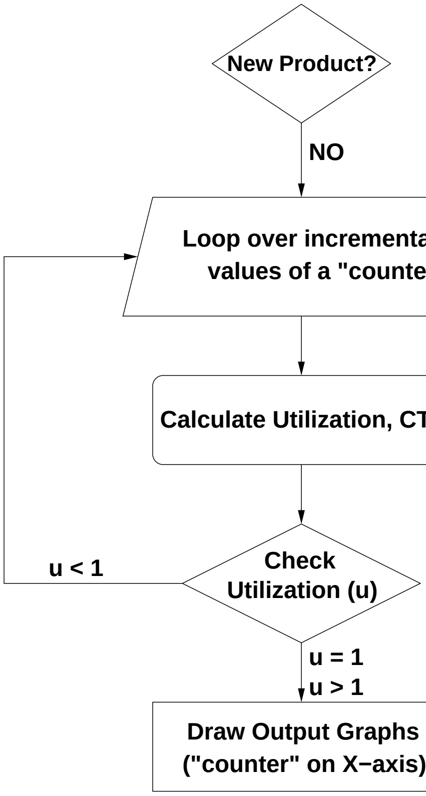
New Product?
NO
Loop over incremental values of a "counter"
Calculate Utilization, CT
Check Utilization (u)
u < 1
u = 1
u > 1
Draw Output Graphs ("counter" on X−axis)

Figure 3.3: Flow chart for Case 2

- advice to the designer to aid in changing the product designs in order to best utilize the production resources.

This tool developed helps the designer estimate how the manufacturing system will perform while processing the product designs. The user is required to input the desired throughput level for the new product. The tool initially confirms that all the resources needed to make the product, as listed in the product design information are actually present in the current factory. If any required resource is not present in the available resource set, the tool advises the designer accordingly. Once the tool has located all the resources needed for processing the new product, it then evaluates whether it is possible to achieve the desired throughput rate. Once feasibility has been established, it is useful if the designer is made aware of the latitude available to increase production requirements within available capacity. The advantages of such information are two-fold:

1. it allows the product development team to propose larger production targets, and
2. it tells the designer that if the market demand is higher than projected, the existing manufacturing system can match the increased requirements.

Case 1 : **New product is introduced**.

In order to make this information available to the designer, it is necessary to treat the new product throughput as an independent variable. The new product throughput is therefore increased in steps and the performance of the system is plotted as a function of the new product throughput.

For each of the resources that is used for processing the product(s) the utilization of the resource is plotted against the throughput of the new product introduced into the system. The graphs begin at the specified value of new product throughput, which increases until some resource being used in the processing becomes overutilized. Figures 3.5, 3.6 and 3.7 show examples of such plots of Utilization vs Throughput and Cycle Time vs Throughput. This indicates to the designer the allowable throughput for the new product. If for the specified throughput itself, a resource reaches maximum utilization ($u_r \geq 1$), the tool advises the designer of the situation and proffers suggestions.

Initially, for the plot of Utilization vs Throughput for each resource, the utilization when the old product set (consisting of the products already being manufactured) is being processed, is plotted against zero throughput for the new product. On the same plot, the utilization of the resource after the new product has been introduced is plotted against the throughput value of the new product, which is then increased in steps. Similarly, in the Cycle Time vs Throughput plot, the cycle time for each product before new product introduction is plotted against zero new product throughput while the cycle times of the product after new product introduction are plotted against corresponding values of the new product throughput. These plots indicate the influence of new product introduction on the manufacturing system performance and the existing products.

Case 2: **No new product in introduced**.

When one or more of the existing products are modified, there is in effect no *new product* and hence the resource utilization is now plotted against a *counter*. The existing product throughputs are a function of this counter. Incrementing this counter value, hence increases the throughput of all the existing products. In a similar manner, the cycle times for each of the products being processed are plotted against the same independent variable to give the next set of output graphs.

In this case, as for Case 1 above, the output parameters are plotted before changing the design parameters of any product (existing product set) and after such change has been made (new product set), on the same graph. Then, the plots indicate the effect that changing any of the products has on the manufacturing system performance and the other products.

The tool also creates a bar graph plot of the utilization at each resource for the given throughput value. Figure 3.8 shows an example of the bar graph. Plotted on the X-axis are the various resources being used for this example. The values of the resource utilization are plotted for the new product throughput value specified by the user. The algorithm in Subsection 3.2.1 gives the operating procedure for the tool.

Redesign The tool first establishes feasibility of making the new product in the present processing scenario. To do this, the tool checks that all the resources needed to make the new product are present in the factory and that no resource is becoming overutilized when the system begins manufacturing the new product. If any product requires a workstation not present in the current factory, the tool informs the user of this exception. If all the required workstations are present in the factory but some resource is becoming overutilized ($u > 1$), the tool advises the user of the need to add capacity to the resource and allows the user to add capacity to the overutilized workstation.

Now, after establishing feasibility, the tool then looks at congestion in the manufacturing system to determine if there is a need for changes to the product design. The tool identifies aspects of the product design that may be changed based on the following heuristic.

Redesign Heuristic : The tool uses a combination of resource utilizations, workstation cycle times, and product cycle times to indicate to the design development team aspects of the design that may be modified in order that the performance of the manufacturing system processing the product design may be improved.

1. The tool first identifies as critical workstations, those workstations that have the highest resource utilization (greater than a threshold value equal to 0.9) and sorts these in order of decreasing utilization. The nonlinear relationship between the station cycle time and resource utilization means that utilizations above the threshold significantly increase the station manufacturing cycle time.

2. The tool then identifies the set of products being processed on each critical workstation. It checks the contribution of the workstation cycle time to the total manufacturing cycle time for the new product, if the new product is being processed on these critical workstations.

3. The tool then identifies the five workstations from the set of critical workstations with the highest workstation cycle times.

4. Next, the tool parses the key product design data set for the new product to identify the design features being processed on each critical workstation.

5. The tool identifies the set of design features for the new product which influence the processing times for the largest subset of the critical workstation

set. These are the critical design features from the new product that most influence the product manufacturing cycle time.

6. The user is advised to consider the possibility of modifying these critical design features in a way that either avoids visiting the critical workstations or minimizes the processing time at such workstations.

Using the DFP Tool. It should be noted here that this heuristic advises the design development team to change critical design features in such a manner so that no workstation in the manufacturing system is highly utilized ($u > 0.9$). If no station is highly utilized (all stations have $u < 0.9$), the tool does not identify any need for design modifications. Thus, for example if, for a system with ten workstations, the utilization of eight workstations is greater than 0.85 but less than 0.9, the tool will not generate any design suggestions for the product. On the other hand, if one workstation has utilization greater than 0.95 and all other workstations have utilizations less than 0.7, the tool will identify the critical design features associated with the one critical workstation and advise design modifications.

If the tool finds an overutilized resource and advises the user to add capacity to this resource, the capacity change is restricted to that run of the tool without changing the factory configuration file. This gives the designer an opportunity to modify the new product design and analyze the performance of the manufacturing system for the modified product designs before considering changes to the manufacturing system configuration.

3.3 Application: Microwave Module DFP Tool

The earlier section explained the DFP tool created based on the proposed approach. This section presents results from customizing the DFP tool to evaluate designs of microwave modules being processed in an electronics assembly shop. This application uses data that an electronic systems manufacturer provided and other synthetic data that was created as part of this research effort. For details about the process planning and processing time estimation, see Minis *et al.* [91].

3.3.1 The Microwave Module

Modern microwave modules (MWMs) have an artwork layer that includes numerous functional components of the circuit. The artwork lies on the dielectric substrate, which is attached to a ground plane that also serves as a heat sink. In addition to the integrated components, MWMs may carry hybrid components, which are assembled separately using techniques such as soldering, wire bonding, and ultrasonic bonding. Mounting these components often requires holes, pockets, and other features in the substrate. Figure 3.4 shows a photograph of a microwave module.

The product's aluminum substrate has a Teflon dielectric layer. The substrate needs to be machined on 4 sides and has 8 holes. In addition, the microwave modules have surface-mount electronic components. The results presented here are for a shop that currently produces two products (*MWM Product 1* and *MWM Product 2*) and is introducing a third (*Improved MWM*).

Table 3.1 gives critical information about the new product while Table 3.2 lists the sequences of operations for the products. The process planning module uses the critical design information about the new product to estimate the part processing

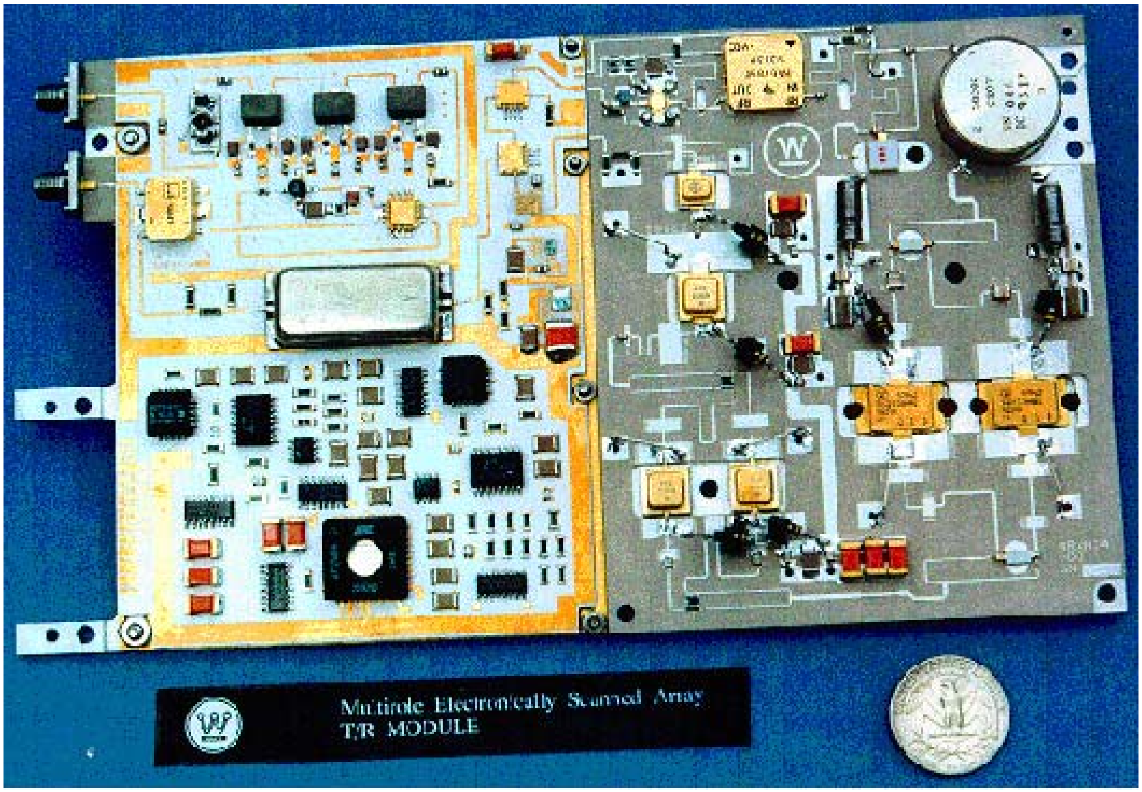

Figure 3.4: The microwave module

Attribute	Value
Number of finished surfaces	4
Length (in.)	4.0
Width (in.)	2.0
Number of holes	8
Average hole depth (in.)	0.25
Etch area (in.^{2})	3.0
Etch thickness (in.)	0.0001
Plating thickness (in.)	0.00001
Mounted Electronic Components	16

Table 3.1: Critical Design Information for Improved MWM

times and setup times at each workstation as listed in Table 3.3. Table 3.4 lists the desired throughput of the three products. As seen in Table 3.2, the existing products (*MWM Product 1* and *MWM Product 2*) each require 7 operations. The new product (*Improved MWM*) requires 9 operations. The portion of the substrate with no artwork is provided with added insulation in the new product. The aim is to minimize possibility of conduction with any extraneous electronic components which may come in contact with the aluminum substrate. This added insulation requires a coating operation on an *Insulator* resource. The process planner uses previously developed rules and algorithms for MWM process planning [92]. The availability of the resource set is between zero and one.

Table 3.3 lists the mean processing requirements for each operation. (The design features for the old and new products are the same with the operation set being different.) Each processing time has some variability as well. The processing times and setup times for all the operations have gamma distributions (See NOTES below for details). It is assumed that the yield at all stations is 100% ($y_{ij} = 1$).

Op. No.	MWM Product 1	MWM Product 2	Improved MWM
1.	Grinding	Milling	Milling
2.	Drilling	Grinding	Grinding
3.	Etching	Drilling	Drilling
4.	Electroplating	Etching	Insulation
5.	Automatic Assembly	Electroplating	Etching
6.	Manual Assembly	Automatic Assembly	Electroplating
7.	Testing	Testing	Automatic Assembly
8.			Manual Assembly
9.			Testing

Table 3.2: MWM Process Plans

Product i	MWM Product 1	MWM Product 2	Improved MWM	Mod. Agg. Proc. Time	
Job processing time (mins)	t_{1j}^{+}	t_{2j}^{+}	t_{3j}^{+}	$t_j^{\star}$	$c_j^{\star}$
$j = 1$: Drilling	11	11	11	12	0.045
$j = 2$: Milling	0	2	2	2	0.14
$j = 3$: Grinding	2.22	2.22	2.22	2.22	1.49
$j = 4$: Electroplating	4	4	4	4	0.05
$j = 5$: Etch	62.4	62.4	62.4	62.5	0.05
$j = 6$: Insulator	0	0	82.68	82.68	0.085
$j = 7$: Automated Assembly	11.67	11.67	11.67	11.67	0.045
$j = 8$: Manual Assembly	32	0	32	32.06	0.22
$j = 9$: Testing	60.12	60.12	60.12	60.12	0.06

Table 3.3: Products Process Plans

3.3.2 The Manufacturing System

The manufacturing facility for these microwave modules is a batch manufacturing system. There is a milling machine and 2 grinding machines that can machine the surfaces to be finished, and a drilling machine to drill the holes. The facility has a plating workstation, 4 etch workstations, 3 insulation workstations, 2 workstations for automated assembly, and 2 workstations for manual assembly. The automated assembly workstation has a screen print machine, a pick-and-place machine, and a reflow oven. The material handling between these machines is automated. The manual assembly workstation has two employees who can attach other component types. The facility has 4 technicians on testing stations to test and tune microwave modules.

NOTES:

1. The values of the Variances for the processing and the setup times are calculated based on the formula for the variance of the *m-Erlang* distribution. In this case $m = 2$.

2. The value of SCV for the interarrival time is 0.5 for all the products. This is because the SCV which is given by

   $$\text{SCV} = \frac{\text{Variance}}{(\text{Mean})^2} \tag{3.25}$$

   for the *2-Erlang* distribution is a function of only the shape parameter α (α = 2 for *2-Erlang*).

Product i	MWM Product 1	MWM Product 2	Improved MWM
Throughput T_i (parts/hour)	6.5	6.5	3.5
Batch size B_i (parts/batch)	10	10	10
SCV arrival	0.5	0.5	0.5

Table 3.4: Desired Product Throughput

Station	j	Util. u_j	
		Existing Product Set	New Product Set
Drill	1	0.24	0.31
Mill	2	0.022	0.034
Grinder	3	0.024	0.031
Plating Machine	4	0.088	0.11
Etch	5	0.34	0.43
Insulator	6	0	0.16
Auto Assembly	7	0.13	0.16
Manual Assembly	8	0.18	0.27
Test Station	9	0.33	0.42

Table 3.5: Resource Utilization

3.3.3 Capacity Analysis

Using the queuing network model presented above, the tool can calculate the average resource utilization at each station. Table 3.5 displays these results for the existing as well as new product sets. Since all $u_j < 1$, all of the stations have sufficient capacity to process the new product. (Note that the existing products do not require processing on the *Insulator* and hence $u_6 = 0$.)

3.3.4 Estimating the Manufacturing Cycle Time

Queuing network model. The tool uses the queuing network model to estimate the average manufacturing cycle time at each workstation. Based on the routing

Station	Average Cycle Time (mins)	
	Two Products	Three Products
Drill	13.00	14.00
Mill	2.07	2.10
Grinder	2.07	2.28
Plating Machine	5.40	6.00
Etch	63.60	64.80
Insulator	0.00	85.20
Auto Assembly	12.00	12.00
Manual Assembly	34.20	36.00
Test Station	62.40	64.20
Total	194.74	286.60

Table 3.6: Cycle Time Estimates: Queuing Network Model

for the new product, is estimates the average manufacturing cycle time as the sum of these workstation cycle times. Table 3.6 summarizes these calculations. The total, when the system is processing the existing products, is 194.74 minutes or 3.25 hours, and 286.60 minutes or 4.78 hours when it is processing the new product set. Table 3.7 shows the cycle time multiple and the sensitivity for the new product.

Outputs. As explained earlier, the throughput of the new product in increased in steps and performance of the manufacturing system is evaluated for different values of new product throughput. This continues until the utilization of some resource exceeds one. The highest feasible throughput is 21 parts per hour. Figure 3.5 shows the plot of utilization of the Etch resource as a function of new product throughput.

Figures 3.6 and 3.7 show the manufacturing cycle times of one of the existing products and the new product as a function of new product throughput respectively.

Station	j	Multiple M_j	Sensitivity S_{3j}
Drill	1	1.22	2.63
Mill	2	1.04	3.68
Grinder	3	1.01	2.15
Plating Machine	4	1.41	3.05
Etch	5	1.03	2.20
Insulator	6	1.01	10.29
Auto Assembly	7	1.02	2.19
Manual Assembly	8	1.09	3.90
Test Station	9	1.04	2.26

Table 3.7: Sensitivity Analysis (New Product)

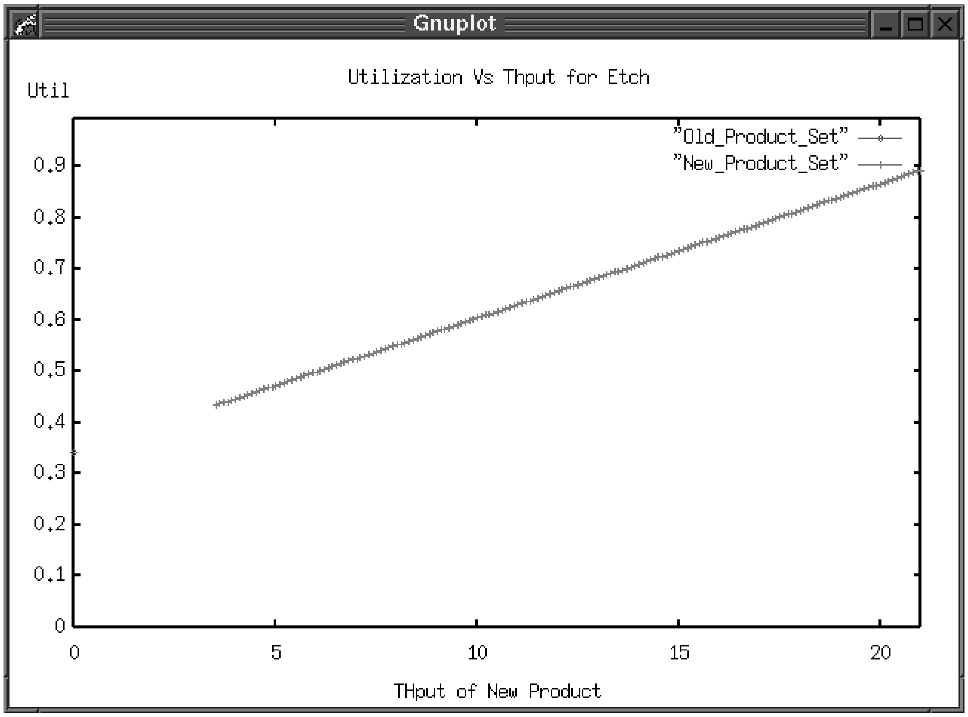

Figure 3.5: Typical tool output showing utilization of Etch as a function of new product throughput

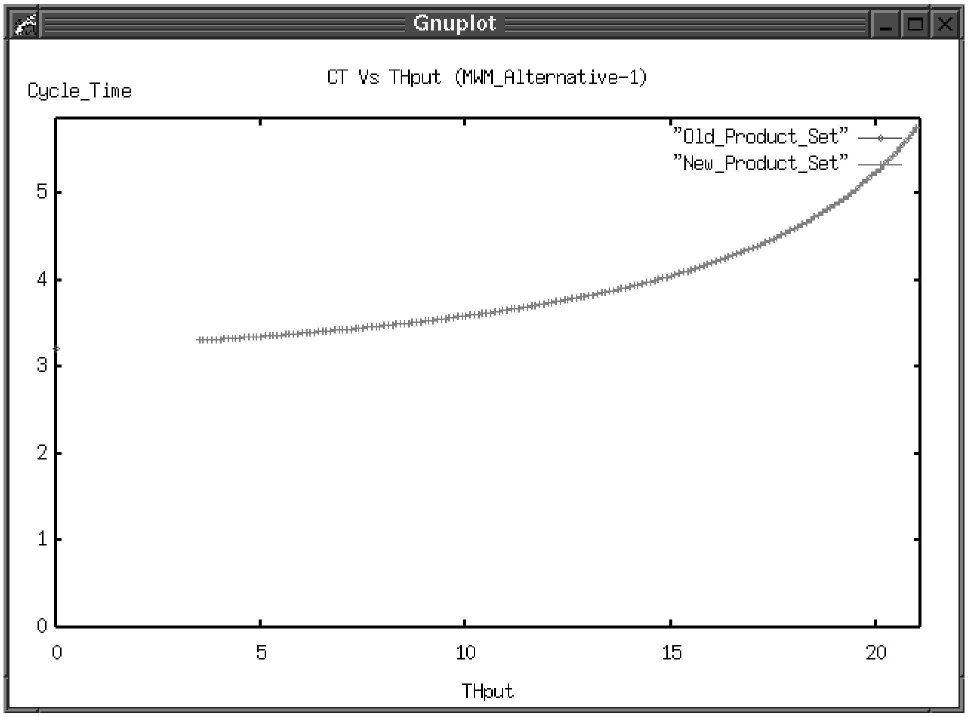

Figure 3.6: Typical tool output showing cycle time of an existing product as a function of new product throughput

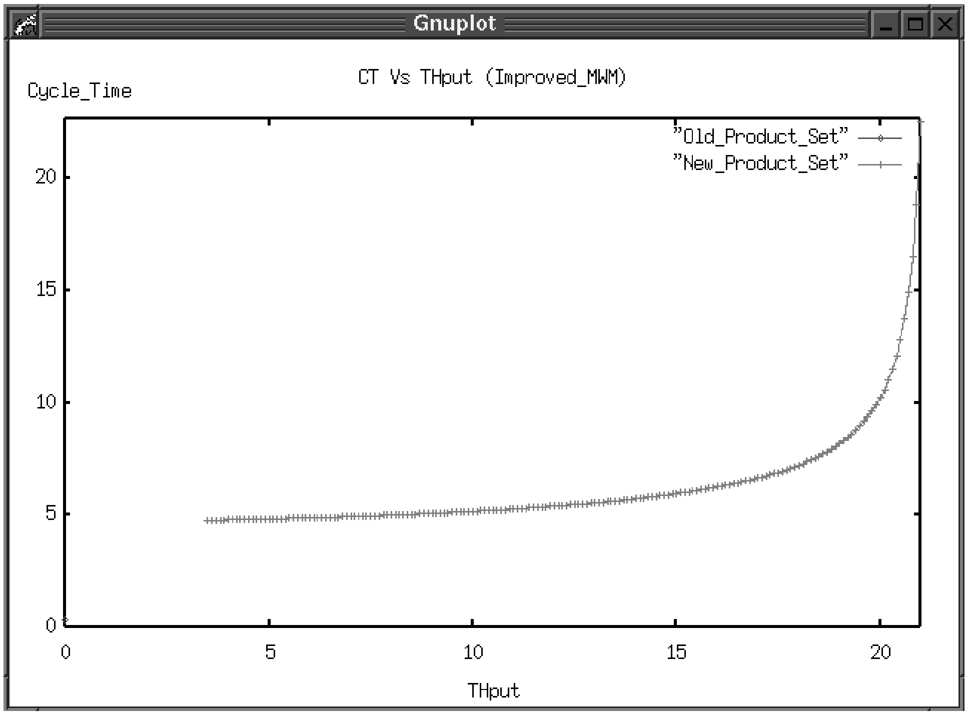

Figure 3.7: Typical tool output showing cycle time of the new product as a function of its throughput

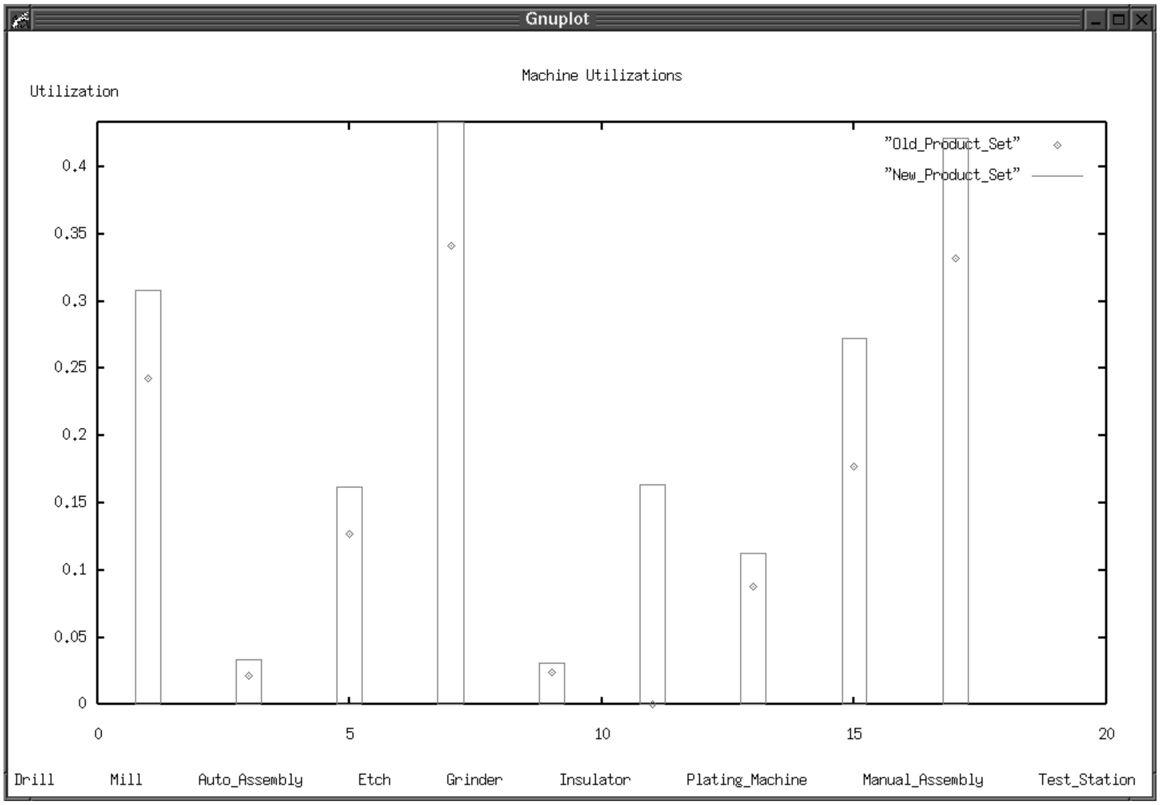

Figure 3.8: Typical tool output showing bar graph of utilization of various resources

Figure 3.8 plots the utilization of the resources in the factory at the new product throughput specified by the user.

Design improvement suggestions. For comparison, the product development team can view a baseline scenario by setting the desired throughput of *Improved MWM* to zero. Table 3.6 shows the average cycle time at each station when the facility manufactures no Improved MWM and when the facility adds *Improved MWM*. As may be seen, at the specified new product throughput the utilization of all the resources is less than 1. Therefore, the manufacturing system is capable of producing the new product.

The plots with new product throughput as the independent variable indicate to the user the allowable increase in new product throughput and present system behavior as the new product throughput increases. In this example, as the new product throughput is increased to 20.92 units, the utilizations of various resources in the processing sequence are well below 1, except Etch ($u = 0.89$) and Test Station ($u = 0.87$). At this throughput value, however, the manufacturing cycle time for the new product has risen to 1354 min. In addition, the change in utilization is greatest for the Insulator. Further, using these results, the tool advises the user that adding capacity to, or reducing the resource requirements for the Etch and Test Station workstations would lead to better performance of the manufacturing system. The tool then indicates that this would require a reduction in the etch area or the etch thickness or both.

Using the plots in Figures 3.7 and 3.8, the product development team has the option to increase the new product throughput so long as the manufacturing cycle time remains less than the acceptable value.

3.4 Summary

This chapter presented a specific approach that determines how manufacturing a new product design affects the performance of the manufacturing system. Design for production (DFP) includes design guidelines, capacity analysis, and estimating manufacturing cycle times. Performing these tasks, like other DFM techniques, early in the product development process can reduce product development time.

This chapter has developed a decision support tool that performs DFP analysis. Unlike previous approaches, the tool quantifies how introducing a new product increases congestion in the manufacturing system. This requires only the critical design information needed to create a process plan and estimate processing times, so it can be used early in the product development process. This tool employs an approximate queuing network model that estimates the manufacturing cycle time of the new product. The tool also calculates the capacity requirements and estimates the average work-in-process inventory. This provides feedback that the product development team can use to reduce manufacturing cycle time. Plots of manufacturing cycle time for the products and resource utilization describe the impact of the new product on the manufacturing system and the existing product set. The tool can quickly evaluate changes to the new product design or changes to the manufacturing system.

Chapter 4

Case Study: Embedding Passives in Printed Circuit Boards

This chapter presents an application domain for the various models and tools developed in the previous chapter. The utility of the models and applicability of the algorithms and tools presented is best demonstrated by applying them to a product domain. The domain chosen here is Printed Circuit Boards (PCBs). Particularly, the design of a PCB with discrete and embedded passive components is analyzed to evaluate the impact of embedding passives on the manufacturing system performance.

Section 2.11 explained the technology in *embedding* passive components into the substrate of a PCB along with relevant literature. Section 2.12 presented some process and design considerations arising from the decision to embed a portion of the passive components. Section 4.1 explains the various processing steps in making a conventional PCB substrate and making one incorporating *embedded passive* components. Sections 4.2 and 4.3 explain various design and processing considerations associated with embedding passive components. Section 4.4 applies the DFP tool developed based on the models in Chapter 3 to understand the effects of embedding passive components for an AS900 CPU printed circuit board, when the manufacturer makes a product mix comprising boards with and without embedded passive components.

4.1 Printed Circuit Board Manufacturing

This section details the processing steps involved in the manufacture of a PCB. The following two subsections list the processing steps for a double sided PCB and a multilayer PCB.

4.1.1 Double Sided Printed Circuit Boards

Manufacturing a double sided printed circuit board involves the following set of operations [27]. (Note that this set only includes the steps associated with making the substrate without reference to techniques for incorporating passive components such as surface mount (SMT) or embedding techniques.) The associated explanation for each step denotes the unit of processing at that step:

1. Material Preparation: Set of panels from which the board is formed (called a stack).

2. Stack and Pin: Set of panels that form the board.

3. Drilling: Set of panels that form the board.

4. Deburr: Set of panels that form the board.

5. Electroless Copper Plating: Batch of panels pass through chemical baths.

6. Imaging: One panel at a time.

7. Pattern Plating: Batch of panels in series of chemical baths.

8. Etch Strip: Batch of panels through conveyorized baths.

9. Solder Mask: One panel at a time.

10. Solder Coating: One panel at a time. (dipping in solder bath and removing excess solder)

11. Gold Plating: One complete panel or sheared into constituent boards.

12. Component Legend: One complete panel or one board.

13. Electrical Test: One board or a panel. Rejects usually discarded, seldom reworked.

14. Final Inspection: Visually inspect one finished PCB.

4.1.2 Multilayer Printed Circuit Boards

This subsection presents the steps for a multilayer (six layer), plated through hole, solder coated board with SMT components and no embedded passives [27].

The internal layers of the board are made from double sided laminates. The steps after the substrate has been assembled (outer layers and internal layers bonded together) are similar to those followed in double sided PCB manufacture, as detailed above.

1. Material Preparation: Layers that form the panel.

2. Clean: Layers cleaned in chemical baths in batches.

3. Imaging: One layer at a time.
4. Etch Strip: Layers in chemical baths in batches.
5. Inspect: One layer at a time. Rejects discarded.
6. Surface Treat (Oxide): Inner-layers in batches pass through chemical baths.
7. Layup: Inner and outer layers arrive and stack is created and surface treated as a job.
8. Lamination: Sets of inner and outer layers are combined to form panels.
9. Stress Relief: Panels are baked in batches
10. Fabricate Tooling Holes and Trim Edges: One panel at a time.
11. Drilling: One panel at a time.
12. Deburr: One panel at a time.
13. Electroless Copper Plating: Batch of panels pass through chemical baths.
14. Imaging: One panel at a time.
15. Pattern Plating: Batch of panels in series of chemical baths.
16. Etch Strip: Batch of panels through conveyorized baths.
17. Inspect: One panel at a time.
18. Solder Mask: One panel at a time.

These are the significant processing operations to be performed to manufacture a multilayer printed circuit board substrate. Subsection 4.1.4 lists the operations sequence for a multilayer PCB with embedded passive components.

4.1.3 Steps Involved in Embedding a Resistor Component into the PCB

For a PCB that has embedded passives, each inner layer with embedded passives needs to undergo the following steps in addition to those for the traditional PCB that has all discrete passive components:

1. apply photoresist polymer,
2. expose polymer to create vias,
3. cure dielectric polymer,
4. develop photoresist,
5. strip photoresist,
6. apply photoresist,
7. print conductor protection pattern,
8. electroless plate resistive layer,
9. etch copper remove excess,
10. strip photoresist, and
11. measure values and clean.

Note that this is only one (of several) approaches to building an embedded resistor.

4.1.4 Making a Multilayer PCB Substrate With Embedded Passive Components

This research studied the steps outlined in Subsections 4.1.1, 4.1.2 and 4.1.3, and developed Figure 4.1 along with Tables 4.1 through 4.16 to depict various details, including numerous smaller tasks that compose the broad steps listed in the previous subsections. Note that the processing operations and substeps shown here are one way of manufacturing the printed circuit board. There may exist other operations, steps and sequences involved in making the PCB or possibly different operation sequences involving the operations and substeps listed here.

Figure 4.1 shows a block diagram representing the network of operations for manufacturing the printed circuit board substrate before the surface mounted components are assembled. Blocks IA to B or C represent operations that need to be conducted on the inner layer pairs before they are assembled together in Block D, the Kitting or Lay-up stage. The figure shows a representative example of a PCB with five inner layer pairs along with the two outer layer pairs. Two inner layer pairs (with Blocks IA to C) contain embedded passive components while three inner layer pairs (with Blocks IA to B) do not have such embedded components. Blocks E to N include operations that the multilayer PCB substrate must complete. When the PCB enters processing sequence represented by Blocks E to N, the inner layer pairs and outer pairs have been composed and laminated together so that all processing to be done on the inner layer pairs has been completed and further processing to be done is limited to the outer layers only. Tables 4.1 to 4.16 list the individual steps that are combined to form the corresponding operations block. The following four operations are just the indicated step:

IA - Clean copper clad laminate

IB - Resist Application	
1	post clean rinse
2	apply resist to laminate - side A
3	apply resist to laminate - side B

Table 4.1: Sub-processes - Block IB

IC - Artwork	
1	punch tooling holes
2	insert artwork - side A
3	insert artwork - side B
4	artwork registration
5	board clean
6	setup - exposure bulb replacement

Table 4.2: Sub-processes - Block IC

ID - Mylar Removal	
1	expose both sides
2	mylar removal - side A
3	mylar removal - side B

Table 4.3: Sub-processes - Block ID

IE - DES (Develop-Etch-Strip) Line	
1	develop both sides
2	forced air dry
3	etch both sides
4	post etch rinse
5	post etch dry
6	strip both sides
7	post strip rinse
8	post strip dry
9	DES line regen

Table 4.4: Sub-processes - Block IE

IF - Testing

B - Insert copper foils on both sides of inner laminate

K - Inspection

The steps constituting an operations block may be completed at a single station or multiple stations which may be a bath or a series of baths inter-spaced with associated drying and rinsing operations and the mean processing time for the constituent step is a small fraction of the total product processing time. Note that this is one possible process of making a PCB. There are other ways of manufacturing the PCB using different materials.

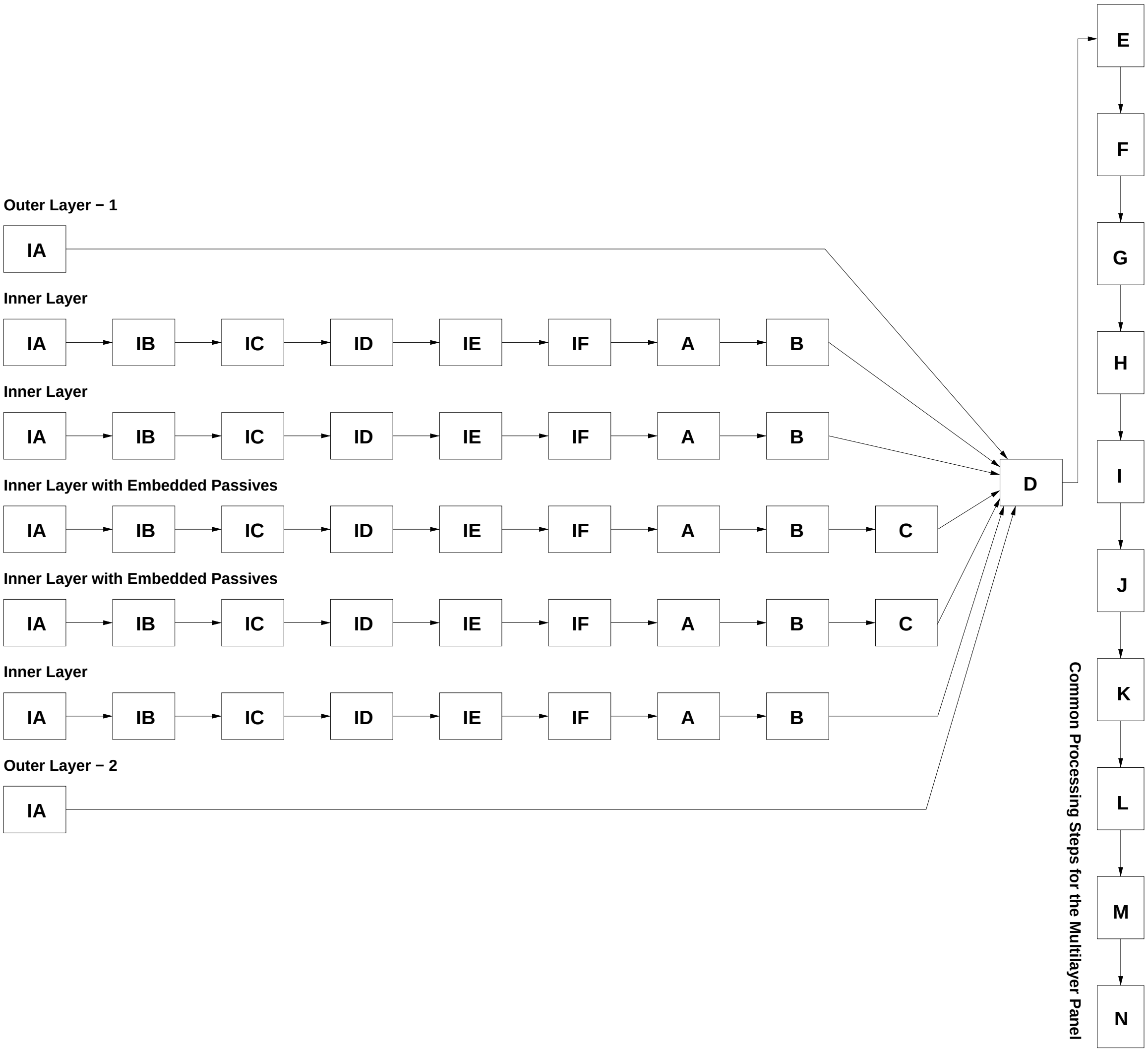

Figure 4.1: Operation Network for PCB Manufacture

A - Inner Layer Pair Cleaning	
1	insert layer pair
2	acid clean
3	rinse C
4	microetch
5	post microetch rinse
6	acid clean
7	rinse A
8	alkali pre-dip
9	rinse B
10	oxide treatment
11	triple post oxide rinse
12	post oxide dry

Table 4.5: Sub-processes - Block A

C - Embedded Resistor Steps	
1	apply photoresist polymer
2	expose polymer to create vias
3	cure dielectric polymer
4	develop photoresist
5	strip photoresist
6	apply photoresist
7	print conductor protection pattern
8	electroless plate resistive layer
9	etch copper remove excess
10	strip photoresist
11	measure values and clean

Table 4.6: Sub-processes - Block C

D - Kitting or Layup of Inner Layer Pairs	
1	punch registration holes
2	layup

Table 4.7: Sub-processes - Block D

E - Lamination	
1	lamination
2	trim flash
3	measure thickness

Table 4.8: Sub-processes - Block E

F - Drill Setup	
1	X-ray registration
2	drill setup 1
3	drill setup 2
4	drill/deburr

Table 4.9: Sub-processes - Block F

G - Desmear	
1	de-smear 1
2	de-smear 2
3	de-smear 3
4	post de-smear clean

Table 4.10: Sub-processes - Block G

H - Electrolysis (Cu Layer Deposition)	
1	bath - neutraganth
2	tap rinse 1
3	bath - securi HCF 45
4	tap rinse 2
5	bath - NH_3 perusulfate
6	tap rinse 3
7	bath - neoganth activo
8	tap rinse 4
9	bath - neoganth reducer
10	bath - noviganth HC
11	bath DI rinse 1
12	bath - acid dip
13	bath - DI rinse 2
14	post electrolysis dry
15	bath - Cu plate electrolyte
16	bath - DI rinse
17	post electrolytic Cu dry
18	vibratory sand

Table 4.11: Sub-processes - Block H

I - Apply Dry Film (for DES)	
1	punch-in-process micro
2	inspection-in-process micro
3	clean laminate
4	apply resist to laminate (side A)
5	apply resist to laminate (side B)
6	insert artwork (side A)
7	insert artwork (side B)
8	artwork registration (side B)
9	exposure maintenance
10	expose both sides
11	mylar removal (side A)
12	mylar removal (side B)

Table 4.12: Sub-processes - Block I

J - DES (Develop-Etch-Strip)	
1	develop both sides
2	post develop rinse
3	forced air dry
4	microetch
5	Cu-plate both sides
6	tap rinse 1
7	NH_3 perusulfate
8	tap rinse 2
9	dip - H_2SO_4
10	tap rinse 3
11	dip HBF_4
12	Sn/Pb plate-side
13	tap rinse 4
14	inspection
15	strip both sides
16	post strip rinse
17	hot air dry
18	etch both sides
19	bath - etch finish
20	post etch rinse
21	post etch dry

Table 4.13: Sub-processes - Block J

L - Hot Air Solder Leveling	
1	oxide removal
2	solder bath
3	flux removal
4	forced air dry

Table 4.14: Sub-processes - Block L

M - DES for Solder	
1	apply solder mask (side A)
2	apply solder mask (side B)
3	insert artwork (side A)
4	insert artwork (side B)
5	artwork registration (side B)
6	exposure maintenance
7	expose both sides
8	mylar removal (side A)
9	mylar removal (side B)
10	develop both sides
11	post develop rinse

Table 4.15: Sub-processes - Block M

N - Final Routing and Inspection	
1	thermal cure
2	liquid silk screen
3	serialization
4	routing setup
5	routing
6	electrical test
7	final micro inspection
8	manual inspection

Table 4.16: Sub-processes - Block N

4.2 Material Flow Considerations for PCB Manufacture

This section details the various material flow and processing scenarios that could be involved in making a PCB. These possibilities arise from the different types of processing stations that the PCB visits as a part of its processing sequence, ranging from electrolyte baths to component assembly stations. It is beneficial to enumerate and understand these considerations here, since these affect the batch size of the product when it visits different stations forming part of the processing sequence. Since the manufacturing cycle time at a station is a function of the batch sizes of the products visiting the station, these material flow considerations directly affect the manufacturing cycle time at the station.

4.2.1 Processing Possibilities

1. Some stations process batches of panel-layers or panels simultaneously. An example of this type of station is a chemical bath for processes like etching or coating. A batch of panel-layers is processed in each bath before being moved to the next bath. Thus, for the station, the job arrives as a batch of panel-layers or panels and leaves as a batch of panel-layers or panels.

2. Some stations perform operations on one panel-layer or one panel at a time. Thus, the input for such a station is a panel-layer or a complete panel at a time and the output is also one panel-layer or a complete panel. Similarly, some stations take input as a panel and output a set of boards.

3. Some stations combine panel-layers into a complete panel. Sets of such panel-

layers for a panel as designed are input to the station and they are combined into a panel. Thus the output from such a station is a completed panel.

4. Stations such as those that embed capacitors combine two batches of panel-layers into one batch of panel-layers.

The manufacturing cycle time at a station is a function of the batch size of the products arriving at the station. The manufacturing system models detailed in Chapter 3 assumes the batch size at the output of the station to be a function of the output batch size from the previous station in the processing sequence and the yield at the station only. The batch size of the product here is likely to change during the processing sequence as a result of different processing considerations listed above. To offset such changes corresponding factors need to be included in the mean processing time calculations at these stations. These are explained in greater detail in Section 4.4.1.

4.2.2 Movement Scenarios

1. Batch of panel-layers or panels arrives at a station ⟶ batch of panel-layers or panels is processed simultaneously ⟶ batch of panel-layers or panels leaves station.

2. One panel-layer or panel arrives at a station ⟶ one panel-layer or panel is processed ⟶ one panel-layer or panel leaves station.

3. Batch of panel-layers or panels arrives at a station ⟶ one panel-layer or panel is processed at a time ⟶ batch of panel-layers or panels leaves station.

4. Batch of panel-layers or panels arrives at a station ⟶ one panel-layer or panel is processed at a time ⟶ one panel-layer or panel leaves station.

5. One panel-layer or panel arrives at a stations ⟶ is processed and waits ⟶ batch of panel-layers or panels leave station.

6. 2 batches of panel-layers arrive at a station ⟶ capacitor is embedded and the two are combined into one batch ⟶ one batch of panel-layers leaves station.

7. Set of l panel-layers arrives at a station ⟶ are combined ⟶ one complete panel leaves the station.

8. Batch of n completed panels arrives at a station ⟶ is processed ⟶ batch of nb boards leaves station.

   OR

   One completed panel arrives at a station ⟶ is processed ⟶ batch of b boards leaves station.

9. Batch of boards arrives at a station ⟶ components are mounted on surface of each board ⟶ individual boards leave station.

The movement scenarios in this subsection further reinforce the need for the factors introduced in the previous subsection. Combining panel-layers into panels, separating boards from panels, processing panel-layers in some operations, panels in others and boards in yet others with panel-layers arriving at and panels leaving some stations, panels arriving at and boards leaving other stations and panels and panel-layers arriving at a station and being processed together, all require that the batch sizes and mean processing time calculations be adjusted so as to accommodate the effects of these batch dynamics.

4.3 Design

4.3.1 Calculating the Number of Layers

Sandborn *et al.* [111] present a model to find the number of layers needed for a PCB incorporating embedded passives. This research modifies the model presented there and the example in Section 4.4 uses the model developed here to calculate the number of layers for a PCB with embedded passives. For the assumptions and detailed component models please refer to Sandborn *et al.* [111]. The model uses the following notation:

N_l^n = number of layers for new board (with embedded passives)

N_l^c = number of layers for conventional board (no embedded passives)

W_l^n = wiring per layer for new board

W_l^c = wiring per layer for conventional board

W_u^n = total length of wiring used for the new implementation

W_u^c = total length of wiring used for the conventional implementation

W_b = total wiring blocked from embedding passives into the board

W_a^c = total length of wiring theoretically available on conventional board

U_c = fraction of wiring used to route the conventional board

U_l = maximum fraction of available wiring actually used

$U = \dfrac{U_c}{U_l} \approx 1$

(assuming that the conventional implementation has effectively used up all the available wiring)

N^c_{IO} = total number of IO in the conventional implementation

N^n_{IO} = total number of IO for PCB with embedded passives

N_R = number of integral resistors

N_{BC} = number of bypass capacitors integrated

A_{R_i} = area occupied by embedded resistor, i

A^n = new board area

A^c = conventional board area

W_r = ratio of wiring needed per layer for new board to that on the conventional one

The total wiring used for a board depends on the total wiring available and the fraction of the total that can be used to route the board (assuming that the outer layers do not have any wiring),

$$\begin{aligned} W^c_u &= W^c_a U_c \qquad (4.1) \\ &= (W^c_l (N^c_l - 2)) U_c \qquad (4.2) \\ W^n_u &= f W^c_u \qquad (4.3) \end{aligned}$$

(assuming that there is no wiring on the reference planes) where,

$$f = \frac{N^c_{IO} - 2N_R - 2N_{BC}}{N^c_{IO}} \tag{4.4}$$

Now,

$$N^c_{IO} - 2N_R - 2N_{BC} = N^n_{IO}$$

Hence,

$$W_u^n = \frac{N_{IO}^n}{N_{IO}^c} W_u^c \tag{4.5}$$

$$\therefore W_u^n = \frac{N_{IO}^n}{N_{IO}^c} W_l^c (N_l^c - 2) U_c \tag{4.6}$$

The total wiring blocked (and hence unavailable for routing) due to the embedded passives is given by,

$$W_b = \left(\frac{\displaystyle\sum_{i=1}^{N_R} A_{R_i}}{A^n} \right) \left(\frac{A^n}{A^c} \right) W_l^c \tag{4.7}$$

$$= \frac{\displaystyle\sum_{i=1}^{N_R} A_{R_i}}{A^c} W_l^c \tag{4.8}$$

The number of layers for the new board (with embedded passives) may be written as,

$$N_l^n = \frac{W_u^n + W_b}{(W_l^n) U_l} \tag{4.9}$$

$$\therefore N_l^n = \frac{\left(\frac{N_{IO}^n}{N_{IO}^c} \right) W_l^c (N_l^c - 2) U_c + \frac{\displaystyle\sum_{i=1}^{N_R} A_{R_i}}{A^c} W_l^c}{W_l^n U_l} \tag{4.10}$$

But, assuming $U \approx U_l$,

$$\therefore N_l^n = \left(\frac{W_l^c}{W_l^n}\right)\left[f(N_l^c - 2) + \frac{\sum_{i=1}^{N_R} A_{R_i}}{A^c U}\right] \tag{4.11}$$

$$= W_r \left[f(N_l^c - 2) + \frac{\sum_{i=1}^{N_R} A_{R_i}}{A^c U}\right] \tag{4.12}$$

The values of the Usage may be set to a value between 0 and 1 since the actual area available for wiring is less than the total board area.

Using the electrical properties of the circuit and the physical properties of the embedded passive, the designer can now calculate the number of layers needed for the PCB.

4.3.2 Design Parameters

Figure 4.2 shows differentially defined layers forming separate patterns of conductors (copper members) and resistors formed using subtractive PWB print and etch techniques.

The design parameters for a PCB with and without embedded passives are as follows:

1. Total number of resistors
2. Total number of capacitors
3. Size of the panel

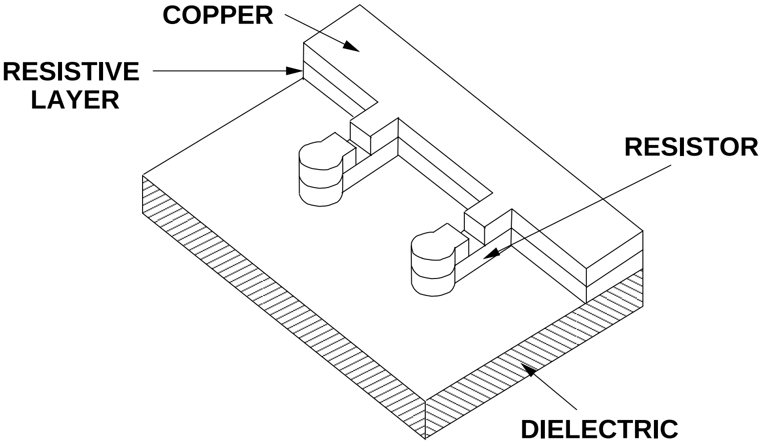

Figure 4.2: Embedded Resistor [101]

4. Size of the board
5. Total wiring requirements
6. Number of layers
7. Number of embedded resistors
8. Number of embedded capacitors
9. Thickness of the board
10. Number of discrete resistors
11. Number of discrete capacitors

Some design parameters listed above depend on other design parameters. The size of the board is defined by the number of embedded and discrete passives and total wiring requirements. The total wiring requirements are governed by the number of embedded and discrete passive components in the PCB. As was shown in Subsection 4.3.1, the total number of layers in the PCB depends on the size of the board, the number of embedded and discrete resistors and bypass capacitors. The decision on the number of passives to embed is taken keeping under consideration the total number of passive components on the PCB, the resultant size of the board and the number of layers in the resulting PCB design. The thickness of the board is directly influenced by the number of layers in it. The number of discrete resistors and capacitors to be soldered atop the PCB depends on the total number of passives in the design of the PCB and the quantity to be embedded.

4.4 Example

This section analyzes the effects of designing a PCB with varying percentages of embedded passives on the product-manufacturing system relationship using an AS900 CPU printed circuit board as an example. The information about the product and the system is based on experiences and information with electronic systems manufacturers. The examples use data that collaborators were able to provide and other synthetic data that was created as part of this research effort.

The tool employed for analyzing the manufacturing system performance as it processes the PCBs is based on the models presented in Chapter 3. The tool is similar to the DFP tool presented in Chapter 3 with the following differences:

1. Since this tool aims to understand the performance of the manufacturing system as the number of embedded passives in the PCB changes, for each run of the tool, it only uses the specified throughput for each product. The tool does not analyze the performance of the system while progressively increasing the new product throughput as was the case with the DFP tool in Chapter 3.

2. Since the new product throughput is no longer the independent variable, this

tool does not plot the product and processing system performance metrics as a function of the new product throughput.

4.4.1 Assumptions

Throughout this section, all examples are explained and analyzed under the following set of assumptions:

1. Until the kitting stage, each layer of the PCB is treated as an individual product. The layers travel in batches through the system.

2. Until the stage where the separation of individual boards occurs, the board travels as part of a panel (which may contain more than one board depending on the size of the board). The board size is always smaller than the panel size.

3. Each discrete resistor and capacitor has two leads. This in turn means two holes are needed for assembly of a discrete passive on the PCB.

4. Once the layers are laminated together, they travel as part of a panel. Each board also travels as part of a panel from the lamination stage to the separation stage. Such panels travel in batches through the system.

5. After the separation stage, the boards travel in batches through the system. The number of boards in a batch may be less than, equal to, or greater than the number of boards that are separated from one panel.

These changes in material movement units are explained in more detail in Section 4.2. These changes are offset using factors, F_{bp} and F_{pl}, as explained in the next subsection.

4.4.2 Processing Steps and Processing Time Models

This example assumes that the PCBs visit the sequence of workstations listed below, based on some information from PCB manufacturers as well as personal discussions [113]. Workstations from this set combine to perform the processing steps listed in Section 4.1.2.

The processing times for the PCB at various stations are most often affected by the area of the PCB, the area of the panel and the number of layers. Based on discussions with experts on electronic package manufacturing [113], this work identified that nearly all steps in the processing sequence are affected by the board area or the number of boards that can be made from one panel which is, in turn, determined by the board area (the panel area being a constant).

In addition, the resist application, artwork, develop-etch-strip, kitting, lamination processes as also those associated with the embedding procedure depend on the number of layers in the PCB. The time required for drilling depends on the number of holes necessary which in turn depends on the number of discrete components. The time for the assembly operations to mount these discretes also depends on their number. The mean processing time for lamination depends on the time taken for the intermediate prepreg layers to melt. This in turn is a function of the heat coefficient for the prepreg material. Based on literature on PCB manufacture, the lamination process is analyzed using the lumped parameter approach for two dimensional heat conduction. The total time needed to separate all boards on a panel depends on the time taken to separate each board. The processing times at stations needed for embedding depend, in addition to the factors mentioned here, on the number of passives to be embedded and the number of additional layers in the PCB as a result of the embedded passives. The design parameters and constants for the product and the manufacturing parameters are:

Notation

l_b = length of board

b_b = breadth of board

m_b = number of sides (for mounting components)

n_{dR} = number of discrete resistors

n_{dC} = number of discrete capacitors

n_{bC} = number of bypass capacitors

n_{a_i} = number of active components of type i

$\mathcal{A}$ = set of active component types

n_{lb} = number of layers for board

d_b = spacing between boards

a_p = area of panel

s_f = feature size of embedded resistor

T_l = layer thickness

T_p = prepreg thickness

θ_c = board material heat transfer coefficient

r_d = drill feed rate

t_c = time to cut board

h_i = number of holes required by a component of type i

$$
\begin{aligned}
t_{d_a} &= \text{time for assembly (discrete)} \\
t_{b_a} &= \text{time for assembly (bypass)}
\end{aligned}
$$

The following equations are used to calculate intermediate quantities, which are functions of the product design and manufacturing parameters.

$$
\begin{aligned}
T_b &= n_{lb}T_l + (n_{lb} - 1)T_p && (4.13) \\
n_h &= 2(n_{dR} + n_{dC}) + \sum_{i \in \mathcal{A}} n_{a_i} h_i && (4.14) \\
a_b &= l_b b_b && (4.15) \\
n_{bp} &= \frac{a_p}{(a_b + d_b(l_b + b_b))}, \; a_b \ll a_p && (4.16) \\
F_{bp} &= \frac{1}{n_{bp}} && (4.17) \\
F_{pl} &= n_{lb}F_{bp} && (4.18) \\
A_R &= s_f^2 N_R && (4.19)
\end{aligned}
$$

where

$$
\begin{aligned}
T_b &= \text{board thickness} \\
n_h &= \text{number of holes in board} \\
a_b &= \text{board area} \\
n_{bp} &= \text{boards per panel} \\
F_{bp} &= \text{factor: board-panel} \\
F_{pl} &= \text{factor: panel-layer}
\end{aligned}
$$

$$\begin{aligned} n_{lb} &= N_l^c \text{ if the PCB is a conventional board,} \\ &= N_l^n \text{ if it has embedded passives} \end{aligned}$$

For this example, the following is the set of workstations that a PCB visits as part of its processing requirements. Figure 4.1 presented a high-level abstraction of the processing sequence for a PCB. On the other hand, Tables 4.1 to 4.16 which included the various substeps corresponding to each of the operation blocks in Figure 4.1 presented a very detailed low-level view of the processing requirements. The sequence of workstations which the products in this example visit, is an intermediate-level processing model which attempts to capture the significant processing time contributing operations. The entities in parentheses next to each workstation name indicate the block(s) in Figure 4.1 whose operation is performed at that workstation. The mean processing times at various workstations are calculated using the design parameters listed above.

1. Resist Application to internal layers (Block IA, IB),

$$t_{RA_l} = 0.01 F_{pl} a_p \tag{4.20}$$

2. Artwork for internal layers (Block IC),

$$t_{A_l} = 4 F_{pl} \tag{4.21}$$

3. DES for internal layers (Block ID, IE),

$$t_{DES_l} = 0.008 F_{pl} a_p \tag{4.22}$$

4. Test for internal layers (Block IF),

$$t_{T_l} = 0.8F_{pl} \tag{4.23}$$

5. Clean and desmear layer (Block A),

$$t_{DS_l} = 2 + 0.4F_{pl} \tag{4.24}$$

6. Apply photoresist polymer (Block C),

$$t_{APR_{EP}} = 0.001F_{pl}a_p \tag{4.25}$$

7. Expose polymer to create vias (Block C),

$$t_{EPR_{EP}} = 10 \tag{4.26}$$

8. Cure dielectric polymer (Block C),

$$t_{CP_{EP}} = 10 \tag{4.27}$$

9. Develop photoresist (Block C),

$$t_{DES_{EP}} = 0.002F_{pl}a_p \tag{4.28}$$

10. Print conductor protection pattern (Block C),

$$t_{PP_{EP}} = 0.001F_{pl}a_p \tag{4.29}$$

11. Electroless plate resistive layer (Block C),

$$t_{EPR_{EP}} = 7 + 0.002F_{pl}a_p \tag{4.30}$$

12. Etch copper remove excess strip photoresist (Block C),

$$t_{ES_{EP}} = 10 + 0.002F_{pl}a_p \tag{4.31}$$

13. Measure values, test and clean (Block C),

$$t_{MT_{EP}} = 10 \tag{4.32}$$

14. Plating via holes in layer (Block D),

$$t_{PV_l} = 0.02F_{pl} \tag{4.33}$$

15. Kitting and Layup (Block D),

$$t_{KL} = 0.4F_{pl}nl_b \tag{4.34}$$

16. Lamination (Block E),

$$t_L = 10 + F_{bp}\theta_c T_b \tag{4.35}$$

17. Drilling holes for pin hole through components (Block F),

$$t_D = 5 + F_{bp}\left(\frac{T_b}{r_d}\right)n_h \tag{4.36}$$

18. Desmear Board (Block G),

$$t_{DS} = 2 + 20F_{bp} \tag{4.37}$$

19. Electrolysis for board [for plating] (Block H),

$$t_P = 7 + 10F_{bp} \tag{4.38}$$

20. Apply Resist to board (Block I),

$$t_{RA} = 0.2F_{bp}a_p \tag{4.39}$$

21. Artwork/Exposure of board (Block J),

$$t_A = 4F_{bp} \tag{4.40}$$

22. DES board (Block J),

$$t_{DES} = 5 + 20F_{bp} \tag{4.41}$$

23. Inspect board (Block K),

$$t_I = 10 + 20F_{bp} \tag{4.42}$$

24. Solder Level (Block L),

$$t_S = 0.1F_{bp}a_p \tag{4.43}$$

25. DES for Solder (Block M),

$$t_{DES_s} = 0.2F_{bp}a_p \tag{4.44}$$

26. Final Routing Check for board (Block N),

$$t_{RC} = 5 + 20F_{bp} \tag{4.45}$$

27. Final Inspection for board (Block N),

$$t_{I_f} = 2 + 2F_{bp}n_{bp} \tag{4.46}$$

28. Separate Boards from panel,

$$t_C = 2 + F_{bp}t_c \tag{4.47}$$

29. Print Solder Paste - side 1,

$$t_{PS_1} = 0.05a_b \tag{4.48}$$

30. Place Discretes on side1,

$$t_{a_1} = \frac{n_{dR} + n_{dC}}{100} \tag{4.49}$$

31. Inspect Assembly,

$$t_{I_1} = 1.2\left(\frac{n_{dR} + n_{dC}}{100}\right) \tag{4.50}$$

32. Print Solder Paste - side 2,

$$t_{PS_2} = 0.05a_b \tag{4.51}$$

33. Place Discretes on side 2,

$$t_{a_2} = \frac{n_{dR} + n_{dC}}{100} \tag{4.52}$$

34. Reflow Solder Connects,

$$t_{I_2} = 1.2\left(\frac{n_{dR} + n_{dC}}{100}\right) \tag{4.53}$$

35. Final Assembly,

$$t_{a_f} = (n_{dR} + n_{dC})t_{d_a} + n_{bC}t_{b_a} \tag{4.54}$$

36. System Test,

$$t_{T_S} = 10 \tag{4.55}$$

PCBs with embedded passive components follow this sequence of workstations. If the PCB has no embedded passives, the workstations in the above list corresponding to Block C should be removed.

4.4.3 Manufacturing System

The manufacturing facility for the product is a PCB fabrication and assembly shop having various types of batch processes, individual part processes, manual inspection, and automated inspection stations. The material handling between stations, which is often automated in the form of conveyor belts, does not contribute significantly to the manufacturing cycle time.

4.4.4 Products

The printed circuit board under consideration is a multilayer board with surface mount components and pin through hole mounted components on both sides of the board. The manufacturer currently makes one type of PCB, called *CPU board*, which serves as the central processing unit for the AS900 controller. Due to associated advantages such as reduced costs, improved electrical performance, lower manufacturing and assembly defects, and a heavy utilization of present workstations, the manufacturer has opted to embed a portion of the passives devices that are currently either surface mounted or pin through hole. The remaining passive devices and the active ones continue to be surface mounted or pin through hole as was the case with the earlier board. Though the company wishes to change the product in this fashion, it needs to continue producing the older product in order to satisfy the requirements of current users, till such time as a total shift to the new product is made.

Therefore, in the new scenario, the facility will make two kinds of products, *CPU board* and *CPU board new*. *CPU board new* contains the embedded passive components. The facility does not manufacture any other product. (It needs to be emphasized, however, that any other products that the facility manufactures can be easily incorporated in the DFP approach in a similar manner.)

The factory currently has a throughput of 5 units/hour of *CPU board*. Over the same production horizon, the factory in the new scenario needs to manufacture some *CPU board* product parts and some *CPU board new* product parts, the total throughput always being 5 units/hour.

The circuit for the CPU that currently defines *CPU board* has key design characteristics as detailed in Table 4.17. System performance trials are conducted while embedding different percentages of discretes in the PCB. The boards are cut out of panels 24 inches long and 18 inches wide. Spacing between boards on a panel is 0.15 inches while the length and breadth of the discrete components are 0.04 and 0.02 inches respectively. The active devices as well as the discrete passives components are pin through hole mounted on both sides of the PCB. Of the non-passive devices in the product, the diodes, zeners, and inductors require two holes, the transistors and transformers require three holes while the network parts and ICs require 50 holes.

Each embedded passive has a maximum feature size of 0.015 inches and the PCB has maximum 6361 IOs, with a wiring ratio of 1.1. For detailed explanations of various terms, please refer to Sandborn *et al.* [111].

No.	Design Parameter	
1.	Length of board	18 inches
2.	Breadth of board	12 inches
3.	Discrete Resistors	627
4.	Discrete Capacitors	54
5.	Bypass Capacitors	53
6.	Network parts	71
7.	Diodes	53
8.	Zeners	17
9.	Transistors	64
10.	Inductors	28
11.	Transformers	12
12.	ICs	108
13.	Number of Layers	12
14.	Number of Sides	2

Table 4.17: PCB Design Features

4.4.5 Experiment Design

To find the effect of embedding passive components into the PCB substrate on manufacturing system performance, the designed experiment analyzes one product and two product scenarios starting with 10% embedded passives for *CPU board new* and then increasing the percentage of embedded passives to 20%, 40%, 60% and 80%, while adjusting the percentage of discretes accordingly. The DFP tool calculates product and system parameters for each product design. Next, the experiment implements different product mixes by changing the constituent product percentages while maintaining the same aggregate, and documents the product and system parameters for each percentage of embedded passives. The following 2-tuples represent the combinations of rates of manufacture of *CPU board new* and *CPU board* for the four product mixes considered: {0.5,4.5}, {1.5,3.5}, {2.5,2.5} and {3,2}, units/hour respectively. Further, the experiment analyzes the effects of different order release rates on system performance by setting different batch sizes of the two products for each *CPU board new* design. With 5 possible product design, 4 possible product mix and 3 order release options, the experiment obtains a data set comprising 60 elements.

The experiment presents results for each set of trial data in the form of station utilizations, station cycle times and product cycle times.

4.4.6 Results and Discussion

As the percentage of passives in *CPU board new* that are embedded in the substrate changes, the number of layers in the multilayer PCB changes. In conjunction, the processing times for the PCB at the various stations in its processing system change. This in turn changes the average manufacturing cycle time at each station in the product's processing sequence. The overall result of these modifications is that the manufacturing cycle time for the new product is different from that of the existing product, *CPU board*. The design changes arising from embedding passives present an option to change the size of the board. This example however, maintains the same size for the board while calculating the number of layers for *CPU board new*.

Table 4.18 contains the cycle times for the product, when the system is only processing *CPU board*. The system processes 5 units/hour of the product in each production horizon. Table 4.19 catalogs the product cycle times when both products are being processed.

Table 4.20 compares the product cycle times for the two product designs where 10% and 80% of the passive components are embedded.

For the scenario outlined above, modifying the product design for *CPU board new* by increasing the percentage of embedded passive components results in a

Product Name	Batch Size	Cycle Time (min)
CPU board	5	875.45
	10	1715.65
	20	3396.07

Table 4.18: Product cycle times (without new product)

Product Name	Batch Size	Throughput units/hour		Cycle Times for Percentage Embeddeds (min)				
		Old	New	10%	20%	40%	60%	80%
CPU board new	5	4.5	0.5	1085.74	1081.76	1074.03	1066.61	1059.36
		3.5	1.5	1082.33	1073.11	1055.88	1039.96	1024.87
		2.5	2.5	1139.55	1115.63	1077.53	1047.12	1020.88
		2.0	3.0	1622.03	1406.41	1275.81	1217.00	1177.21
	10	4.5	0.5	2135.67	2127.95	2112.94	2098.48	2084.30
		3.5	1.5	2124.48	2107.01	2074.13	2043.57	2014.42
		2.5	2.5	2218.23	2174.62	2104.27	2047.28	1997.52
		2.0	3.0	3056.90	2680.11	2448.35	2341.22	2267.08
	20	4.5	0.5	4235.53	4220.34	4190.77	4162.22	4134.18
		3.5	1.5	4208.79	4174.82	4110.66	4050.79	3993.52
		2.5	2.5	4375.60	4292.61	4157.76	4047.62	3950.79
		2.0	3.0	5926.70	5227.57	4793.49	4589.70	4446.87
CPU board	5	4.5	0.5	866.92	864.63	860.27	856.19	852.29
		3.5	1.5	858.53	851.03	837.18	824.64	812.94
		2.5	2.5	892.59	870.46	835.89	808.97	786.22
		2.0	3.0	1214.40	1000.83	874.19	819.13	782.99
	10	4.5	0.5	1698.15	1693.82	1685.53	1677.75	1670.29
		3.5	1.5	1677.98	1663.93	1637.84	1614.01	1591.63
		2.5	2.5	1730.79	1690.75	1627.45	1577.42	1534.61
		2.0	3.0	2285.70	1913.13	1689.50	1590.12	1523.51
	20	4.5	0.5	3360.62	3352.20	3336.07	3320.88	3306.28
		3.5	1.5	3316.88	3289.75	3239.15	3192.75	3149.02
		2.5	2.5	3407.20	3331.34	3210.57	3114.33	3031.41
		2.0	3.0	4428.32	3737.73	3320.13	3132.09	3004.56

Table 4.19: Product cycle times (with new product)

progressive decrease in the manufacturing cycle times for *CPU board new* as well as *CPU board*. When the requirements of *CPU board* per production horizon are

Product Name	Batch Size	Throughput units/hour		Percent Change in Cycle Time from 10% Embeddeds to 80% Embeddeds
		Old	New	
CPU board new	5	4.5	0.5	-2.43
		3.5	1.5	-5.31
		2.5	2.5	-10.41
		2.0	3.0	-27.42
	10	4.5	0.5	-2.41
		3.5	1.5	-5.18
		2.5	2.5	-9.95
		2.0	3.0	-25.84
	20	4.5	0.5	-2.39
		3.5	1.5	-5.11
		2.5	2.5	-9.71
		2.0	3.0	-24.97
CPU board	5	4.5	0.5	-1.69
		3.5	1.5	-5.31
		2.5	2.5	-11.92
		2.0	3.0	-35.52
	10	4.5	0.5	-1.64
		3.5	1.5	-5.15
		2.5	2.5	-11.33
		2.0	3.0	-33.35
	20	4.5	0.5	-1.62
		3.5	1.5	-5.06
		2.5	2.5	-11.03
		2.0	3.0	-32.15

Table 4.20: Product cycle time comparison

decreased from 5 units/hour to 4.5 units/hour with 10% of the passives embedded for *CPU board new*, the manufacturing cycle time for *CPU board* reduces as expected. However for PCB designs with 10% and 20% passives embedded, as the contribution of *CPU board* to the product mix decreases, the manufacturing cycle time for *CPU board* progressively decreases, before beginning to increase, at a point exceeding the value when the system was only processing *CPU board*. This behavior is not observed for the other product designs of *CPU board new*, and for these designs, the manufacturing cycle time for *CPU board* progressively reduces. The manufacturing cycle time for *CPU board new* initially reduces as the contribution of *CPU board new* to the overall product mix is increased, before beginning to increase. This behavior is observed for all product designs.

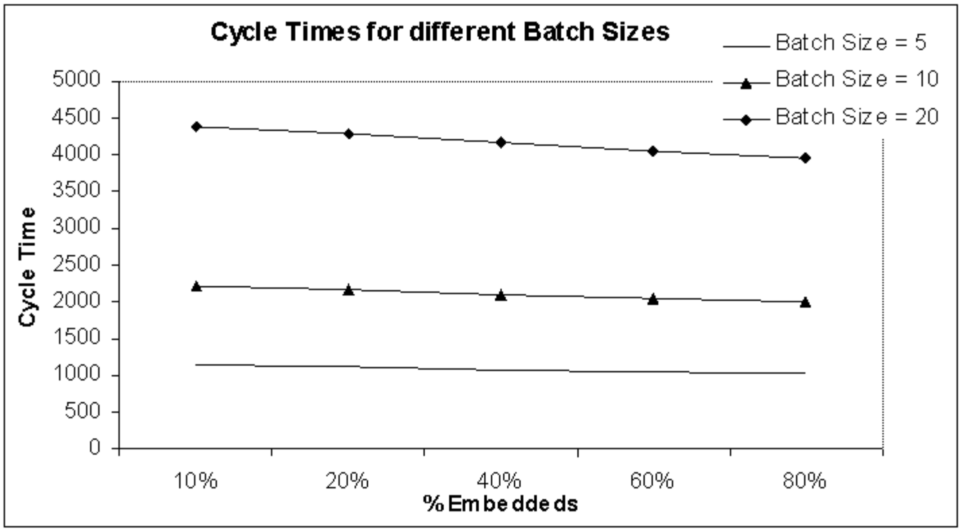

Figure 4.3: Manufacturing cycle time with increasing percentage of embeddeds

Table 4.20 shows the percentage reduction in manufacturing cycle time for *CPU board new* and *CPU board* as the product design changes. For each order release, as the contribution of *CPU board new* to the product mix increases, the percentage reduction in manufacturing cycle time from products with 10% embeddeds to those with 80% embeddeds increases. This may be attributed to reduced utilizations for the component assembly stations due to the modified product design. Figure 4.3 shows the plot of the manufacturing cycle time for *CPU board new* as the percentage of embedded passives increases from 10% to 80%. The plot compares the values for different order releases when the product mix comprises 50% each of products *CPU board* and *CPU board new*. The reduction in the manufacturing cycle time for *CPU board new* as the number of embedded passives increases is larger for a batch size of 20 than for a batch size of 5.

The number of embedded passives also affects the impact of changing the product mix. In Table 4.19, when the batch size is 20, the increase in manufacturing cycle time for *CPU board new* from when the system needs a throughput of 2.5 units/hour of *CPU board new* to when it needs a throughput of 3.5 units/hour is comparatively much larger with 10% passives embedded (35.45%) than with 80% passives embedded (12.56%). This is because the utilization of the *DES for Inner Layer* station increases from 0.96 to 0.99 for 10% embedded passives as opposed to an increase from 0.91 to 0.94 for 80% embedded passives. As a consequence, the manufacturing cycle time for the station increases from 875 min to 1644.5 min when *CPU board new* has 10% embedded passives.

Table 4.21 shows the utilizations for a sample 6 resources from the manufacturing system. As seen in the table, the utilizations decrease as the number of embedded passives increases. The results are for the case when the batch size for the two products is 20, and the system makes 0 units/hour, 0.5 units/hour and 3 units/hour of *CPU board new* respectively.

Design improvement suggestions. The tool generates design suggestions for the PCB design team for some product designs. Notably, for the board design with 10% embedded components, the tool suggests modifying the panel-to-layer factor. Since the panel-to-layer factor is a measure of the size of the PCB and the number of layers in the PCB (it is defined as the ratio of the number of layers in the PCB to the number of boards that can be made from a single panel), reducing the panel-to-layer would require either reducing the number of layers in a panel or increasing the number of boards that can be cut from a panel i.e. reducing the size of the board. The tool presents this suggestion based on the high utilization

%Embedded	Workstation Name					
	Resist Apply IL	Artwork IL	DES IL	Drill	Inspect Assy	Final Assy
0	0.5020	0.5888	0.8070	0.7560	0.3169	0.3722
10	0.5240	0.6138	0.8379	0.7564	0.3137	0.3685
20	0.5222	0.6120	0.8367	0.7517	0.3106	0.3647
40	0.5188	0.6086	0.8342	0.7429	0.3042	0.3573
60	0.5154	0.6052	0.8318	0.7348	0.2979	0.3498
80	0.5120	0.6017	0.8294	0.7274	0.2916	0.3424
10	0.6339	0.7386	0.9922	0.7582	0.2979	0.3500
20	0.6236	0.7282	0.9849	0.7301	0.2789	0.3273
40	0.6032	0.7076	0.9703	0.6770	0.2407	0.2828
60	0.5828	0.6871	0.9559	0.6288	0.2030	0.2382
80	0.5624	0.6665	0.9413	0.5841	0.1647	0.1937

Table 4.21: Resource Utilization

of the *Measure values, test and clean* (0.964) and *DES for internal layers* (0.992) workstations. Note that the mean processing time for *Measure values, test and clean* is a fixed value independent of the product design (Equation 4.32) and hence the only option to reduce the utilization of this station is to add more resources to the workstation.

As the percentage of embedded passives increases to 80%, the system does not advise the designer to change any particular aspect of the design for any product mix. As was explained in Chapter 3, the heuristics programmed into the current tool for design analysis are geared more towards avoiding over-utilization of any portion of the manufacturing system. For the board design with higher percentage of embedded passives, no station becomes very highly utilized and hence the tool does not identify any need for product design modifications. The reason for these moderate station utilization values may be attributive to a reduced layer-to-panel factor owing to a reduction in the number of discrete passives and the increase in the number of embedded bypass capacitors.

4.5 Summary

This chapter presented a DFP tool to analyze printed circuit boards with embedded passives. The tool provides the product development team the capability to analyze PCB designs with varying percentages of embedded passives with respect to their performance in the given manufacturing system. The chapter further presented an example of application of the PCB-DFP tool to model an AS900 CPU board with a large set of active and passive devices.

The tool calculates the utilizations and cycle times for all 28 workstations that the product visits when there are no embedded passives and 36 workstations when passives are embedded. The values of these parameters are calculated for all product designs. Different product designs are obtained with different percentages of embedded passives (from 10% to 80%). The product cycle times are found to decrease as the percentage of embedded passive components increases. The utilization and hence cycle times of the stations in the processing sequence also decrease. The tool generates redesign suggestions depending on the manufacturing system performance parameters for some product designs.

Chapter 5

Modeling Process Drift

In many manufacturing systems, a station processes a batch of parts. A portion of the batch will become defective due to the variability of the process. These bad parts must be detected and discarded. The bad parts may be detected at the current station or at an inspection station that the parts visit at a later step in the processing sequence. This chapter analyzes the case where some bad parts are detected and discarded at the current station while other bad parts are detected and discarded at the next inspection station. It tries to understand the relationship between manufacturing cycle time and product yield. Towards that end, this chapter presents an enhanced model for estimating the manufacturing cycle time for a set of products being processed in a given manufacturing system, one that incorporates the effects of various yield losses.

The first half of this chapter explains process drift and develops models for manufacturing systems with process drift. Section 5.4 presents algorithms for applying these models to flow shop manufacturing systems and Section 5.5 presents examples of such flow shop systems along with comparisons with simulation models. The latter half of this chapter details the difficulties in applying the flow shop algorithm to the general job shop manufacturing system. It presents algorithms to analyze a subset of the general job shop manufacturing scenario set.

5.1 Manufacturing Cycle Time and Yield

Process drift is a common occurrence in many manufacturing processes where machines become dirty (leading to more contamination) or other aspects change, leading to degraded performance. Statistical process control tracks process quality to determine when the process has gone out of control (has drifted beyond its specifications). This research uses the term *scrap yield* (y_s) to describe the fraction of parts that do not have obvious flaws. This is also referred to as the *pass fraction* P [112], modeled as $P = Y_{in}^{f_c}$, where Y_{in} is the yield of the parts entering the step and f_c is the fault coverage of the step.

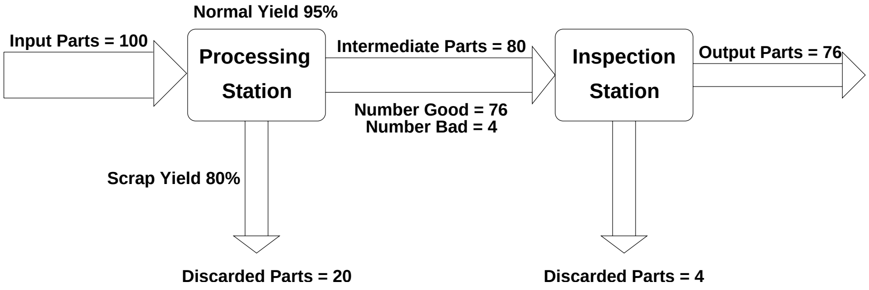

Figure 5.1: Normal yield conditions at the inspection station

These parts which do not have obvious flaws continue to the next step. However, some of these parts have undetected flaws (which are found at a subsequent inspection station). Moreover, the size of the fraction with undetected flaws depends upon whether and how long the process has been operating within specifications. *Normal yield* (y_n) condition exists when the process is within specifications (see Figure 5.1). The *reduced yield* (y_r) (which is lower than the normal yield) occurs when the process behavior has drifted beyond its specifications. (see Figure 5.2).

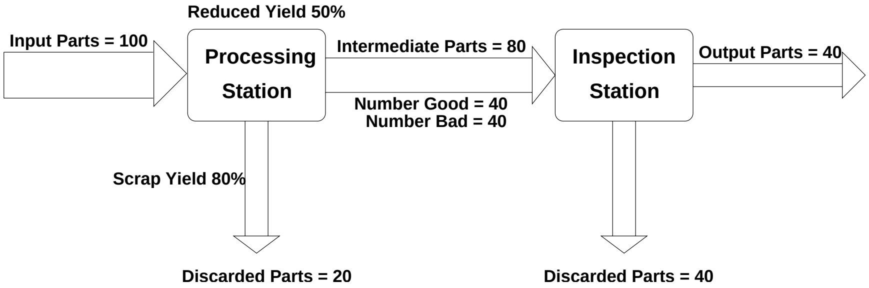

Figure 5.2: Reduced yield conditions at the inspection station

Figures 5.3 and 5.4 show the process equivalent for the blocks that form a part of the processing and inspection stations respectively in a product's processing sequence.

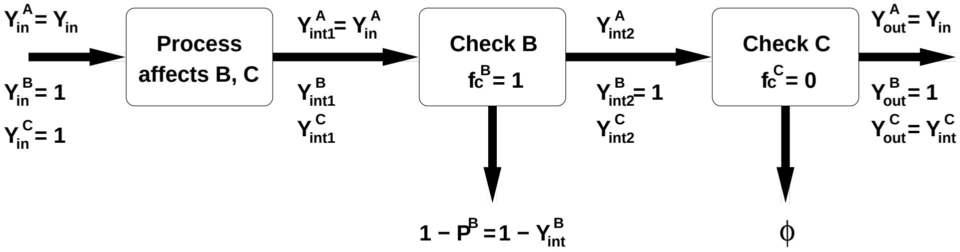

Figure 5.3: Processing station block diagram internals

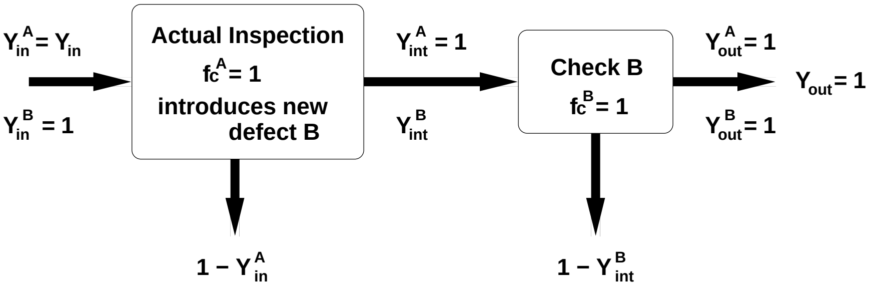

Figure 5.4: Inspection station block diagram internals

Consider that a part entering the station has three types of features, *A, B, C.*

A, B, C are independent characteristics for the product design. The processing station affects only features B and C, without affecting feature A. The input yield at the processing station, $Y_{in} = Y_{in}^{A}$.

$$\begin{aligned} P^{B} &= (Y_{int}^{B})^{f_c^B} = Y_{int}^{B} = y_s && (5.1) \\ P^{C} &= (Y_{int}^{C})^{f_c^C} = 1 && (5.2) \\ Y_{out}^{A} &= Y_{int2}^{A} = Y_{int1}^{A} = Y_{in}^{A} = Y_{in} && (5.3) \\ Y_{out}^{B} &= Y_{int2}^{B} = (Y_{int1}^{B})^{1-f_c^B} = 1 && (5.4) \\ Y_{out}^{C} &= Y_{int2}^{C} = Y_{int1}^{C} = y_n && (5.5) \end{aligned}$$

Therefore,

$$\begin{aligned} Y_{out} &= Y_{out}^{A} Y_{out}^{B} Y_{out}^{C} && (5.6) \\ &= Y_{in} \times 1 \times y_n && (5.7) \end{aligned}$$

Thus, the scrap and normal yields are independent of each other. Process drift conditions mean that Y_{int}^{C} has decreased.

The discussion thus far has focussed on the processing station internals shown in Figure 5.3. A similar explanation can be presented for inspection stations internals as shown in Figure 5.4. At the inspection station, the inspection procedure may be said to potentially introduce an error in feature B of the product which is then checked for and the defective parts discarded. The pass fractions corresponding to the original defect that the inspection station looks for (feature A) and (feature B) are both equal to one and hence the output yield from the station is also equal to one.

For modeling purposes, this work assumes that a process goes out of control at a frequency that can be expressed as a drift rate. When it is in this state, the fraction of parts with undetected flaws equals the reduced yield. The time that the process remains out of control depends upon how long it takes a batch of parts to move from that station to the inspection station. This time is called the detection time. Clearly, a larger detection time implies that the process will operate out of control for a longer period of time, which reduces the throughput (the number of usable parts produced). The detection time depends on the position of the inspection station in the processing sequence and the manufacturing cycle time at the stations that follow the process that is out of control. Since detecting a process drift depends on the time that elapses before the first defective product arrives at the subsequent inspection station, the manufacturing cycle times for the stations in the processing sequence for the product from the processing station where the drift occurs and the next inspection station contribute to the detection time for the process drift.

5.2 Manufacturing System Model

Chapter 3 presented a model for calculating the manufacturing cycle time and throughput for a product. The model presented in Chapter 3 assumed a fixed yield for each workstation in the system (scrap yield) attributive to a portion of products found defective at departure from the station. The enhanced model presented here has a similar basis, however, in addition to the fixed scrap yield, the model incorporates effects of process drift and delay in detecting the drift. This detection time and associated delay are functions of the manufacturing cycle time at processing stations in the system. This model similarly assumes that no product visits a processing station more than once. An underlying assumption in developing these models is that when a process defect occurring at a processing station is detected at the nearest inspection station, all possible process defects at that station are fixed immediately[1].

Data Requirements. The manufacturing system model requires the following data: For each workstation, the number of resources available, the rate at which process drift is likely to occur, and the mean time to failure and mean time to repair a resource; For each product the job size (number of parts) at point of entry into the processing system and the release rate (number of parts per hour of factory operation), the sequence of workstations that each job must visit, the mean setup time (per job) at each workstation and its variance, the mean processing time (per part) at each workstation and its variance, the scrap, normal and reduced yields at each workstation.

The model developed incorporates the effects of three possible sources of error. As a result, three types of yields manifest themselves at a station:

1. A *scrap yield*: this is a constant yield for a processing station arising out of errors in individual parts in a job of a product which are detected at the processing station itself and discarded there.

2. A *normal yield*: this is a constant yield for a processing station arising out of errors at the station, in one or more parts of a job which may only be detected and discarded at the nearest inspection station.

[1] The model assumes the repair time for the process to be negligible.

3. A *reduced yield*: this yield is associated with process drift at a station in the processing sequence for a product. The process drift affects all subsequent jobs of products that are processed at the defaulting station, until the process drift is detected at the nearest inspection station. All such parts are then to be discarded at the inspection station. The reduced yield for a workstation is the average over all the resources at the workstation where process drift occurs. Also, only one resource at the workstation drifts at a time.

Mathematical Notation

$\mathcal{I}$ = the set of all products

B_{i0} = job size of product i at release

c_i^r = SCV of job interarrival times for product i

$\mathcal{J}$ = the set of all processing stations

$\mathcal{F}$ = the set of all inspection stations

n_j = number of resources at station j

m_j^f = mean time to failure for a resource at station j

m_j^r = mean time to repair for a resource at station j

ρ_j = process drift rate for station j

R_i = sequence of stations that product i must visit

$R_i \subset \mathcal{J} \cup \mathcal{F}$

Q_{ij} = subsequence of R_i, that starts with the station that follows j and ends with the next inspection station for $j \in R_i \cap \mathcal{J}$

R_{ij} = subsequence of R_i that starts from the beginning of R_i and ends with j, $\forall j \in R_i$

$H(i, j)$ = station that product i visits immediately before station j

t_{ij} = mean part process time of product i at station j

c_{ij}^t = SCV of the part process time

s_{ij} = mean job setup time of product i at station j

c_{ij}^s = SCV of the setup time

y_{ij}^{s} = scrap yield of product i at station j

y_{ij}^{n} = normal unchecked yield of product i at station j

y_{ij}^{r} = reduced unchecked yield of product i at station j

T_i^a = arrival (release) rate of product i (parts per hour)

The following preliminary observations can be made:

1. Q_{ij} has exactly one element $m \in R_i \cap \mathcal{F}, j = H[i, m]$; it contains all stations in R_i after j up to and including m. Q_{ij} is empty if there is no inspection station in R_i after j. $R_{ij} \cup Q_{ij} = R_{im}$.

2. $H(i, j) = 0$ if j is the first station that product i visits.

3. $H(i, j) \neq j, \ \forall i \in \mathcal{I}, \ j \in R_i$

4. $R_{i0} = \{\}$

5. $R_{ij} = R_{[i,H(i,j)]} \cup \{j\}$

For example, consider a processing sequence for product i with four processing stations followed by an inspection station. For this product and processing sequence,

- $R_i = (1, 2, 3, 4, 5)$

- $R_{i3} = (1, 2, 3)$

- $Q_{i3} = (4, 5)$

- $R_{i3} \cup Q_{i3} = R_i = (1, 2, 3, 4, 5)$

- $H(i, 3) = 2$

- $R_{i3} = R_{i2} \cup \{3\} = (1, 2, 3)$

Because there is no reentrant flow for any product in the system, the following lemmas hold:

Lemma 1 *If $j \in R_i$ and $k \in R_{ij}$, then $j \notin R_{ik}$.*

Lemma 2 *If $k \in Q_{ab}$ and $f \in Q_{ab} \cap \mathcal{F}$, then $k \in R_{af}$, $\forall a \in \mathcal{I}$, $b \in R_a \cap \mathcal{J}$.*

These lemmas find basis in the definitions of Q_{ij} and R_{ij} stated earlier.

Initial Calculations. The desired throughput and the input batch size are used to calculate the release rate for the products in the system.

z_{ij} = average unchecked yield due to drift for product i at station j

Z_{ij} = hidden yield of product i from process drift at output of station j

x_i = arrival rate of product i (jobs per hour)

A_j = availability of a resource at station j

V_j = the set of products that visit station j

B_{ij} = average job size after processing at station j

t^+_{ij} = total process time of product i at station j

c^+_{ij} = SCV of the total process time

t^+_j = aggregate process time at station j

c^+_j = SCV of the aggregate process time

$t_j^\star$ = modified aggregate process time at station j

$c_j^\star$ = SCV of the modified aggregate process time

DT_{ij} = expected delay in detection of a process drift in product i occurring at station j, $j \in R_i \cap \mathcal{J}$

$DT_j^\star$ = expected delay in detection of a process drift at station j, $j \in \mathcal{J}$

u_j = the average resource utilization at station j

$CT_j^\star$ = the average cycle time at station j

CT_i = the average cycle time of jobs of product i

T_i^o = actual throughput of product i at the end of R_i

$$
\begin{aligned}
x_i &= \frac{T_i^a}{B_{i0}} && (5.8)\\
A_j &= \frac{m_j^f}{m_j^f + m_j^r} && (5.9)\\
V_j &= \{i \in \mathcal{I} : j \in R_i\} && (5.10)
\end{aligned}
$$

Process Drift Calculations. A process drift may occur at a processing station and is detected only at the next inspection station in the processing sequence. The time that elapses before detection depends on the manufacturing cycle time at stations in the processing sequence,

$$
\begin{aligned}
DT_{ij} &= \sum_{g \in Q_{ij}} CT^{\star}_{g}; \quad \forall j \in R_i \cap \mathcal{J} && (5.11) \\
DT^{\star}_{j} &= \min_{i \in V_j} \{DT_{ij}\}, \ \forall j \in \mathcal{J} && (5.12)
\end{aligned}
$$

The process drift ρ_j indicates the frequency of unacceptable deviations of the process for station j from nominal processing parameters. This is likely to perpetrate itself among all subsequent products processed at the station.

$\frac{1}{\rho_j}$ is the mean time between the detection and repair of one process drift and the occurrence of the next one. The *hidden yield* at a resource is the time weighted average of the normal and reduced yields at the resource.

$$
\therefore \ z_{ij} = \frac{\frac{1}{\rho_j} y^{n}_{ij} + DT^{\star}_{j} y^{r}_{ij}}{\frac{1}{\rho_j} + DT^{\star}_{j}} \qquad (5.13)
$$

The job size changes as it is processed at successive stations. The average job size at a processing station is influenced by the yields of the preceding operations. The effects of process drifts at various processing stations are translated to the nearest inspection station (in the form of a hidden yield multiplier) and the yield at the inspection station is adjusted accordingly. If there is no process drift at station j, $\rho_j = 0$ and $z_{ij} = y^{n}_{ij}$. Once the effects of process drift for a set of processing stations have been accounted for at the processing station, the value of the hidden yield multiplier is reset in order that the effects may not be duplicated at the next inspection station. Equations 5.15 and 5.16 calculate the modified batch size for processing and inspection stations respectively. The model assumes that process drifts do not occur at inspection stations, the inspection stations detect all errors perfectly and the attrition in batch size at these stations in only attributive to the normal or reduced yields (not scrap yield).

$$Z_{i0} = 1 \tag{5.14}$$

$$\left.\begin{array}{rcl} B_{ij} & = & B_{[i,H(i,j)]}\, y^s_{ij} \\ Z_{ij} & = & Z_{[i,H(i,j)]} z_{ij} \end{array}\right\} \text{if } j \in R_i \cap \mathcal{J} \tag{5.15}$$

$$\left.\begin{array}{rcl} B_{ij} & = & B_{[i,H(i,j)]} Z_{[i,H(i,j)]}\, y^s_{ij} \\ Z_{ij} & = & 1 \end{array}\right\} \text{if } j \in R_i \cap \mathcal{F} \tag{5.16}$$

Aggregation. Aggregation calculates, for each product, the processing time of each job at each station. It also calculates, for each station, the average processing time, weighted by each product's arrival rate. Finally, it modifies the aggregate processing times by adjusting for the resource availability. The time spent by a job at station j is the sum of the part processing times and the setup time.

$$t^+_{ij} = B_{[i,H(i,j)]} t_{ij} + s_{ij} \tag{5.17}$$

$$(t^+_{ij})^2 c^+_{ij} = B_{[i,H(i,j)]} t^2_{ij} c^t_{ij} + s^2_{ij} c^s_{ij} \tag{5.18}$$

Equation 5.18, which is used to calculate c^+_{ij}, holds because the variance of the total process time is the sum of the variance of the part process times and the variance of the job setup time. The aggregate process time of jobs at station j is the weighted average of all the jobs that visit station j. Each product is weighted by its release rate, as shown in Equation 5.19. Equation 5.20 calculates the mean of the square aggregate process time, which can be used to determine the SCV c_j^+.

$$t_j^+ = \frac{\sum_{i \in V_j} x_i t_{ij}^+}{\sum_{i \in V_j} x_i} \tag{5.19}$$

$$(t_j^+)^2 (c_j^+ + 1) = \frac{\sum_{i \in V_j} x_i (t_{ij}^+)^2 (c_{ij}^+ + 1)}{\sum_{i \in V_j} x_i} \tag{5.20}$$

Equations 5.21 and 5.22 modify the mean and SCV for the process times by adding the effects of resource availability.

$$t_j^\star = \frac{t_j^+}{A_j} \tag{5.21}$$

$$c_j^\star = c_j^+ + 2A_j(1 - A_j)\frac{m_j^r}{t_j^+} \tag{5.22}$$

Arrival and Departure Processes. The arrival process at each station depends upon the products that visit the station. Some products are released directly to the station, while others arrive from other stations. The departure process depends upon the arrival process and the service process.

V_{0j} = the set of products that visit station j first, $\{i \in V_j : H(i,j) = 0\}$

V_{hj} = the set of products that visit station h immediately before j, $\{i \in V_j : H(i,j) = h\}$

$$\begin{aligned}
\lambda_j &= \text{total job arrival rate at station } j \\
\lambda_{hj} &= \text{arrival rate at station } j \text{ of jobs from station } h \\
q_{hj} &= \text{proportion of jobs from station } h \text{ that next visit station } j \\
c_j^a &= \text{SCV of interarrival times at station } j \\
c_j^d &= \text{SCV of interdeparture times at station } j
\end{aligned}$$

$$\begin{aligned}
\lambda_j &= \sum_{i \in V_j} x_i && (5.23) \\
\lambda_{hj} &= \sum_{i \in V_{hj}} x_i && (5.24) \\
q_{hj} &= \frac{\lambda_{hj}}{\lambda_h} && (5.25)
\end{aligned}$$

Equations 5.26 and 5.27 estimate the SCVs for the departure and arrival processes.

$$\begin{aligned}
c_j^d &= 1 + \frac{u_j^2}{\sqrt{n_j}}(c_j^\star - 1) + (1 - u_j^2)(c_j^a - 1) && (5.26) \\
c_j^a &= \sum_{h \in \mathcal{J} \cup \mathcal{F}} ((c_h^d - 1)q_{hj} + 1)\frac{\lambda_{hj}}{\lambda_j} + \sum_{i \in V_{0j}} c_i^r \frac{x_i}{\lambda_j} && (5.27)
\end{aligned}$$

Solving the above set of equations yields the complete set of c_j^a and c_j^d for all stations.

Performance Measures. The performance measures of interest are the average utilization and throughput of stations and the manufacturing cycle time. The average manufacturing cycle time of a job depends upon the manufacturing cycle time at each station it visits.

$$\begin{aligned} u_j &= \frac{t_j^\star}{n_j} \sum_{i \in V_j} x_i && (5.28) \\ CT_j^\star &= \frac{1}{2}(c_j^a + c_j^\star) \frac{u_j^{(\sqrt{2n_j+2}-1)}}{n_j(1-u_j)} t_j^\star + t_j^\star && (5.29) \\ CT_i &= \sum_{j \in R_i} CT_j^\star && (5.30) \end{aligned}$$

Also, it is important to remember that the throughput is less than the release rate due to yield losses. The throughput for product i depends on the job size for product i at the last station in the processing sequence for product i. If k is the last station in R_i ($R_{ik} = R_i$),

$$T_i^o = x_i B_{ik} \tag{5.31}$$

Discussion. The significant difference between the model presented here and the one in Chapter 3, as explained earlier is that this model considers the presence of process drift in the manufacturing system and the impact that manufacturing cycle time has on yield and throughput. Yield losses lead to a significant reduction in the batch size for the product during processing. Equations 5.15 and 5.16 indicate models to calculate this reduced batch size. This may be considered to be equivalent to reseting the state of the station (the state of the station being a binary quantity). This assumption finds basis in anticipated repair situations wherein a failed station would have all sources of failure fixed before being deemed fit to resume service.

This work assumed a two state model for simplification purposes. Alternatively, each resource may be assumed to have multiple states. In that case, there would be a yield value associated with every state. As a resource in the manufacturing system attains a particular state, the model would need to incorporate the corresponding yield value to calculate the batch size for the product at the workstation.

5.3 Effects of Process Drift

Process drift is different from the yield loss at a station:

- The defect due to a process drift may not be detected at the workstation where it occurred.

- Once a process drift occurs at a station, it affects each product batch for all products that are processed at the station. Thus, once the error occurs, until it is detected and rectified, the workstation will have a lower yield for all subsequent product batches.

- The error may be detected only at a functional or quality test station at some point downstream from where it occurs in the process flow.

- After the occurrence of the error, before detection, all stations continue to process batches continuously.

Thus, the error due to a process drift is different from that due to scrap yield at a workstation and the effects on the system that result, are also different. Some of the effects of a process drift are as follows:

- The process defect is not detected at the station where it occurs[2]. Hence some time elapses between occurrence and detection. The time that elapses is equal to the sum of the manufacturing cycle times of all the workstations in the processing sequence for the product, between the station where the defect occurs and the next inspection station.

- During this time, the defective products continue to be processed on the subsequent processing stations as per their processing sequence.

- Additionally, during this time, the batches of products that are in line at the station where the process defect occurs, continue to be processed on the defective workstation.

- The effects of process drift are two-fold;

  - The products which are processed at the defective station are in error and need to be scrapped.

  - The time spent by the other workstations which continue to process these defective products is wasted. Thus these stations lose capacity.

- Due to potential attrition from process drift, it is necessary to start more jobs.

- Since the manufacturing cycle time at a workstation depends on the resource utilization (Equation 5.28), which in turn is a function of the release rates of products visiting the workstation (Equation 5.29), higher product release rates increase the resource utilization, leading to an increase in the station

[2] This research assumes that the drift is detected at a subsequent inspection station.

manufacturing cycle time and the manufacturing cycle time for each product. As a result, the detection time for a process drift defect increases.

Reduced throughput due to process drift reduces the revenue from the new product. In addition, process drift leads to shortages contributing to shortage costs.

5.4 Flow Shops

This section presents an algorithm to calculate the system performance for a flow-shop manufacturing system processing multiple products (all products that are being processed in the given manufacturing system have the same processing sequence). The computations are based on the mathematical model explained in Section 5.2.

Consider a system processing a given product set such that all products in the product set visit the same sequence of stations. Each product routing in the product set therefore has the same set of processing stations. This is known as a flow shop. For the system, there is a set of products $\mathcal{I}$ being processed, and $\exists \mathcal{S} : R_i = \mathcal{S}, \forall i \in \mathcal{I}$. Then the stations may be renumbered as $1, 2, \ldots, n$; $n = \mid \mathcal{J} \cup \mathcal{F} \mid$. $V_j = \mathcal{I}$ and $V_{j-1,j} = \mathcal{I}$ for all stations. Equation 5.27 can then be simplified as follows:

$$c_1^a = \frac{\sum_{i \in \mathcal{I}} c_i^r x_i}{\sum_{i \in \mathcal{I}} x_i} \tag{5.32}$$

$$c_j^a = c_{j-1}^d, 2 \leq j \leq n \tag{5.33}$$

Note that,

$$
\begin{aligned}
Q_{ij} &= \{j+1,\ \ldots,\ m\},\ m \in \mathcal{F} \\
H_{(i,j)} &= j-1,\ \forall i \in \mathcal{I},\ 1 \leq j \leq n
\end{aligned}
$$

Algorithm. Flow_Shop_Cycle_Time

```
Main: AGGREGATION($\mathcal{I}, \mathcal{J} \cup \mathcal{F}$)

1. for each $j \in \mathcal{S}$
        compute availability $A_j$

2. for each $i \in \mathcal{I}$
        compute $x_i = \frac{T_i^a}{B_{i0}}$

3. set $h = 1, 1 \in \mathcal{J}$

4. while $h \leq n$
        set $j = h$
        while $j \in \mathcal{J}$
            for each $i \in \mathcal{I}$
                $B_{i,j} = B_{i,j-1} y_{ij}^s$
            CALCULATE-CT($j$)
            $j = j + 1$
        set $m = j,\ m \in \mathcal{F}$
        for $j = h$ to $m - 1$
            calculate $DT_j^\star = \sum_{k=j+1}^{m} CT_k^\star$
```

```
    for $i \in \mathcal{I}$
        calculate $z_{i,j}$
        $Z_{i,j} = Z_{i,j-1} z_{i,j}$
for $i \in \mathcal{I}$
    $B_{i,m} = B_{i,m-1} Z_{i,m-1} y^s_{i,m}$
    CALCULATE-CT($m$)
    $Z_{i,m} = 1$
$h = m + 1$
```

Function: CALCULATE-CT(p)

Function: CALCULATE-CT(p) estimates the manufacturing cycle time for station p using the closed form solution described by Equations 5.17 to 5.29. Since all products being processed in the manufacturing system have the same processing sequence, the algorithm begins calculating the manufacturing cycle time at the first station in the sequence and progresses along the sequence evaluating station parameters depending on whether the workstation is a processing station or an inspection station. The algorithm has complexity $O(IS)$ where I is the number of products being processed in the system and S is the number of resources in the processing sequence.

5.5 Flow Shop Example

Consider the product, microwave modules, which was explained in detail in Chapter 3. Though the product domain is the same, there are some modifications to the processing sequence in order to help illustrate the current situation better. The following paragraphs list these changes.

The situation is identical to that in Chapter 3, where, the manufacturing facility currently processes two products *MWM Product 1* and *MWM Product 2*. The company intends to introduce a new product *Improved MWM* into market. The manufacturing system is an electronics assembly shop. It is a flow-shop system to the extent that all products being processed in the system have the same processing sequence. Changes to the part processing time for any process could be caused by changes to the product design or the manufacturing process.

The information about the product and the system are based on experience with an electronic systems manufacturer. This example uses data that the collaborators for this work were able to provide and other synthetic data created as part of the research effort.

5.5.1 Two Products

The two products *MWM Product 1* and *MWM Product 2*, currently being processed in the system have processing sequences in the form of a set of processing stations inter spaced with inspection stations as follows,

1. *Processing Station 1:* Machine holes and pockets
2. *Inspection Station 1:* Inspect_1 (check for presence of burrs from machining)
3. *Processing Station 2:* Plate (electroless, or autocatalytic plating)
4. *Processing Station 3:* Plate (electroplating)
5. *Processing Station 4:* Etch (clean, apply photoresist, expose, develop, etch, clean)
6. *Inspection Station 2:* Inspect_2 (check for etching errors)

7. *Processing Station 5:* Automated Assembly (mount and solder surface mount components)

8. *Processing Station 6:* Manual Assembly (attach other components)

9. *Inspection Station 3:* Test (and tune as necessary)

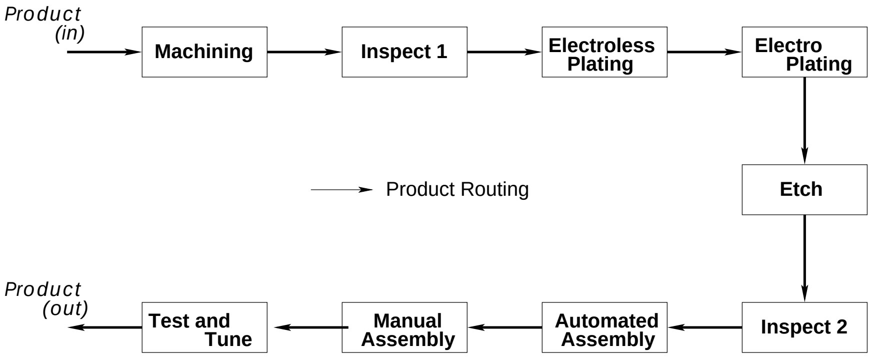

Figure 5.5: Routings for the Flow Shop

Figure 5.5 shows the routings for the two products in the manufacturing system. As explained earlier, the processing system currently processes two products *MWM Product 1* and *MWM Product 2*. Table 5.1 shows various system parameters when the system is processing only these two products.

Table 5.1 shows the processing times at various workstations when process drift occurs with normal yield $y_{ij}^n = 0.95,\ \forall i, j$ and reduced yield $y_{ij}^r = 0.95,\ \forall i, j$, while the process drift rate $\rho_j = 0.9,\ \forall j$.

With the presence of process drift in the manufacturing system, the average batch size at output reduces drastically from the input batch size. The throughput which is a product of the product release rate and the output batch size reduces in the case as well. Table 5.2 shows the values of the system performance measures.

j	Workstation Name	Processing Times t_{ij}^{+}	
		MWM Product 1	*MWM Product 2*
1	Machining	41.52	63.18
2	Inspection_1	90.00	165.00
3	Electroless Plating	32.67	32.67
4	Electroplating	60.61	60.61
5	Etch	57.01	64.02
6	Inspection_2	45.04	80.08
7	Automated Assembly	42.08	0.00
8	Manual Assembly	2.42	4.83
9	Test and Tune	66.23	102.48

Table 5.1: System processing two products, $\mathcal{J} = \{1, 3, 4, 5, 7, 8\}$, $\mathcal{F} = \{2, 6, 9\}$

System Performance Measure	Value
Average input batch size	7.5
Average output batch size	1.274
Release rate	0.1876
Manufacturing cycle time (hours)	14.66
Throughput (parts/min)	0.03184

Table 5.2: Output parameters for two product system

5.5.2 System With New Product

The new product *Improved MWM* has the same processing sequence as the products being currently processed in the system. There are distinctions in the design of the new product in terms of the number of discrete electronic components mounted on the substrate, along with some changes to the geometry of the substrate. The processing sequence however remains the same as the old products. Table 5.3 shows the system parameters when it is processing two and three products. The process drift rates, normal and reduced yields remain the same when the new product is introduced.

Table 5.4 shows a comparison between the system performance when it is pro-

j	*MWM Product 1*			*MWM Product 2*			*Improved MWM*	
	t^{+}_{ij}	Batch Size, B_{ij}		t^{+}_{ij}	Batch Size, B_{ij}		t^{+}_{ij}	Batch Size, B_{ij}
		Three Prod.	Two Prod.		Three Prod.	Two Prod.		
1	41.518	5	5	63.176	10	10	69.811	12
2	90	5	5	195	10	10	63.176	12
3	32.667	5	5	32.667	10	10	32.667	12
4	60.608	5	5	60.608	10	10	60.608	12
5	60	5	5	74	10	10	70	12
6	60	1.7187	1.7217	130	3.4373	3.4433	110	4.1248
7	47.187	1.7187	1.7217	50.624	3.4373	3.4433	64.373	4.1248
8	3.437	1.7187	1.7217	24.749	3.4373	3.4433	6.875	4.1248
9	81.69	0.8464	0.849	153.743	1.6928	1.698	133.12	2.3014

Table 5.3: System parameters for the two and three product scenarios

cessing two products and when it is processing three. The comparison shows that though the average manufacturing cycle time for the system processing three products has nearly doubled from that for the system processing two, the average throughput has doubled as well.

	Two Products	Three Products
Manufacturing Cycle Time (hours)	15.3072	30.8812
Arrival (Release) Rates (parts/min)	0.5	0.76
Throughput (parts/min)	0.03183	0.06666

Table 5.4: Comparison between the two scenarios

5.5.3 Comparison

The model presented in the earlier sections may be evaluated by comparing its results to those for a discrete event simulation model. This work created two simulation models using Arena©[3] one each for the two products and three products

[3] Arena is a registered trademark of Rockwell Automation

scenarios. The following paragraphs briefly explain the models. The Arena© models were created by Sara Hewitt and are described in greater detail in [64].

Arena Models.

The entities entering the manufacturing system are *raw products.* The finished products are obtained after these raw products pass through a nine-step process. The simulation model creates raw products according to an exponentially distributed interarrival time. The processing times at each step follow an Erlang-2 distribution. When Arena creates raw products it assigns the raw product processing times for each station in the system and a batch size specifying the number of raw products in the batch. The products are then routed to the first manufacturing station.

This Arena model creates defects as entities that trigger a process to become out-of-control. Arena creates defects according to an exponentially distributed interarrival time with a mean equal to one over the drift rate $(\frac{1}{\rho_j})$. Each step has its own drift rate, so each step has its own unique type of defect entity; that is, the defect that causes step three to go out-of-control is different and independent of the defect that will cause step four to go out-of-control. When Arena creates a defect, the defect immediately travels from the create block to the station that the defect will cause to go out-of-control (as shown in Figure 5.6). When a defect is detected, the inspection station fixes only that defect; if there are multiple defects at a station when one defect is detected, only one of the defects is corrected.

There is a defect counter for each processing step. As mentioned before, a different type of defect entity affects each step. Therefore, the defect counter for processing step three only counts the step three defects in the system while the

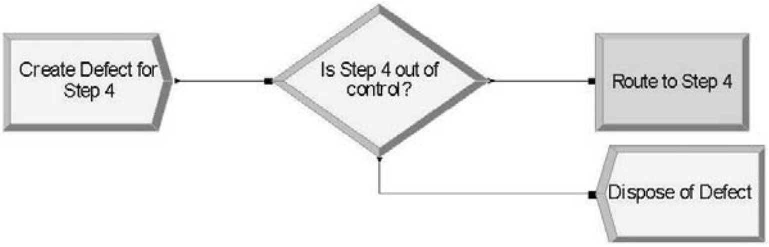

Figure 5.6: Defect traverse in block diagram

defect counter for step four only counts the step four defects in the system and so on. When a defect arrives at a station, the defect counter for that station is incremented by one. A manufacturing step is deemed to be out-of-control whenever its defect counter equals one. The defect remains at the station until a raw product arrives at the station. When the raw product arrives, it checks to see if there are any defects waiting at the station. If there are no waiting defects, the raw product is processed and continues through the system. If there is a defect waiting at the station, the defect entity is "joined" to the raw product. The joined raw product and defect entity is akin to a sticker being placed on the raw product indicating that the step is out-of-control. The raw product and the defect now go through the system together, obeying the processing times and rules for the raw product. (see Figure 5.7).

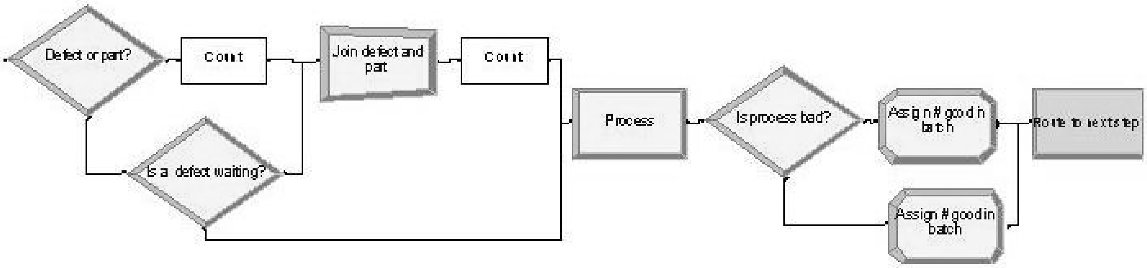

Figure 5.7: Raw product and defect block

At an inspection station the raw product and defect are delayed for a specified inspection processing time. The raw product and defect are then split apart and travel through a series of logic blocks that identify defect entities. Whenever the logic blocks detect a defect entity, they pull the defect out of the system, decrease the defect counter by one and dispose of the defect entity. (see Figure 5.8)

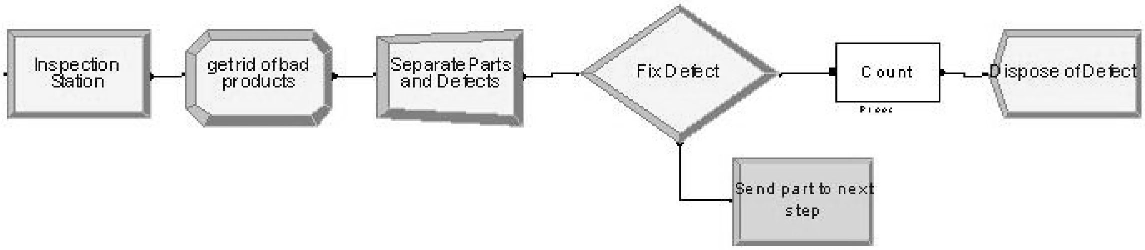

Figure 5.8: Logic block defect detection

The number of good products in a batch is recalculated at each step. The calculation is a function of the previous number of good products in the batch and the yield of the step, which depends on whether or not the step is out of control. The number in the batch is recalculated only at inspection stations.

Note that the Arena model needs the batch size to have an integer value. For example, if the batch size is 98 and the yield is 98%, the expected number of good parts in a batch is (0.98)(98) = 96.04. However, the Arena model treats 96.04 as 96, thereby reducing the yield to 97.96% instead of 98%. In order to create integer numbers for the number of good parts in a batch, and maintain the correct yield, the number of good parts in the batch is calculated using a modified formula. This modified expression calculates the number of good parts in a batch, B_{ij}, as either the rounded down integer value of $B_{ij} = B_{[i,H(i,j)]}\ y^r_{ij}$, or as the batch size $B_{ij} = B_{[i,H(i,j)]}$. The batches are calculated assuming 100% yield calculation for a fraction of the time and the integer value equal to $B_{[i,H(i,j)]}\ y^r_{ij}$ for the remainder of the time. The Arena model implements this by having each batch go through a probability module that determines if the batch will be multiplied by a fractional yield or by a 100% yield. The probability module is re-evaluated for each batch that passes through the probability module. The probability x that $B_{ij} = B_{[i,H(i,j)]}$ is determined for each batch as follows:

$$x = \frac{B_{[i,H(i,j)]} y^r_{ij} - Integer[B_{[i,H(i,j)]} y^r_{ij}]}{B_{[i,H(i,j)]} - Integer[B_{[i,H(i,j)]} y^r_{ij}]} \tag{5.34}$$

5.5.4 Trials

The analytical and simulation models were each run for five different values of the processing time at the Etch station. The values were obtained by modifying an additive constant in the formula to calculate the processing time at the Etch station. Table 5.5 shows the inputs for the experiment conducted to compare the two models for one value of processing time at the Etch station, for the new product set. This set of experiments was repeated for 5 values of processing time at the Etch station, leading to 45 experiments in total. This set of experiments was also repeated for the case where the manufacturing system processes only the old product set. Twenty trials of Arena models were run for 200,000 minutes each, with no warm-up period.

5.5.5 Results

Tables 5.6, 5.7 and 5.8 present sample results comparing the analytical and simulation models. The results are only for the three products case and correspond to one value of the Etch processing time. The results for the other four Etch processing times were very similar. Process drift causes a reduction in the batch

Variable	Values	Abbreviation
Etch processing time (value of additive constant to job processing time in min.)	5 10 15 20 25	
Batch sizes $B_{1,0}, B_{2,0}, B_{3,0}$	50, 100, 150 100, 200, 300 150, 300, 450	S1 S2 S3
Arrival rates T_1^a, T_2^a, T_3^a	0.0208, 0.0104, 0.0035 0.0417, 0.0208, 0.0069 0.0625, 0.0313, 0.0139	L1 L2 L3

Table 5.5: Scenarios for model comparison

size as the product proceeds along the operations sequence due to the reduced yield under conditions of drift. The throughput for the system is calculated using the output batch size for a product and the release rate for the product. As the batch size decreases, the throughput for the system decreases. Hence, the system performance measure of interest in this example is the system throughput, which in turn, means the output batch size for the products. Note that the detection time for the process drift is a function of the manufacturing cycle times for the processing stations, as was shown earlier.

The simulation results represent a 95% confidence interval. The batch sizes for the different trials at Inspection Station 1 are shown in Table 5.6. At this station the results from the analytical model are all within 1% of the average simulation results, which is within the 95% confidence interval for the simulation results. Table 5.7 shows the results of the batch sizes at Inspection Station 2. At this station some of the analytical results are outside the confidence interval. For *MWM Product 1*, the analytical results are within 1% of the average simulation results. For *MWM Product 2*, they are within 1.6% of the average analytical value.

	Desired	*MWM Product 1*		*MWM Product 2*		*Improved MWM*	
Batch	Prod. Qty.	Analy.	Simulation	Analy.	Simulation	Analy.	Simulation
S1	L1	40.506	40.442 ± 0.217	81.012	80.953 ± 0.445	121.518	121.540 ± 0.903
	L2	40.505	40.461 ± 0.236	81.011	80.836 ± 0.538	121.516	121.360 ± 0.823
	L3	40.248	40.163 ± 0.149	80.495	80.348 ± 0.284	120.743	120.440 ± 0.447
S2	L1	76.939	77.027 ± 0.586	153.877	153.810 ± 1.208	230.816	231.700 ± 1.918
	L2	76.938	77.236 ± 0.442	153.876	154.380 ± 0.827	230.814	232.030 ± 1.369
	L3	76.644	76.428 ± 0.385	153.287	153.780 ± 0.767	229.931	229.040 ± 1.402
S3	L1	112.598	112.680 ± 0.589	225.195	225.220 ± 1.251	337.793	337.530 ± 2.087
	L2	112.597	112.580 ± 0.579	225.194	225.330 ± 1.318	337.791	336.760 ± 2.080
	L3	112.267	112.050 ± 0.626	224.534	223.920 ± 1.166	336.802	336.190 ± 1.818

Table 5.6: Batch Sizes at Inspection Station 1

	Desired	*MWM Product 1*		*MWM Product 2*		*Improved MWM*	
Batch	Prod. Qty.	Analy.	Simulation	Analy.	Simulation	Analy.	Simulation
S1	L1	20.946	20.532 ± 0.163	41.893	41.110 ± 0.328	62.839	62.325 ± 0.682
	L2	20.495	20.466 ± 0.153	41.862	41.020 ± 0.387	62.793	62.418 ± 0.601
	L3	20.285	19.549 ± 0.275	40.571	39.392 ± 0.343	60.856	59.570 ± 0.496
S2	L1	34.391	33.959 ± 0.430	69.861	68.283 ± 0.963	104.792	104.070 ± 1.388
	L2	34.928	34.153 ± 0.299	69.857	68.731 ± 0.599	104.785	104.180 ± 1.244
	L3	34.474	34.494 ± 0.268	69.988	67.966 ± 0.593	103.481	102.100 ± 1.039
S3	L1	47.853	47.654 ± 0.352	95.705	94.682 ± 0.872	143.558	142.360 ± 1.794
	L2	47.851	47.710 ± 0.339	95.703	95.296 ± 0.778	143.554	143.570 ± 1.333
	L3	47.404	47.136 ± 0.313	94.807	93.780 ± 0.677	142.211	142.211 ± 1.012

Table 5.7: Batch Sizes at Inspection Station 2

	Desired	*MWM Product 1*		*MWM Product 2*		*Improved MWM*	
Batch	Prod. Qty.	Analy.	Simulation	Analy.	Simulation	Analy.	Simulation
S1	L1	11.294	10.826 ± 0.081	22.588	22.005 ± 0.160	33.882	33.188 ± 0.349
	L2	11.286	10.825 ± 0.077	22.572	21.984 ± 0.197	33.857	33.291 ± 0.287
	L3	10.925	10.302 ± 0.091	21.849	21.019 ± 0.158	32.774	31.814 ± 0.250
S2	L1	18.207	17.604 ± 0.232	36.414	35.534 ± 0.492	54.621	53.830 ± 0.824
	L2	18.206	17.685 ± 0.170	36.411	35.703 ± 0.348	54.617	53.926 ± 0.738
	L3	17.955	17.491 ± 0.142	35.910	35.374 ± 0.301	53.865	52.980 ± 0.509
S3	L1	24.565	24.072 ± 0.187	49.131	48.369 ± 0.425	73.696	72.767 ± 0.851
	L2	24.565	24.098 ± 0.175	49.129	48.749 ± 0.464	73.694	72.949 ± 0.692
	L3	24.308	23.807 ± 0.150	48.616	48.008 ± 0.311	72.923	71.966 ± 0.515

Table 5.8: Batch Sizes at Output

For *Improved MWM*, the analytical results are within 1.3% of the corresponding average simulation values, but except for two cases (throughput L3 for batch size S2 and S3), lie within the confidence interval.

Table 5.8 shows the batch sizes at the last inspection station (Test and Tune). Here, for trials with input batch size S1, the analytical results are within 3% of the average simulation value except for *MWM Product 1* with batch size S2 when the difference is within 6%. The analytical batch size values for the three products almost always lie outside the confidence interval at the last inspection station even though the difference in the analytical value and the average simulation value is not very large. The confidence interval extrema are within 1.2% of the average value for any of the trials.

Note. Owing to attrition of jobs from process drift, it becomes essential to have a higher release rate for jobs into the manufacturing system. Notably, it is no longer possible to state that increasing or decreasing the processing time at any station in the processing sequence would correspondingly affect the overall manufacturing cycle time of the product.

As an illustration, consider the case where a modification to a product design results in a decrease in the processing time at a station in its processing sequence. Intuition would dictate that a decrease in the processing time (which contributes to the product manufacturing cycle time) would result in an associated decrease in the overall manufacturing cycle time for the product. However, a decrease in the processing time results in reduced manufacturing cycle time at the station. Hence, the detection time for process drift is reduced, is discovered earlier and Z_{ij} are reduced. As a result, the batch size of product i leaving the next inspection station is greater than before. Therefore the manufacturing cycle time for i at all further processing stations is greater resulting in a higher cumulative manufacturing cycle time for the product.

Consider the product and manufacturing scenario for a flow shop detailed above. For the system processing three products, the processing time for the electroless plate station is incremented uniformly from 2.67 min to 62.67 min. Figure 5.9 helps to demonstrate an instance of the above conjecture. The figure depicts system performance as a function of this increasing processing time. As can be seen, the cycle time decreases (albeit by a small amount) before uniformly increasing, as the processing time increases. The throughput continuously increases, though by a small amount.

Mathematical foundations for these conjectures are presented in [23].

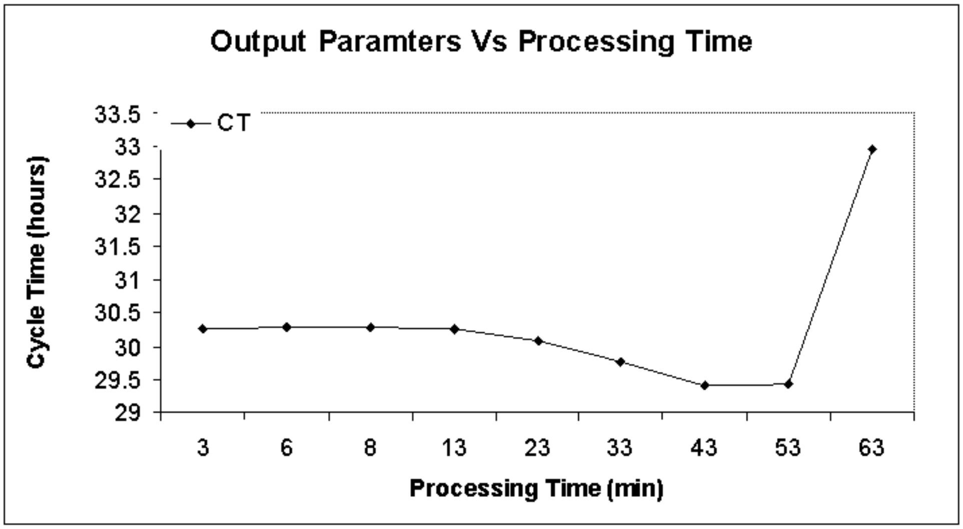

Figure 5.9: System Output Vs Processing Time

5.6 Creating a Graph Representation for the System Parameters

As explained in the introduction, a later section in this chapter will show the inadequacies of the algorithm for the flow shop case to handle the general job-shop case. However, before that is attempted it is necessary to introduce some concepts and algorithms which will then be used as tools to explain the reasons.

Definitions.

System Graph : A *System Graph* $\mathcal{SG}$ is a directed graph representation of relationships between key performance measures for a given manufacturing system processing a given set of products, which may be used to determine system characteristics and evaluate performance.

Predecessor Set : The *Predecessor Set* Π_z for a node z in $\mathcal{SG}$ is the set of nodes that immediately precede z in the directed graph; that is, $\pi_z \in \Pi_z$ if and only if $\exists$ directed edge (π_z, z) connecting π_z to z.

Predecessor
Successor

Figure 5.10: Node Relation

Successor Set : The *Successor Set,* Σ_z for a node z in $\mathcal{SG}$ is the set of nodes that immediately succeed z in the directed graph; that is, $\sigma_z \in \Sigma_z$ if and only if $\exists$ directed edge (z, σ_z) connecting z to σ_z.

Note, $a \in \Pi_b$, if and only if $b \in \Sigma_a$.

Sub-Graph : A *Sub-Graph* $\mathcal{SSG}$ is a graph such that $\mathcal{SSG} \subset \mathcal{SG}$.

Figure 5.10 shows the relationship between the nodes of the graph. The following algorithm creates system graph $\mathcal{SG}$ for the given product and manufacturing system parameters. The algorithm requires the following notation,

Notation.

B Node $\mapsto$ Node in $\mathcal{SG}$ associated with the batch size of product i

Z Node $\mapsto$ Node in $\mathcal{SG}$ associated with the hidden yield of product i

CT Node $\mapsto$ Node in $\mathcal{SG}$ associated with the manufacturing cycle time of station j

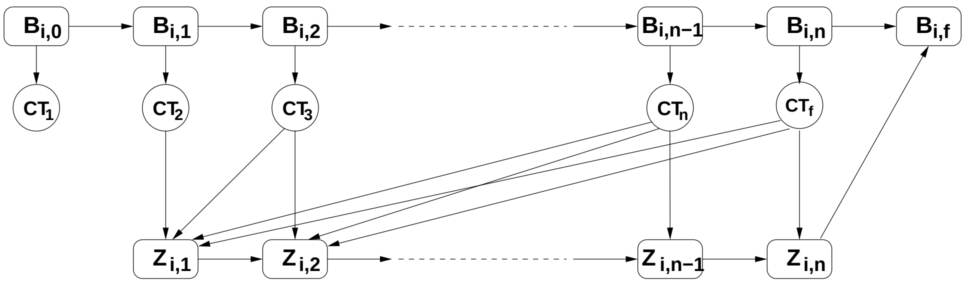

Figure 5.11: Sub-graph

Algorithm. Graph_Creation

```
Main: CREATION($\mathcal{I}, \mathcal{J} \cup \mathcal{F}, \mathcal{SG}$)

1. initialize $\mathcal{SG}$

2. for each $i \in \mathcal{I}$
       create node $B_{i0}$
       add $B_{i0}$ to $\mathcal{SG}$
```

```
3. for each $i \in \mathcal{I}$
       for each $j \in R_i$
           create node $B_{ij}$
           create node $Z_{ij}$
           add $B_{ij}$ to $\mathcal{SG}$
           add $Z_{ij}$ to $\mathcal{SG}$

4. for each $j \in \mathcal{J} \cup \mathcal{F}$
       create node $CT_j$
       add $CT_j$ to $\mathcal{SG}$

5. for each $i \in \mathcal{I}$
       for each $j \in R_i$
           create edge $B_{i,H(i,j)} \longrightarrow B_{ij}$
           create edge $B_{i,H(i,j)} \longrightarrow CT_j$
       for each $j \in R_i \cap \mathcal{F}$
           create edge $Z_{i,H(i,j)} \longrightarrow B_{ij}$
       for each $j \in R_i \cap \mathcal{J}$
           create edge $Z_{i,H(i,j)} \longrightarrow Z_{ij}$
       for each $j \in R_i$
           for each $k \in Q_{ij}$
               create edge $CT_k \longrightarrow Z_{ij}$

6. for each $j \in \mathcal{J}$
       for each $i \in V_j$
           for each $h \in V_j,\ h \neq i$
               for each $k \in Q_{hj}$
```

```
if $\nexists$ edge $CT_k \longrightarrow Z_{ij}$
    create edge $CT_k \longrightarrow Z_{ij}$
```

Explanation. In Step 2, the initial batch size refers to the arrival batch size for product i at the first station in its processing sequence R_i. The departure batch at a processing station depends on the arrival batch size (Equations 5.15 and 5.16). The arrival batch size is equal to the departure batch size at the earlier processing station in the processing sequence, assuming no attrition in material movement between stations. Thus, (in Step 3) the node corresponding to the batch size at a station for a product becomes a successor node for the batch size at the previous station in its processing sequence. It is seen from Equation 5.16 that the batch size at the inspection station depends on the hidden yield of the previous processing station. Hence, the hidden yield for the earlier processing station should be a predecessor node for the batch size at an inspection station. Step 5 adds these directed edges.

The hidden yield at a processing station depends on the hidden yield at the previous processing station (Equation 5.15). Hence, a node corresponding to the hidden yield of the previous processing station should be a predecessor node for the node corresponding to the hidden yield at each station in the processing sequence. Step 5 adds these directed edges. Equations 5.15 to 5.29 show that the manufacturing cycle time at each station is a function of the batch size of the product arriving at the station. Step 5 adds these directed edges. The hidden process yield at each station depends on the defect detection time for a defect occurring in a product at the station (Equation 5.13). The defect detection time, as Equations 5.11 and 5.12 show, depends on the manufacturing cycle times for the stations in the processing sequence of the product between the station where the process drift occurs and the nearest inspection station. Step 5 incorporates these dependencies into the system graph.

Step 6 accounts for the fact that a process drift at a station is likely to be detected at the inspection station in the shortest possible detection time between the two stations. This means that if different products are processed at a station with process drift and follow different routings to inspection stations, the detection time for the drift is the smaller of the sums of manufacturing cycle times of the resources that the products visit before arriving at the respective inspection stations, as Equation 5.12 indicates. Thus cycle times at stations that are not in a product's processing sequence could influence the hidden yield for the product at a processing station in its processing sequence.

5.7 Identifying the Dependencies in System Parameters

There exist dependencies between various product and system variables in the model represented by Equations 5.8 to 5.31. A graph is a convenient way to represent these dependencies. In addition, this graph can be used to determine the process for calculating these variables. The *System Graph* defined in the previous section can be used for this purpose. This is discussed in more detail in Section 5.9.

Figure 5.11 depicts the basic subgraph for product i. This represents a part of the processing sequence for the product, which has n processing stations between

- successive inspection stations, or
- between the start of the processing sequence and the first processing station,

or

- between the last inspection station and the end of the processing sequence [Note that for this case, the last batch size node in the sub-graph would be removed along with the edge connecting the $Z_{i,n}$ node to it.]

In Figure 5.11, stations, $1, 2, \cdots n \in \mathcal{J}, f \in \mathcal{F}$. This sub-graph has $(n + 1)$ nodes associated with the batch sizes, n nodes associated with the hidden yield, and $(n + 1)$ nodes associated with the manufacturing cycle times making the total number of nodes equal to $(3n + 2)$. The sub-graph has n edges connecting the hidden yield nodes, $(n + 1)$ edges connecting the batch size nodes, $(n + 1)$ edges connecting the batch size nodes to the manufacturing cycle time nodes and $\frac{n(n+1)}{2}$ edges connecting the hidden yield nodes to the manufacturing cycle time nodes making the total number of edges in the graph equal to $[(3n+2)+\frac{n(n+1)}{2}]$. This sub-graph can be used to show the dependencies between various system parameters. This is explained with an example in Subsection 5.8.3.

5.8 Examples of System Graphs

This section presents some examples that illustrate how the system graph may be created for different manufacturing scenarios.

5.8.1 Example 1: Implementing the Algorithm for Two Products in a Flow Shop

Consider Figure 5.12 that shows the routings for two products being processed in a given manufacturing system.

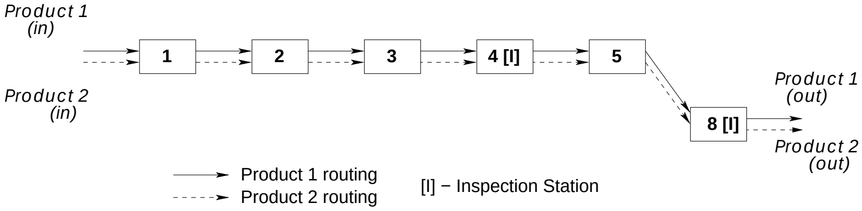

Figure 5.12: Two Products Processing Sequence

The two products have the same processing sequence which comprises a combination of processing and inspection stations. Figure 5.13 shows the system graph for this two product system created based on the graph creation algorithm detailed in Section 5.6. The following is a detailed recipe for creating the system graph:

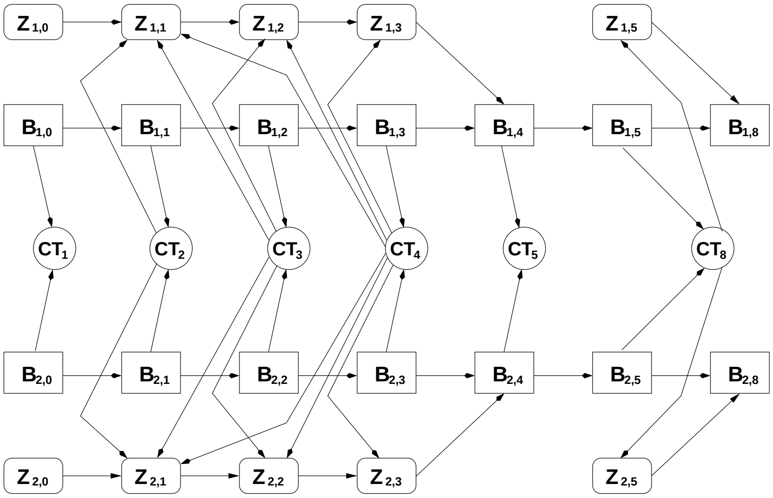

Figure 5.13: System Graph for Product Set from Figure 5.12

1. Create nodes $B_{1,0}, \cdots, B_{1,8}$ and $B_{2,0}, \cdots, B_{2,8}$ corresponding to the input batch sizes for stations that products 1 and 2 visit.

2. Create nodes $CT_1, \cdots, CT_8$ corresponding to the workstations in the processing system.

3. Create nodes $Z_{1,0}, \cdots, Z_{1,3},\ Z_{1,5},\ Z_{2,0}, \cdots, Z_{2,3},\ Z_{2,5}$ corresponding to the hidden yields at those processing stations likely to influence the yield at the inspection stations.

4. Create directed edges connecting the batch size node for a workstation in a product's processing sequence to the batch size node corresponding to the next workstation in the sequence. Thus, $B_{1,0} \longrightarrow B_{1,1} \longrightarrow B_{1,2} \cdots,\ B_{2,0} \longrightarrow B_{2,1} \longrightarrow B_{2,2}, \cdots$ and so on.

5. Create directed edges from each batch size node for a product to the manufacturing cycle time corresponding to the next station in the product's processing sequence. Thus, $B_{1,0} \longrightarrow CT_1,\ B_{2,0} \longrightarrow CT_1,\ B_{1,1} \longrightarrow CT_2,\ B_{2,1} \longrightarrow CT_2$ and so on.

6. Create directed edges connecting the hidden yield node for a processing station in a product's processing sequence to the batch size node corresponding to the next processing station in the sequence. Thus, $Z_{1,0} \longrightarrow Z_{1,1} \longrightarrow Z_{1,2}, \cdots,\ Z_{2,0} \longrightarrow Z_{2,1} \longrightarrow Z_{2,2}, \cdots,$ and so on.

7. Create a directed edge from the hidden yield node corresponding to the earlier workstation in the processing sequence for a product to the batch size node corresponding to an inspection station. Thus, $Z_{1,3} \longrightarrow B_{1,4},\ Z_{2,3} \longrightarrow B_{2,4},\ Z_{1,5} \longrightarrow B_{1,8},\ Z_{2,5} \longrightarrow B_{2,8}$.

8. Create directed edges to the hidden yield node at a workstation from each manufacturing cycle time node corresponding to processing stations between that processing station and the next inspection station. Thus, $CT_2 \longrightarrow Z_{1,1},\ CT_3 \longrightarrow Z_{1,1},\ CT_4 \longrightarrow Z_{1,1},\ CT_2 \longrightarrow Z_{2,1},\ CT_3 \longrightarrow Z_{2,1},\ CT_4 \longrightarrow Z_{2,1}$.

5.8.2 Example 2: Implementing the Algorithm for Three Products in a Simple Job Shop

Consider the following scenario for this example. The manufacturing company currently produces two products and is developing a third. The process plans for the existing two products have the same processing sequence while the new product has a different one. While the new product does visit some of the stations that the old product set is processed on, its processing sequence also includes some additional workstations.

The system processes two products *MWM Product 1* and *MWM Product 2*, whose process plans have the following sequence:

1. Mill (one through and two blind pockets and two holes).
2. Electroless, or autocatalytic plating.
3. Etch (clean, apply photoresist, expose, develop, etch, clean)
4. Inspect (check for validity of etched tracks)
5. Automated Assembly (mount and solder surface mount components)
6. Test (functional testing and validation)

The new product to be introduced into the system, *Improved MWM* has a modified design which requires the assembly of some components manually. The final system test is incorporated into the manual assembly step. In addition, the milling step is replaced by a milling plus grinding operation in order to accurately finish the substrate which is potentially likely to mate with an external component (where the microwave module is mounted). Further, the Etch operation is replaced with an Artwork operation requiring a more sophisticated dedicated workstation.

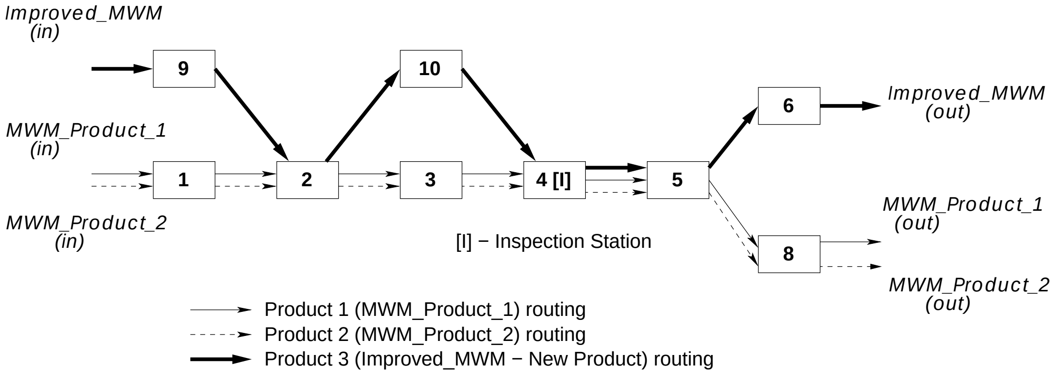

Figure 5.14: Processing Sequence for 3 Product Set

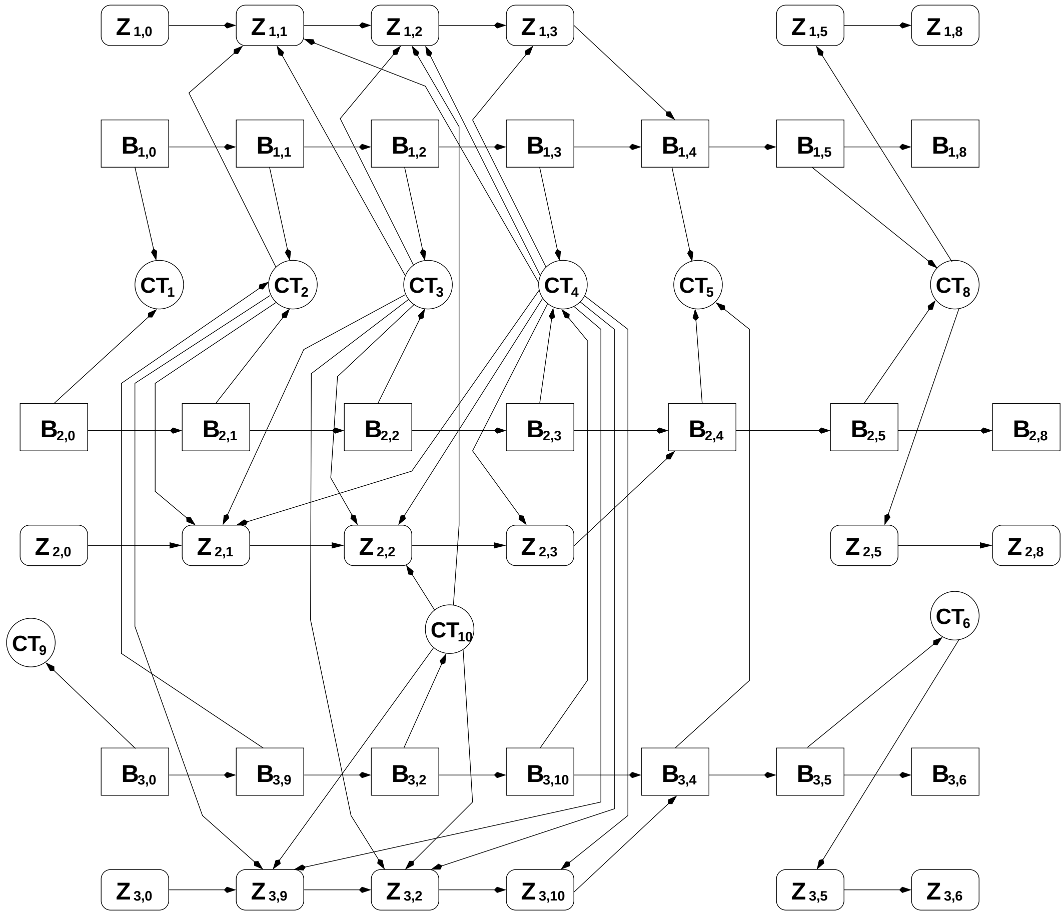

Figure 5.15: System Graph for 3 Product Set

The remaining sequence remains the same as the existing products. Hence, the processing sequence for *Improved MWM* is;

1. Grind and Mill
2. Electroless plating.
3. Artwork
4. Inspect (check for validity of etched tracks)
5. Automated Assembly (mount and solder surface mount components)
6. Manual Assembly

The processing sequence for the system with the old and new products is shown in Figure 5.14.

The system graph for the three product system may be drawn using the algorithm in Section 5.6. This is shown in Figure 5.15.

5.8.3 Example 3: Implementing the Algorithm for a Job Shop

Consider Figure 5.16 showing the routings for two products. The two products require a total of 10 processing stations while sharing some stations. Stations $\{1, 2, 3, 5, 6, 7\} \in \mathcal{J}$ while stations $\{4, 8, 9, 10\} \in \mathcal{F}$.

The processing sequences of the two products have processing and inspection stations and they share some stations. Product 1 has routing $\{1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 6 \rightarrow 7 \rightarrow 8\}$, while Product 2 has routing $\{5 \rightarrow 6 \rightarrow 7 \rightarrow 9 \rightarrow 1 \rightarrow 2 \rightarrow 3 \rightarrow 10\}$.

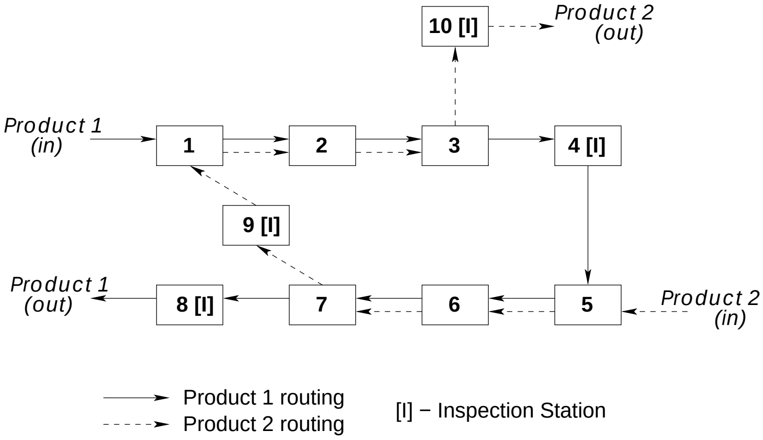

Figure 5.16: Sample Routings for two Products

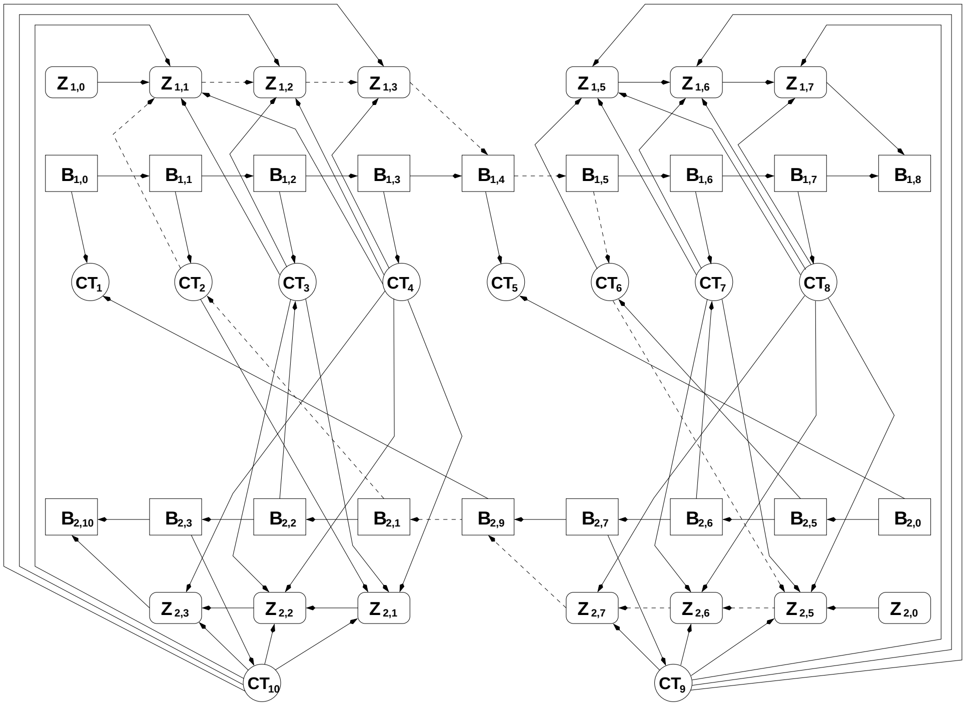

Figure 5.17: System graph for system in Figure 5.16

Figure 5.17 shows the system graph for the system in Figure 5.16 created using the graph creation algorithm. This graph is different from the earlier created system graphs in that it contains loops. The sections of the graph represented by dashes show one such loop present in the graph.

Variable	Key Factors affecting Variable		
	Batch Size	Process Drift Yield	Cycle Time
$B_{1,8}$	$B_{1,7}$	$Z_{1,7}$	
CT_8	$B_{1,7}$		
$B_{1,7}$	$B_{1,6}$		
$Z_{1,7}$		$Z_{1,6}$	CT_8
CT_7	$B_{1,6}$		
$B_{1,6}$	$B_{1,5}$		
$Z_{1,6}$		$Z_{1,5}$	CT_8, CT_7
$\underline{\mathbf{CT_6}}$	$\underline{\mathbf{B_{1,5}}}$		
$\underline{\mathbf{B_{1,5}}}$	$\underline{\mathbf{B_{1,4}}}$		
$Z_{1,5}$		$Z_{1,4}$	CT_8, CT_7, CT_6
CT_5	$B_{1,4}$		
$\underline{\mathbf{B_{1,4}}}$	$B_{1,3}$	$\underline{\mathbf{Z_{1,3}}}$	
CT_4	$B_{1,3}$		
$B_{1,3}$	$B_{1,2}$		
$\underline{\mathbf{Z_{1,3}}}$		$\underline{\mathbf{Z_{1,2}}}$	CT_4
CT_3	$B_{1,2}$		
$B_{1,2}$	$B_{1,1}$		
$\underline{\mathbf{Z_{1,2}}}$		$\underline{\mathbf{Z_{1,1}}}$	CT_4, CT_3
CT_2	$B_{1,1}$		
$B_{1,1}$	$B_{1,0}$		
$\underline{\mathbf{Z_{1,1}}}$		$Z_{1,0}$	$CT_4, CT_3, \underline{\mathbf{CT_2}}$

Table 5.9: (continued)

Variable	Key Factors affecting Variable		
	Batch Size	Process Drift Yield	Cycle Time
CT_1	$B_{1,0}$		
$B_{2,10}$	$B_{2,3}$	$Z_{2,3}$	
CT_{10}	$B_{2,3}$		
$B_{2,3}$	$B_{2,2}$		
$Z_{2,3}$		$Z_{2,2}$	CT_{10}
CT_3	$B_{2,2}$		
$B_{2,2}$	$B_{2,1}$		
$Z_{2,2}$		$Z_{2,1}$	CT_{10}, CT_3
$\underline{\mathbf{CT_2}}$	$\underline{\mathbf{B_{2,1}}}$		
$\underline{\mathbf{B_{2,1}}}$	$\underline{\mathbf{B_{2,9}}}$		
$Z_{2,1}$		$Z_{2,9}$	CT_{10}, CT_3, CT_2
CT_1	$B_{2,9}$		
$\underline{\mathbf{B_{2,9}}}$	$B_{2,7}$	$\underline{\mathbf{Z_{2,7}}}$	
CT_9	$B_{2,7}$		
$B_{2,7}$	$B_{2,6}$		
$\underline{\mathbf{Z_{2,7}}}$		$\underline{\mathbf{Z_{2,6}}}$	CT_9
CT_7	$B_{2,6}$		
$B_{2,6}$	$B_{2,5}$		
$\underline{\mathbf{Z_{2,6}}}$		$\underline{\mathbf{Z_{2,5}}}$	CT_9, CT_7
CT_6	$B_{2,5}$		
$B_{2,5}$	$B_{2,0}$		
$\underline{\mathbf{Z_{2,5}}}$		$Z_{2,0}$	$CT_9, CT_7, \underline{\mathbf{CT_6}}$

Table 5.9: (continued)

Variable	Key Factors affecting Variable		
	Batch Size	Process Drift Yield	Cycle Time
CT_3	$B_{2,0}$		

Table 5.9: Dependence Table

Table 5.9 shows the dependence of various parameters for the products and stations on other parameters of the system. One of the loops created by the inter-dependencies is shown by the underlined entities in the table. Thus, $CT_6 \rightarrow Z_{2,5} \rightarrow Z_{2,6} \rightarrow Z_{2,7} \rightarrow B_{2,9} \rightarrow B_{2,1} \rightarrow CT_2 \rightarrow Z_{1,1} \rightarrow Z_{1,2} \rightarrow Z_{1,3} \rightarrow B_{1,4} \rightarrow B_{1,5} \rightarrow CT_6$ is one of the loops where CT_6 depends indirectly on CT_2 which in turn depends on CT_6. (Here, $P \rightarrow Q$ implies that P governs Q or that the value of Q cannot be calculated until the value of P is known.) Similar analyses shows the interdependencies between $CT_2 \Leftrightarrow CT_7, CT_3 \Leftrightarrow CT_6, CT_3 \Leftrightarrow CT_7$ and so on.

Hence it is necessary to check the system parameters for interdependencies and the presence of such loops in order that appropriate solution techniques for solving the problem may be identified.

5.9 Using the System Graph

Section 5.4 presented an algorithm for estimating system parameters for a flow shop processing one or more products.

However, the algorithm may not work for the more general job-shop problem. Such a job-shop scenario has product set, $\mathcal{I}$ being processed in a given manufacturing system $\mathcal{J} \cup \mathcal{F}$. Process sequence R_i is the set of processing steps or operations for product $i \in \mathcal{I}$. In a job-shop scenario, $R_i \neq R_p,\ i, p \in \mathcal{I}$. Thus, different products follow different routes through the processing system. In general, the analysis of a job-shop scenario requires systematic evaluation of the system variables. However, due to the interdependencies between system parameters, a different algorithm (like the algorithm proposed in Section 5.11) is necessary.

The following lemmas are valid for the system graph $\mathcal{SG}$ created using the algorithm in Section 5.6.

Lemma 3 *There exists no node $n \in \mathcal{SG}$ such that $n \in \Pi_n$ or $n \in \Sigma_n$.*

If $\exists$ a directed edge from node $n \in \mathcal{SG}$ to itself, this would mean that the quantity represented by node n depends upon itself. However, based on Algorithm: Graph_Creation presented in Section 5.6 (as also Equations 5.8 to 5.31, used to calculate various parameters for the manufacturing system and the product set that is processed), it is clear that no system parameter depends upon itself. Hence, $\nexists n \in \mathcal{SG}$ such that $n \in \Pi_n$ or $n \in \Sigma_n$.

Lemma 4 *There exist no nodes a and $b \in \mathcal{SG} : a \in \Sigma_b$ and $b \in \Sigma_a$.*

By definition, system graph $\mathcal{SG}$ contains B nodes, Z nodes and CT nodes. From the construction of $\mathcal{SG}$ given by Algorithm: Graph_Creation (Step 5), $\mathcal{SG}$ contains the following types of directed arcs:

- $B_{i,H(i,j)} \longrightarrow B_{ij}$
- $B_{i,H(i,j)} \longrightarrow CT_j$
- $Z_{i,H(i,j)} \longrightarrow B_{ij}$
- $Z_{i,H(i,j)} \longrightarrow Z_{ij}$

- $CT_k \longrightarrow Z_{ij}$

These follow from relationships between the system parameters defined by Equations 5.8 to 5.31, for a given manufacturing system processing a set of products $\mathcal{I}$. Since these are the only edges possible, it is clear that there does not exist a simple loop between any two nodes of $\mathcal{SG}$.

The following lemmas can be verified based on the dependence of various quantities in the mathematical model represented by Equations 5.8 to 5.30 and Algorithm: Graph_Creation.

Lemma 5 *For any $i \in \mathcal{I}$, $j \in \mathcal{J}$, there is an arc $\in \mathcal{SG}$ from $B_{i,H(i,j)}$ to B_{ij}.*

For a processing station ($j \in \mathcal{J}$), according to Equation 5.8, the batch size of product i, $i \in V_j$ at station j is a function of the batch size at the previous station and the scrap yield of product i at station j. Hence, it is possible to calculate the batch size for product i at j given the batch size at the previous station $H(i,j)$ that product i visits.

Lemma 6 *For any $i \in \mathcal{I}$ $j \in \mathcal{J}$, there is an arc $\in \mathcal{SG}$ from $Z_{i,H(i,j)}$ to Z_{ij}.*

The hidden yield of product i from process drift at output of station j depends on the time delay between occurrence and detection of the process defect (which occurs at the closest inspection station). This in turn depends on the effective unchecked yield due to drift for product i at each station in the processing sequence of i prior to j. Thus, the hidden yield of product i at station at j may be calculated if the hidden yield of product i at the previous station $H(i,j)$ that product i visits is known.

Lemma 7 *For any $j \in \mathcal{F},\ i \in \mathcal{I}$, there is an arc $\in \mathcal{SG}$ from $B_{i,H(i,j)}$ to B_{ij} and from $Z_{i,H(i,j)}$ to B_{ij}.*

From Lemma 5 above, the batch size for product i at resource j can be calculated if the batch size for i at $H(i, j)$ is known. The effects of hidden yield due to delay in detection of process defects that occur during processing are translated to the next inspection station in the sequence. These effects, in turn, cause a reduction in the batch size for product i at the inspection station. Thus, in order to be able to calculate the batch size for i at inspection station $j \in \mathcal{F}$, it is essential to first calculate the hidden yields at all processing stations prior to it. Conversely, if the hidden yields at all processing stations prior to an inspection station are known, using Lemma 5, the batch size at the inspection station can be calculated.

Theorem 1 *The only possible edges in $\mathcal{SG}$ are $\mathcal{B} \rightarrow \mathcal{B}, \mathcal{B} \rightarrow \mathcal{CT}, \mathcal{Z} \rightarrow \mathcal{B}, \mathcal{Z} \rightarrow \mathcal{Z}$, and $\mathcal{CT} \rightarrow \mathcal{Z}$.*

Proof :

The proof for this Theorem follows directly from Steps 5 and 6 of Algorithm: Graph_Creation.

Q.E.D.

Corollary 1 *For any $\mathcal{SG}$, the following properties hold:*

- $\nexists B_{ij}$ *and* $Z_{hk} \in \mathcal{SG}\ (i, h \in \mathcal{I},\ j \in R_i,\ k \in R_h)\ : Z_{hk} \in \Sigma_{B_{ij}}$.
- $\nexists B_{ij}$ *and* $CT_k \in \mathcal{SG}\ (i \in \mathcal{I},\ j \in R_i,\ \ k \in \mathcal{J} \cup \mathcal{F}) : B_{ij} \in \Sigma_{CT_k}$.
- $\nexists Z_{ij}$ *and* $CT_k \in \mathcal{SG}\ (i \in \mathcal{I},\ j \in R_i,\ k \in \mathcal{J} \cup \mathcal{F}) : CT_k \in \Sigma_{Z_{ij}}$.

- $\nexists CT_j$ *and* $CT_k \in \mathcal{SG}$ $(j, k \in \mathcal{J} \cup \mathcal{F}) : CT_j \in \Sigma_{CT_k}$.

Theorem 2 *Consider a manufacturing system with* $j \in \mathcal{J}$ *(processing stations),* $f \in \mathcal{F}$ *(inspection stations). It processes one product (*$\mathcal{I} = \{i\}$*). Let* $\mathcal{SG}$ *be the system graph for this manufacturing system. Then,* $\mathcal{SG}$ *cannot contain a cycle.*

Proof :

Let m be the last station in R_i.

Let p be the number of stations in the set $\mathcal{F} \cap R_i$

If $m \in \mathcal{F}$, then $\mathcal{SG}$ can be partitioned into p sub-graphs.

If $m \in \mathcal{J}$, then $\mathcal{SG}$ can be partitioned into $(p + 1)$ sub-graphs.

These sub-graphs correspond to different subsequences of R_i. Each subsequence ends with a station in $\mathcal{F}$ if $m \in \mathcal{F}$ (or the last station in R_i if $m \in \mathcal{J}$).

Based on the definition of $\mathcal{SG}$, there exists no path from one sub-graph to an earlier sub-graph.

Consider one sub-graph of $\mathcal{SG}$. Let b be the first station in the sub-sequence.

Each sub-graph may be partitioned into four sub-sets in the following manner: $\mathcal{A}$ contains the nodes B_{ib} and $B_{ij},\ j \in Q_{ib} \cap \mathcal{J}$; $\mathcal{B}$ contains the nodes CT_b and $CT_k,\ k \in Q_{ib}$; $\mathcal{C}$ contains the nodes Z_{ib} and $Z_{ij},\ j \in Q_{ib} \cap \mathcal{J}$; $\mathcal{D}$ contains the node B_{if} where $f \in Q_{ib} \cap \mathcal{F}$ (if $\mathcal{F}$ exists, else it is an empty set). See Figure 5.18.

$\mathcal{A}$ contains a simple path, and $\exists$ edges from nodes in $\mathcal{A}$ to nodes in $\mathcal{B}$. Also, $\exists$ edges from nodes in $\mathcal{B}$ to nodes in $\mathcal{C}$. $\mathcal{C}$ contains a simple path and $\mathcal{B}$ does not contain any edges.

If $\mathcal{D}$ is not empty, let $n = H(i, f)$. There is an edge from Z_{in} in $\mathcal{C}$ to B_{if} in $\mathcal{D}$.

The only edges from one sub-graph to another are the one from B_{if} to B_{ig} and the one from B_{if} to CT_g, where $f = H(i, g)$, and $B_{ig},\ CT_g$ belong to the next sub-graph.

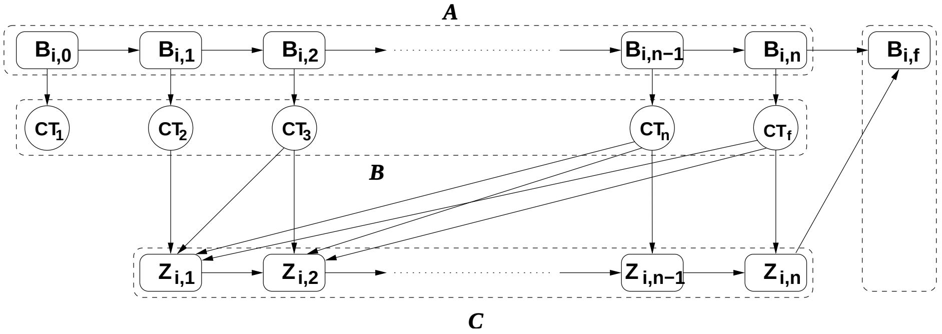

Figure 5.18: 4 Sets in the sub-graph of$\mathcal{SG}$

There are no cycles in sets $\mathcal{A}$, $\mathcal{B}$, $\mathcal{C}$, $\mathcal{D}$. Thus, there are no cycles in the sub-graph.

Therefore, there is no cycle in $\mathcal{SG}$.

Q.E.D.

Theorem 3 *Consider a manufacturing system with $j \in \mathcal{J}$ (processing stations), $f \in \mathcal{F}$ (inspection stations), processing a set of products $i \in \mathcal{I}$. Let $\mathcal{SG}$ be the system graph for this manufacturing system. If $\exists$ a cycle $\mathcal{C}$ in $\mathcal{SG}$, then $\mathcal{C}$ must contain at least two manufacturing cycle time nodes.*

Proof :

First it is shown that if $\exists$ a cycle $\mathcal{C}$ in $\mathcal{SG}$, then such a cycle contains at least one type CT node.

Based on Lemmas 3 and 4, $\mathcal{C}$ contains at least 3 nodes.

By definition, $\mathcal{SG}$ comprises only three types of nodes: B, Z, and CT nodes.

Suppose $\mathcal{C}$ has all type B nodes.

Let B_{ij} be one of these nodes, and B_{hk} be the next node in the cycle. From Step 3 in the graph creation algorithm, the edge between these nodes exists if and only if $h = i,\ j = H(i,k),\ j \in R_{ik}$. (The output batch size for a product at a resource can be influenced by the input batch size at the resource only if it is the input batch size for the same product.)

Continuing in this manner, it is possible to show that any node in $\mathcal{C}$ must therefore be a node $B_{iy},\ y \in R_i$, and $j \in R_{iy}$.

Let B_{ic} be the node that precedes B_{ij} in $\mathcal{C}$. Note that $c \in R_i,\ j \in R_{ic}$. However, an edge from B_{ic} to B_{ij} exists iff $c = H(i,j)$. Thus, $c \in R_{ij}$. But, this contradicts Lemma 1.

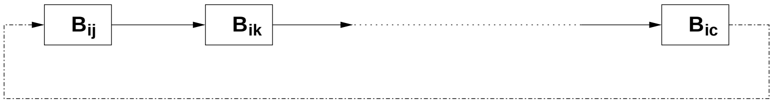

Figure 5.19: Cycle $\mathcal{C}$ with all B type nodes

Hence, $\mathcal{C}$ cannot contain only type B nodes.

Suppose $\mathcal{C}$ contains only type Z nodes.

Let Z_{ij} be one of these nodes and Z_{hk} be the next node. The hidden yield for product i, at a station j depends on the hidden yield at earlier stations in R_i, where $j \notin \mathcal{F}$. From Step 5 in the graph creation algorithm, an edge between nodes Z_{ij} and Z_{hk} exists iff $h = i,\ j = H(i,k),\ k \notin \mathcal{F},\ j \in R_{ik}$.

Continuing in this manner, it is possible to show that any node in $\mathcal{C}$ must therefore be a node $Z_{iy},\ y \in R_i$, and $j \in R_{iy}$.

Let $d \in Q_{iy} \cap \mathcal{F}$. As argued above, $\exists$ a directed path from Z_{iy} to Z_{ie}, $e = H(i,d)$ comprising all type Z nodes. However, per Lemma 7, Z_{ie} must connect to B_{id}.

Hence, $\mathcal{C}$ cannot contain all type Z nodes.

Suppose $\mathcal{C}$ consists of type B and type Z nodes only.

Then, in $\mathcal{C}$, there must be a directed edge from a type B node to a type Z node and there must be a directed edge from a type Z node to a type B node.

However, this contradicts Corollary 1. Therefore, it is not possible that $\mathcal{C}$ contains only type B and Z nodes.

Thus, cycle $\mathcal{C}$ must have at least one type CT node (Figure 5.20). Suppose $\mathcal{C}$ contains only one type CT node. Let this node be CT_k.

From Theorem 1, the successor of CT_k in $\mathcal{C}$ must be a type Z node, say Z_{ab}. From Step 5 in Algorithm: Graph_Creation, $k \in Q_{ab},\ a \in \mathcal{I}, b \in \mathcal{J}$ (which implies $b \in R_{ak}$).

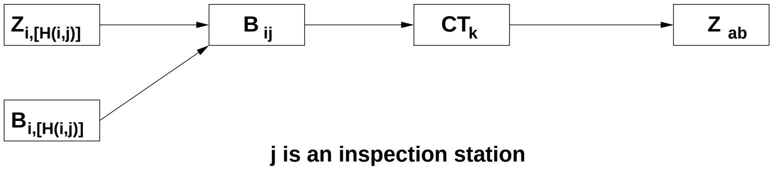

Figure 5.20: Connections between type B, CT and Z nodes

From Theorem 1, the predecessor of CT_k must be a type B node. Without loss of generality let B_{ij} be the predecessor node of CT_k in $\mathcal{SG}$. From Step 5 of Algorithm: Graph_Creation, it is seen that $k \in R_i,\ j = H(i,k)$.

Consider product $a \in \mathcal{I}$.

If product $i = a$, $\mathcal{C}$ contains path $B_{aj} \longrightarrow CT_k \longrightarrow Z_{ab}$. For $\mathcal{C}$ to exist, there must be a directed path from Z_{ab} to B_{aj} (Figure 5.21). This path contains no type CT nodes since the only type CT node in $\mathcal{C}$ is CT_k. Hence the path must contain a sequence of type Z nodes followed by a sequence of type B nodes.

Let $f \in Q_{ab} \cap \mathcal{F}$. An edge can exist from Z_{ae} to B_{af} only if $e = H(i,f)$ and $f \in \mathcal{F}$.

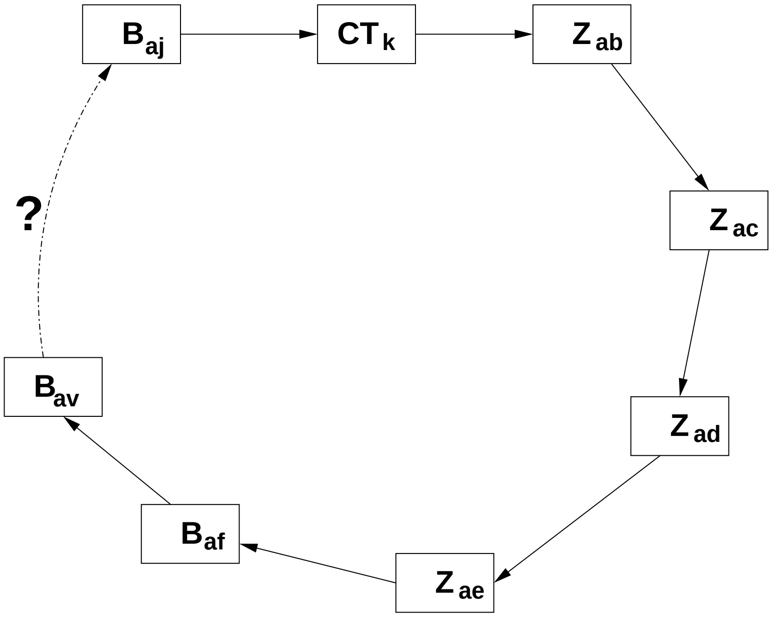

Figure 5.21: Instance of a cycle in $\mathcal{SG}$ when product $a = i$

Thus, if $Q_{ab} \cap \mathcal{J} \neq \emptyset$, the directed path from Z_{ab} contains nodes Z_{ac}, $\forall c \in Q_{ab} \cap \mathcal{J}$ till Z_{ae}. This is followed by B_{af}, which is then followed by nodes B_{ap}, $\forall p \in R_{aj}$.

Therefore, there exists a directed path from B_{af} to B_{aj}. This implies $f \in R_{aj}$.

But, $j = H(a, k)$, which implies $R_{aj} \subset R_{ak}$, which implies $f \in R_{ak}$.

Therefore, station f is between stations b and k in R_a ($b \in R_{ak}, f \in Q_{ab}, f \in R_{ak}$). This contradicts the statement that $k \in Q_{ab}$.

Therefore, it is not possible that C contains only one type CT node.

Consider that $i \neq a$. Then there must exist a path from Z_{ab} to B_{ij}. Further, the path must contain nodes Z_{ac}, $\forall c \in Q_{ab} \cap \mathcal{J}$.

The first type B node must be $B_{af}, f \in Q_{ab} \cap \mathcal{F}$.

Let $\mathcal{V} \subset \mathcal{C}$ be the directed path from B_{af} to B_{ij}.

$\mathcal{V}$ must contain nodes B_{ad}, $d \in R_a$. This is true since $\mathcal{C}$ only contains one type CT node, CT_k and there is no $B \longrightarrow Z$ node in $\mathcal{SG}$. (Step 5 of Algorithm:

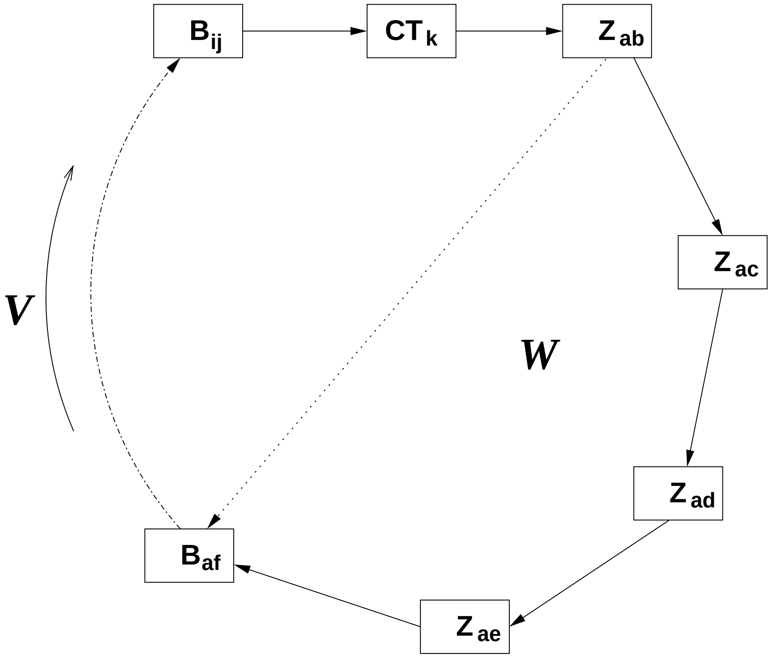

Figure 5.22: Instance of a cycle in $\mathcal{SG}$

Graph_Creation.) Further, from Algorithm: Graph_Creation, there is no edge from any B_{ad} to B_{ij}.

Therefore, it is not possible that $\mathcal{C}$ contains only one type CT node.

Because there is no cycle in $\mathcal{SG}$ with exactly one type CT node, $\mathcal{C}$ must contain at least two type CT nodes.

Q.E.D.

Corollary 2 *If $\exists$ cycle $\mathcal{C} \subset \mathcal{SG}$, $\mathcal{C}$ comprises two or more paths $\mathcal{H}$ of the following form,*

$$CT_k \longrightarrow Z_{ab} \longrightarrow \cdots \longrightarrow B_{af} \longrightarrow \cdots \longrightarrow B_{ax} \longrightarrow CT_m,\ m \neq k.$$

Proof :

The proof of this corollary follows from the proof of Theorem 3 and the definition and construction of the system graph $\mathcal{SG}$ for a manufacturing system processing a given set of products explained in Algorithm: Graph_Creation.

Theorem 4 *If $R_i = R_a$, $\forall i, a \in \mathcal{I}$, then there is no cycle $\mathcal{C}$ in $\mathcal{SG}$.*

Proof :

Consider a manufacturing system with $j \in \mathcal{J}$ (processing stations), $f \in \mathcal{F}$ (inspection stations), processing a set of products $i \in \mathcal{I}$. Let $\mathcal{SG}$ be the system graph for this manufacturing system.

Let g be the last station in R_i, $\forall i \in \mathcal{I}$.

Let s be the number of stations in $R_i \cap \mathcal{F}$, $i \in \mathcal{I}$.

If $g \in \mathcal{F}$, then $\mathcal{SG}$ can be partitioned into s sub-graphs.

If $g \in \mathcal{J}$, then $\mathcal{SG}$ can be partitioned into $(s+1)$ sub-graphs.

Figure 5.23 shows such partitioning of a sample system graph for a manufacturing system processing two products.

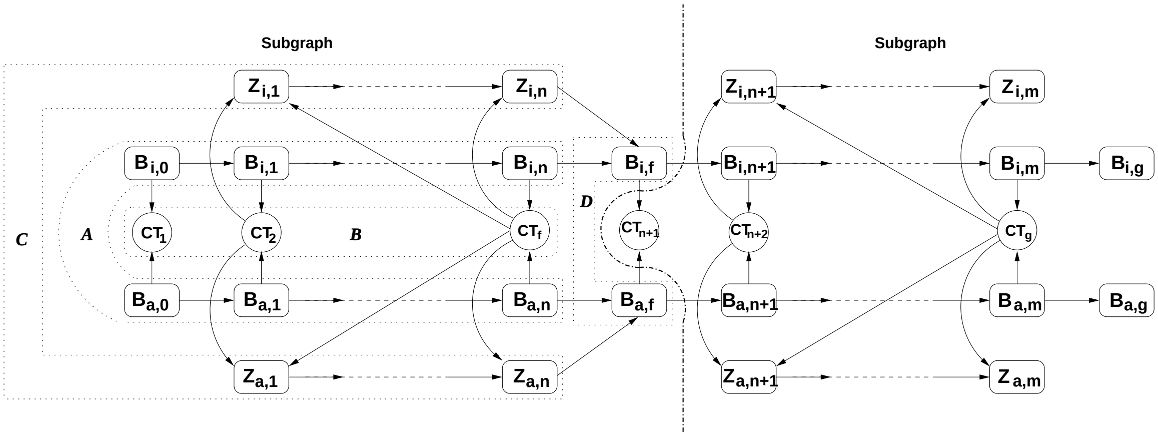

Figure 5.23: Subdivided system graph for a flow shop

These sub-graphs include nodes associated with corresponding subsequences of $R_i, \forall i \in \mathcal{I}$. Consider a subsequence that begins with station $b \in \mathcal{J}$ and ends with station $f \in Q_{ib} \cap \mathcal{F}$ (or the last station in R_i).

The nodes in the corresponding sub-graph can be partitioned into four disjoint sub-sets in the following manner: $\mathcal{A}$ is a set containing the nodes B_{ib} and $B_{ij},\ j \in Q_{ib} \cap \mathcal{J}, \forall i \in \mathcal{I}$; $\mathcal{B}$ contains the nodes CT_b and $CT_k,\ \forall k \in Q_{ib}$; $\mathcal{C}$ is a set containing the nodes Z_{ib} and $Z_{ij},\ j \in Q_{ib} \cap \mathcal{J}, \forall i \in \mathcal{I}$; $\mathcal{D}$ is a set containing the nodes B_{if} where, $f \in Q_{ib} \cap \mathcal{F}, \forall i \in \mathcal{I}$ (if f exists, else $\mathcal{D}$ is an empty set).

From the definition of $\mathcal{SG}$ and Lemmas 5 and 6 it is clear that sets $\mathcal{A}$ and $\mathcal{C}$ contain disjoint simple paths, and the only edges between the sets are limited to the edges depicted in Figure 5.24.

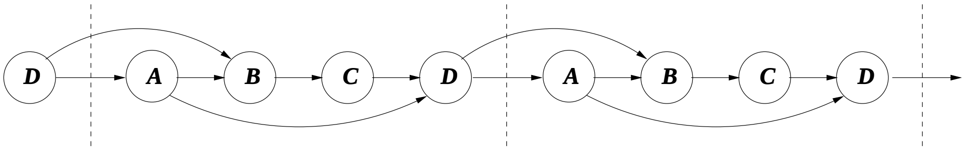

Figure 5.24: Set Interconnectivity

Since there are no edges from one type CT node to another, set $\mathcal{B}$ does not contain any edges. Since there are no edges between $B_{if}, B_{af},\ \forall i, a \in \mathcal{I}$, set $\mathcal{D}$ contains no edges.

In a sub-graph, the only edges between subsets are those from nodes in $\mathcal{A}$ to nodes in $\mathcal{B}$, those from nodes in $\mathcal{A}$ to nodes in $\mathcal{D}$, those from nodes in $\mathcal{B}$ to nodes in $\mathcal{C}$, and those from nodes in $\mathcal{C}$ to nodes in $\mathcal{D}$ (see Figure 5.24).

Therefore, there is no cycle in a sub-graph.

When $g \in \mathcal{J}$, (as shown in Figure 5.23), the sub-graph corresponding to the subsequence containing g does not have set $\mathcal{D}$. As a result the only directed set connectivity for the subsets in this subgraph are forward connections from $\mathcal{A}$ to $\mathcal{B}$ to $\mathcal{C}$. Thus, there is no cycle in this sub-graph.

It has been established that there is no cycle in any sub-graph. Further, there exists no path from one sub-graph to an earlier sub-graph. The only edges that connect one sub-graph to another are the edges from B_{if} to B_{ih} and CT_h, where $f = H(i, h)$. This adds edges from set $\mathcal{D}$ in one sub-graph to the sets $\mathcal{A}$ and $\mathcal{B}$ in the next sub-graph (as shown in Figure 5.24).

Hence, there is no cycle in $\mathcal{SG}$.

Q.E.D.

<u>NOTE</u>: This proof exploits the special structure of the system graph $\mathcal{SG}$ in the flow shop production scenario. The algorithm to calculate system properties for a flow shop, Algorithm: Flow_Shop_Cycle_Time, detailed earlier in this chapter also exploits this special structure.

5.10 Detecting Loops in the System Graph

The system graph developed in Section 5.6 is a connected directed graph. The problem of finding a loop in the system graph is equivalent to finding a cycle in a directed graph. A graph search technique such as depth-first search may be modified to traverse the graph and search for cycles in the graph. The dashed lines in Figure 5.25 along with the nodes they connect indicate one cycle in the system graph from Figure 5.17.

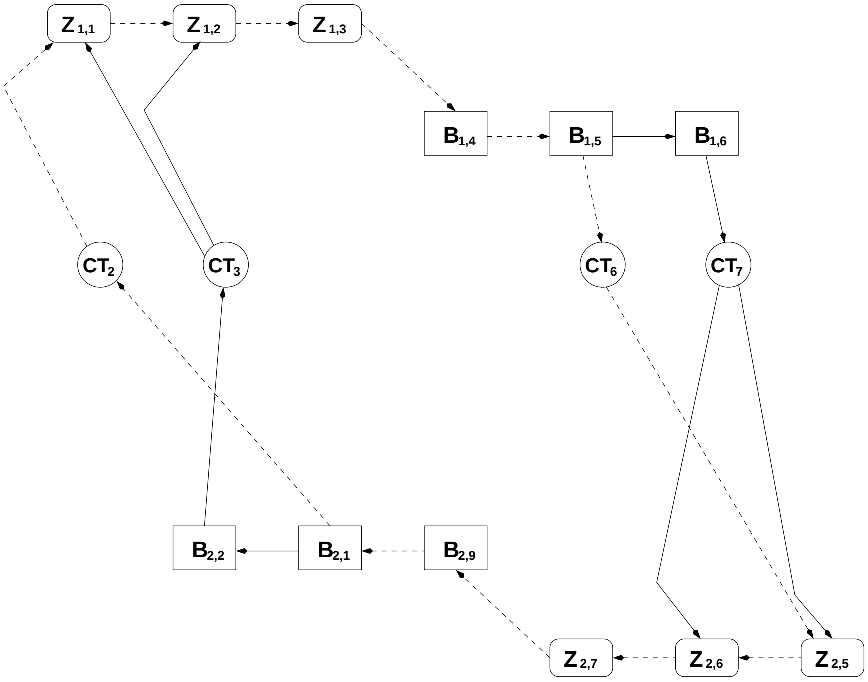

Figure 5.25: Cycles in the System Graph

5.10.1 Algorithm Description

This algorithm to detect a cycle in the system graph $\mathcal{SG}$ is adapted from the version presented by Cormen *et al.* [28] for depth first search of a directed graph.

Algorithm. Detect_Cycle

```
Main: DETECT($\mathcal{SG}$)

1. for each vertex $u \in \mathcal{SG}$
        do $\mathcal{C}_u$ = WHITE
        do $\pi_u = \{ \}$

2. for each vertex $u \in \mathcal{SG}$
        do if $\mathcal{C}_u$ = WHITE
```

```
            then VERTEX-VISIT($u$)
3. return NO
4. stop
```

Function: VERTEX-VISIT(u)

```
1. $\mathcal{C}_u$ = GRAY
2. for each $v \in \Sigma_u$
       do if $\mathcal{C}_v$ = GRAY
              return YES
              exit
       else if $\mathcal{C}_v$ = WHITE
              then add $u$ to $\pi$[v]
                   VERTEX-VISIT($v$)
3. $\mathcal{C}_u$ = BLACK
4. end
```

Note that in the algorithm and in the following explanation, $\mathcal{C}_u$ or $\mathcal{C}[u]$ refers to color[u], the color associated with vertex u.

5.10.2 Explanation

Step 1 colors all the vertices of the system graph white and sets the predecessor sets of all vertices to null. This is the initialization step. In Step 2, a vertex of the graph is picked arbitrarily and checked for color. If it is white, the VERTEX-VISIT(u) routine is invoked. Every time VERTEX-VISIT(u) is invoked, vertex u

becomes the root of a new tree in the graph. When VERTEX-VISIT(u) is called, the color of u is first set to gray. Step 2 targets the successor set for the vertex u, Σ_u. If the color of a vertex in Σ_u is white, then u is added to the predecessor set of v and the function is called recursively. If the color of vertex v in Σ_u is gray, this means that the vertex was discovered but not finished earlier implying that there is a cycle in the graph. The algorithm terminates returning a boolean that acknowledges the presence of a cycle in the graph. If after traversing the entire graph, no cycle is found, a boolean value is returned indicating the fact.

5.10.3 Example

Consider the system graph $\mathcal{SG}$ in Figure 5.17 for the processing station set shown in Figure 5.16. Algorithm: Detect_Cycle is applied to $\mathcal{SG}$. Initially the algorithm colors all vertices white and sets their π fields to NIL. Without loss of generality, let the algorithm pick vertex CT_9 as the start vertex. It calls VERTEX-VISIT(CT_9) which colors this vertex gray, i.e. $\mathcal{C}[CT_9]$ = gray. Again without loss of generality let edge between CT_9 and $Z_{2,7}$ be the edge picked for exploration. Since $\mathcal{C}[Z_{2,7}]$ = white, the algorithm adds it to the predecessor set of vertex CT_9 and calls VERTEX-VISIT($Z_{2,7}$), and this recursion continues. Figure 5.26 shows the first three steps of this recursion on a part of the system graph. Figure 5.27 shows a later position as the algorithm traverses $\mathcal{SG}$. Here, the graph traversal has reached a stage where after discovering vertex $Z_{2,6}$, the algorithm has picked $Z_{2,7}$ for analysis and seen that $\mathcal{C}[Z_{2,7}]$ = gray. The algorithm terminates at this point returning the fact that a cycle in present in $\mathcal{SG}$.

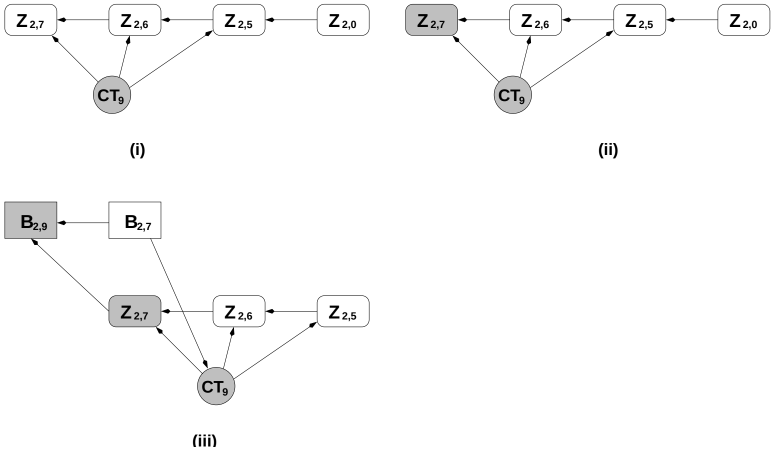

Figure 5.26: Cycle detection algorithm example - Figure 1

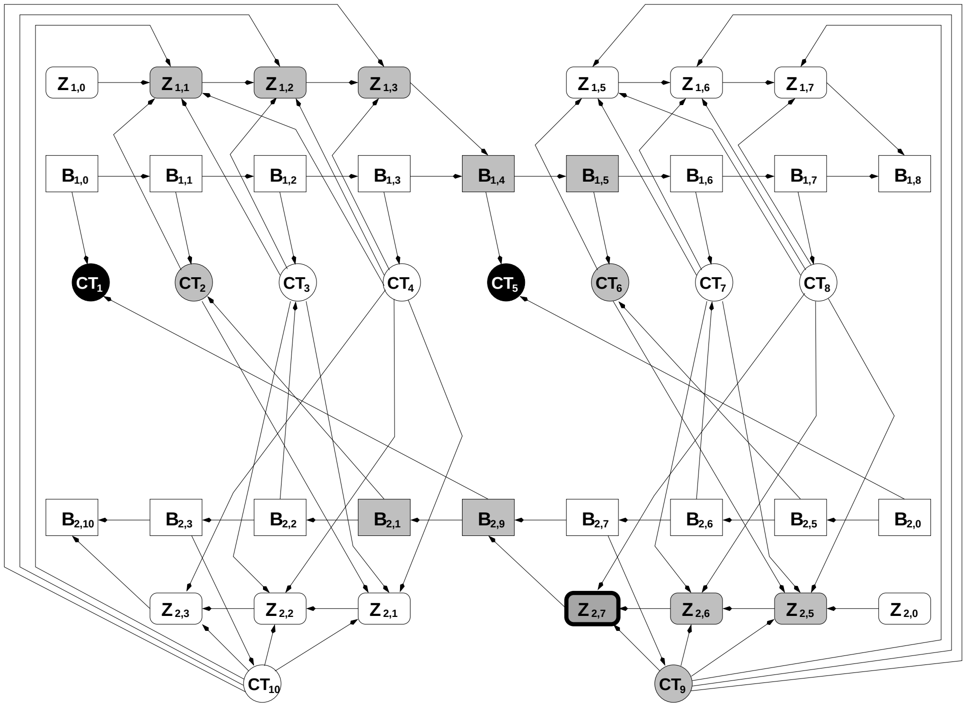

Figure 5.27: Cycle detection algorithm example - Figure 2

5.11 Calculating System Parameters for the Simple Job Shop

This section presents an algorithm to find the system parameters for a job shop scenario whose system graph has no cycle. Note that Step 3 requires Algorithm:

Detect_Cycle explained in Section 5.10.

Algorithm. Job_Shop_Cycle_Time (no cycle)

```
Main: AGGREGATION($\mathcal{I}$,$\mathcal{J} \cup \mathcal{F}$)

1. for $j \in \mathcal{J} \cup \mathcal{F}$
       compute availability $A_j$

2. create $\mathcal{SG}$

3. if exists cycle in $\mathcal{SG}$
     exit
   else continue

4. for $i \in \mathcal{I}$
       compute $x_i$ using $T_i^a$ and $B_{i0}$

5. for $n \in \mathcal{SG}$
       $\mathcal{P}_n = \Pi_n$

6. create set $\mathcal{W} = \{n \in \mathcal{SG} : P_n = \{\}\}$

7. for $n \in \mathcal{W}$
       remove $n$ from $\mathcal{W}$
       if $n \in \mathcal{B}$
          compute $B_{ij}$
       if $n \in \mathcal{Z}$
          compute $Z_{ij}$
       if $n \in \mathcal{CT}$
          compute $CT_j$
```

```
for $m \in \Sigma_n$
    $\mathcal{P}_m = \mathcal{P}_m - \{n\}$
    if $P_m = \{\}$
        add $m$ to $\mathcal{W}$
```

<u>NOTE</u>: For this algorithm, $\mathcal{P}_n$ refers to the set of predecessor nodes for node n which have not been visited ($\mathcal{P}_n \subset \Pi_n$). Also, $m \in \mathcal{P}_n$ iff $\exists$ arc from m to n in $\mathcal{SG}$. The sets $\mathcal{B}, \mathcal{Z},$ and $\mathcal{CT}$ refer to sets of type B, Z and CT nodes respectively. Note that because $\mathcal{SG}$ has no cycle, the algorithm will visit every node in $\mathcal{SG}$ once.

The number of type CT nodes is S. The effort at each node is $O(IS)$. The total effort for the type CT nodes is therefore $O(IS^2)$.

The number of type B nodes is $O(IS)$. The effort at each node is $O(S)$. The total effort for the type B nodes is therefore $O(IS^2)$.

The number of type Z nodes is $O(IS)$. The effort at each node is $O(S)$. The total effort for the type Z nodes is therefore $O(IS^2)$.

Thus, the computational complexity of the algorithm is $O(IS^2)$.

5.12 Example

Consider the product set detailed in Subsection 5.8.2. Figure 5.14 shows the processing sequences for the product set. Figure 5.15 shows the system graph. Algorithm: Detect_Cycle can be used to verify that no cycles exist in the system graph.

For more details of the product and part processing time and setup time calculations, please refer to rules and algorithms for microwave module process planning developed by Lam *et al.* [63, 92].

For this system, the sequence of steps to calculate the manufacturing cycle times for the three products, based on the algorithm presented in Section 5.11 is enumerated in Table 5.10.

Step	Calculate	Add to	Step	Calculate	Add to
No.	and remove	$\mathcal{W}$	No.	and remove	$\mathcal{W}$
	from $\mathcal{W}$			from $\mathcal{W}$	
1.		$B_{1,0}, B_{2,0}, B_{3,0},$	23.	$Z_{1,2}$	$Z_{1,3}$
		$Z_{1,0}, Z_{2,0}, Z_{3,0}$	24.	$Z_{2,2}$	$Z_{2,3}$
2.	$B_{1,0}$	$B_{1,1}$	25.	$Z_{3,2}$	$Z_{3,10}$
3.	$B_{2,0}$	$B_{2,1}, CT_1$	26.	$Z_{1,3}$	$B_{1,4}$
4.	$B_{3,0}$	$B_{3,9}, CT_9$	27.	$Z_{2,3}$	$B_{2,4}$
5.	$Z_{1,0}, Z_{2,0}, Z_{3,0}$		28.	$Z_{3,10}$	$B_{3,4}$
6.	CT_1, CT_9		29.	$B_{1,4}$	$B_{1,5}$
7.	$B_{1,1}$	$B_{1,2}$	30.	$B_{2,4}$	$B_{2,5}$
8.	$B_{2,1}$	$B_{2,2}$	31.	$B_{3,4}$	$B_{3,5}, CT_5$
9.	$B_{3,9}$	$B_{3,2}, CT_2$	32.	CT_5	
10.	CT_2		33.	$B_{1,5}$	$B_{1,8}$
11.	$B_{1,2}$	$B_{1,3}$	34.	$B_{1,8}$	
12.	$B_{2,2}$	$B_{2,3}, CT_3$	35.	$B_{2,5}$	$B_{2,8}, CT_8$
13.	CT_3		36.	$B_{2,8}$	
14.	$B_{3,2}$	$B_{3,10}, CT_10$	37.	CT_8	$Z_{1,5}, Z_{2,5}$
15.	CT_10		38.	$Z_{1,5}$	$Z_{1,8}$
16.	$B_{1,3}$	$B_{1,4}$	39.	$Z_{1,8}$	
17.	$B_{2,3}$	$B_{2,4}$	40.	$Z_{2,5}$	$Z_{2,8}$
18.	$B_{3,10}$	$B_{3,4}, CT_4$	41.	$Z_{2,8}$	
19.	CT_4	$Z_{1,1}, Z_{2,1}, Z_{3,9}$	42.	$B_{3,5}$	$B_{3,6}, CT_6$
20.	$Z_{1,1}$	$Z_{1,2}$	43.	$B_{3,6}$	
21.	$Z_{2,1}$	$Z_{2,2}$	44.	CT_6	$Z_{3,5}$
22.	$Z_{3,9}$	$Z_{3,2}$	45.	$Z_{3,5}$	$Z_{3,6}$

Table 5.10: Steps for the job shop example

The design characteristics for the products are explained in Chapter 3. Table 5.11 lists the quantity requirements for the products. These are used to determine the product release rate. Table 5.12 includes the processing properties for all the workstations in the manufacturing system.

	MWM Product 1	*MWM Product 2*	*Improved MWM*
Batch Size (parts/job)	5	10	12
Throughput (parts/day)	20	20	12
Release Rate (jobs/min) (at input)	0.0083	0.004147	0.002083

Table 5.11: Throughput requirements

Station Name	Processing Time (min)	Drift Rate	Normal Yield	Reduced Yield
Mill	41.77	0.9	0.9	0.7
Grind and Mill	12.12	0.9	0.9	0.7
Electroless Plate		0.9	0.9	0.7
Etch	54.29	0.8	0.9	0.7
Artwork	8.66	0.99	0.9	0.7
Inspection	84.29	0.9	1	1
Auto. Assembly	29.77	0.85	0.95	0.8
System Test	101.51	0.92	1	1
Manual Assembly	7.58	0.89	0.95	0.8

Table 5.12: Processing station properties

The processing time at the electroless plate station is incremented from 22.67 min to 62.67 min in steps of 5 min and the performance of the system is estimated for these values. Figure 5.28 shows a plot of the average manufacturing cycle time for *MWM Product 1*/*MWM Product 2* and *Improved MWM* as the processing time at electroless plate increases. The reduction in batch size and consequentially the throughput of the products as the processing time is not significant (about 0.1%) for this example. This is because the processing time increase occurs close to the start of the processing sequence and hence does not contribute greatly to the detect time.

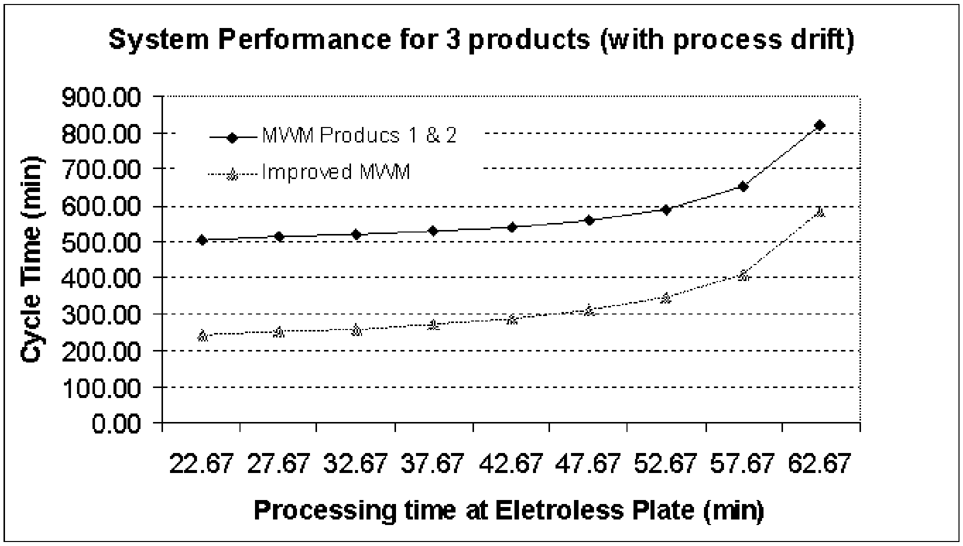

Figure 5.28: Average manufacturing cycle time for products versus processing time at electroless plate

5.13 Summary

This chapter addressed the issue of manufacturing systems that have workstations that undergo process drift. A key issue is that the process drift is detected at a subsequent inspection station (not at the station where it occurs). It presented mathematical models to calculate various performance measures for such systems, improving upon the models presented in Chapter 3. Different production systems were considered to demonstrate the utility of these mathematical models.

For flow shops, these models may be applied to calculate the output batch sizes at each station for a product's processing sequence. These may then be used for calculating the manufacturing cycle times at the stations and for the products. For the more general job shop case applying the models is more involved. When different products follow different routings in the system, there may exist interdependencies between the system and product parameters. Hence prioritizing calculations for these parameters becomes difficult. This chapter presented an approach to solve this case.

The approach defined a system graph representing the product processes and associated workstations. It asserted certain properties of this graph and suitably justified the assertions where necessary. The approach analyzed the system graph to identify possible product-system parameter interdependencies. Further, if a cycle existed in the system graph due to the parameter interdependencies, this chapter presented an algorithm to detect such cycles. The chapter presented an algorithm to determine system performance when the system graph does not contain any cycles. Suitable examples were presented to substantiate various algorithms in the chapter.

Chapter 6

Overall Impact of Reducing Manufacturing Cycle Time

This chapter addresses the overall economic benefits of reducing manufacturing cycle time for a product. These benefits range from reduced inventory to improved product supply predictions. There are economic incentives associated with each of these benefits and this chapter presents models to estimate these economic benefits. These benefits can be compared to other costs or metrics that change as a result of modifying the product design. Section 6.1 describes how reducing the manufacturing cycle time contributes to the main goals of any manufacturing enterprise:

1. reduced costs,
2. greater revenue, and
3. higher profit.

Sections 6.2 and 6.3 present mathematical models to quantify the relations represented by these edges. Section 6.4 integrates these models into a composite cost model, and Section 6.5 summarizes the chapter.

6.1 Relating Manufacturing Cycle Time to Economic Gain

Manufacturing cycle time is composed of processing times and non-processing times. The processing times depend on the manufacturing operations involved. These are governed by the type and properties of material, and the type of workstations used. Considerable work towards reducing processing times has been conducted in DFM research. Minimizing setup times has also been the focus of many researchers. Chapter 2 presented details of relevant literature pertaining to DFM tools and approaches to estimating setup times. However, reducing the non-processing components such as queue times and move times also significantly reduces total manufacturing cycle time. Lowering manufacturing cycle time for the product would have a significant positive impact on the economic returns from new product introduction [42]. More recently, Rajagopalan [106] uses the inventory cost and lead time as metrics to help a manufacturer decide whether to adopt a make-to-order or a make-to-stock strategy or a combination of both. The work presents heuristics to aid the decision process based on the expected number of setups, the anticipated lot sizes and considerations of capacity, congestion and inventory cost. They further present bounds for models based on these heuristics. The models developed, however, do not include many of the costs associated with the production process. Significant among these are the penalty costs attributed to order fulfillment on time, among other costs associated with manufacturing cycle time. The models presented in further sections of this chapter can be used to better qualify the models and the production strategy decision presented by Rajagopalan.

As discussed in Chapter 2, the product design has considerable influence on the manufacturing cycle time for the product. Changing the manufacturing cycle time has multiple impacts. Some of these impacts are direct and others are indirect. Figure 6.1 illustrates the relationships between the different impacts. In the figure, ovals represent product and process performance improvement opportunities, and rectangles indicate the end effects of such improvements. This section explains these relationships, referring to the edges of the graph in Figure 6.1. One goal of this work is to model the economic impact quantitatively. Discussing the impact of a change in manufacturing cycle time is a type of sensitivity analysis that helps identify the important issues. Note that statements about the benefits of reducing manufacturing cycle time also implicitly indicate the negative impacts of increasing it.

Reducing manufacturing cycle time for a product results broadly in the following benefits:

1. Lower inventories,

2. Process improvements,

3. Product improvements, and

4. Better order fulfillment.

These lead to more profitable products. To describe the numerous direct and indirect benefits of reducing manufacturing cycle time, the relationship model presented in Figure 6.1 was developed as part of this research. The following subsections explain various edges and components in Figure 6.1 in detail along with their

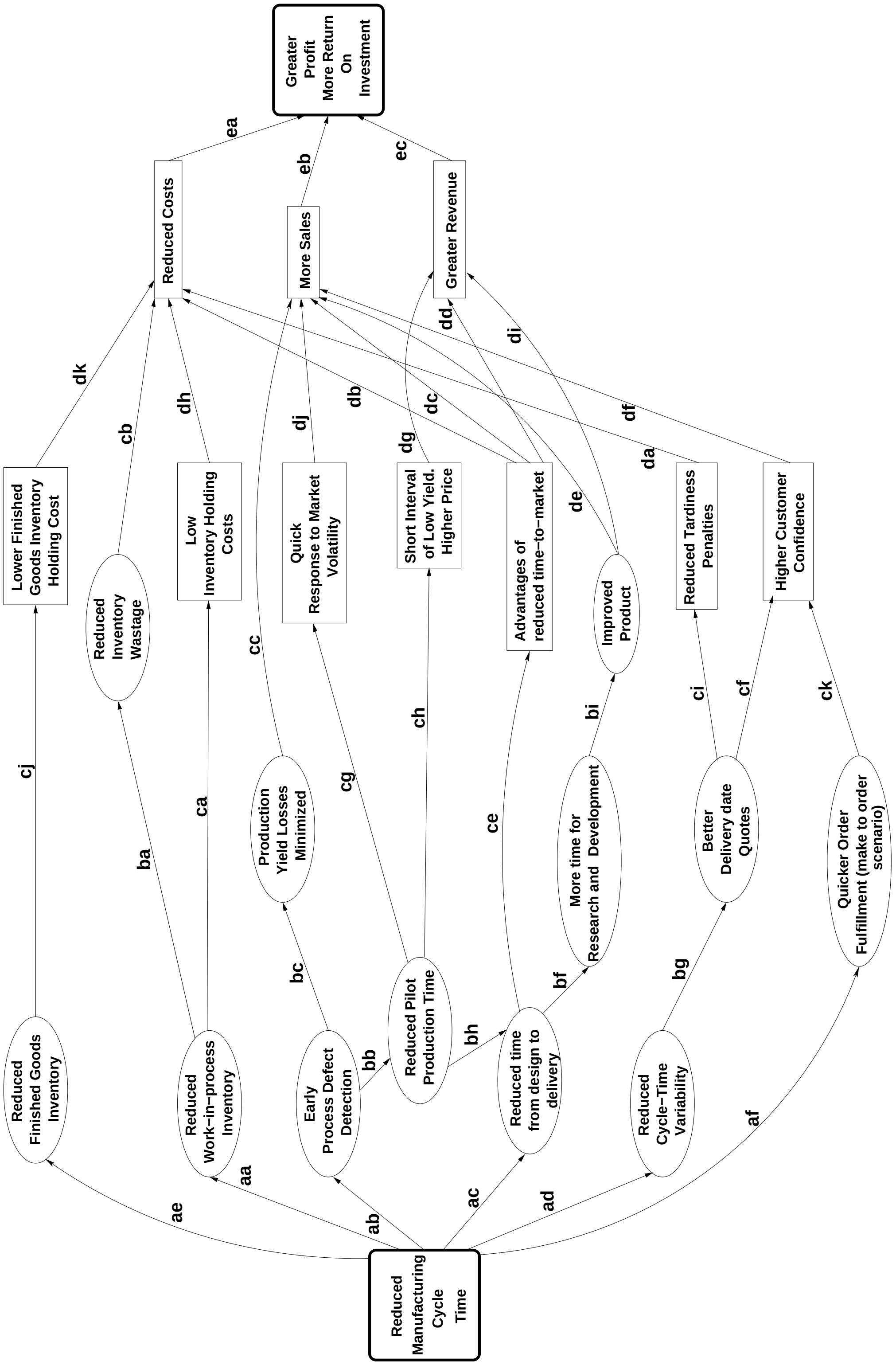
Greater Profit More Return On Investment
ea
eb
ec
Reduced Costs
More Sales
Greater Revenue
dk
cb
dh
dj
db
dg
dc
dd
de
di
da
df
Lower Finished Goods Inventory Holding Cost
Reduced Inventory Wastage
Low Inventory Holding Costs
Quick Response to Market Volatility
Short Interval of Low Yield. Higher Price
Advantages of reduced time−to−market
Improved Product
Reduced Tardiness Penalties
Higher Customer Confidence
cj
ba
ca
cc
cg
ch
ce
bi
ci
cf
ck
Production Yield Losses Minimized
More time for Research and Development
Better Delivery date Quotes
Quicker Order Fulfillment (make to order scenario)
bc
bb
bh
bf
bg
Reduced Pilot Production Time
Reduced Finished Goods Inventory
Reduced Work−in−process Inventory
Early Process Defect Detection
Reduced time from design to delivery
Reduced Cycle−Time Variability
ae
aa
ab
ac
ad
af
Reduced Manufacturing Cycle Time

Figure 6.1: Map from reduced manufacturing cycle time to economic gains

significance. Taken together, these relationships show how reducing manufacturing cycle time increases product profitability.

6.1.1 Lower Inventories

Inventory causes a manufacturing organization to commit to costs associated with the inventory quantity, its value, and the length of time that the inventory is carried. By committing capital to holding inventory, the organization loses the opportunity of using these funds for other purposes such as acquiring equipment and improving or developing products. Thus the cost of investment or capital cost is incurred from the inventory investment. Along with capital cost, Vollmann *et al.* [139] list other costs associated with holding inventory, some of which include costs of inventory obsolescence and operating costs involved in storing inventory. In addition to these direct costs of holding inventory, there are associated indirect overhead costs such as preparation costs and personnel costs. Perhaps the most significant effects of inventory are on customer service, order fulfillment and shortage. These are explained in detail in further sections.

edge aa: Reduced cycle time results in reduction in work-in-process inventory for a given throughput. This is substantiated by Little's Law [65].

edge ba: Reduced work in process inventory means that whenever a new product is introduced, and production of an old product is suspended, there is less waste from the incomplete jobs (of the earlier product) in-process which are rendered useless [85].

edge cb: Reducing the inventory waste reduces the cost associated with the material and processing of these products thus lowering the losses to the manufacturer and reducing a contributor to the product cost [85].

edge ca: Reduction in the work-in-process inventory means that the costs associated with holding this inventory are also reduced [139]. See Section 6.2 for a more detailed explanation of the costs associated with holding WIP as also models to estimate this cost.

edge dh: The cost of holding WIP contributes to the total inventory holding cost. Inventory holding costs contribute to the total cost of the product. Hence WIP holding cost reduces the overall product cost.

edge ae: Reducing the manufacturing cycle time for a product results in a lower lead time demand. This in turn means that it is possible to maintain a lower finished goods inventory for the product. See Section 6.3 for a detailed explanation for this edge.

edge cj: A lower finished goods inventory means that the finished goods inventory carrying cost reduces. See Section 6.3 for further explanation and model.

edge dk: The finished goods inventory carrying cost contributes to the total inventory holding cost. Inventory holding costs contribute to the total cost of the product. Hence minimizing the finished goods inventory cost reduces the overall product cost.

6.1.2 Process Improvements

The following edges and portions of Figure 6.1 explain the benefits of estimating and reducing manufacturing cycle time to better manufacturing processes.

edge ab: Reduced cycle time allows for earlier detection of any problems that might be present in the manufacturing process.

edge bc: Quicker feedback ensures that any defects in the process are rectified immediately, thereby minimizing the quantity of defective products (yield loss) [96].

edge cc: Lower yield losses reduce the number of faulty parts, and hence a larger percentage of jobs started are available for sale [96].

edge bb: Early defect detection not only increases the production quickly during the pilot production phase but also ensures faster process feedback which is very critical [96].

edge cg: In a dynamic market scenario, reduced manufacturing cycle time and hence product development cycle time means that the manufacturer is able to cope with market volatility effectively. This is especially true for products like semiconductor wafers that have such dynamic markets where not only does the product type change frequently but the demand also fluctuates considerably [96].

edge dj: A quicker response to changes in the market demand means that the changing requirements of the market can be met more easily and earlier, resulting in higher sales [25].

edge ch: The nature of products like semiconductor wafers is such that the selling price decreases rapidly over time. Hence, the product returns are maximum closest to the launch and it is necessary to supply as much of the product immediately after launch. An added advantage to the manufacturer is the possibility of charging a higher price during this period due to a lack of competent competition.

edge dg: Increased production with higher prices during the early phases of product introduction results in higher revenue [96].

6.1.3 Product Improvements

The following edges and portions of the map present how reducing manufacturing cycle time can result in better product development resulting in higher gains.

edge ac: Reducing manufacturing cycle time means that the product completes manufacturing faster. Manufacturing cycle time is a component of the total product development time. Hence reducing this time reduces the total development time (from conceptualization to actual delivery to the customer). Figure 6.2 shows the reduction in the total development time due to reduction in manufacturing cycle time.

edge bh: Reducing time for pilot production runs means that the total product development time to which it contributes is also reduced [25, 96].

edge ce: A shorter overall product development time ensures a reduced time-to-market for the product.

edges db, dc, dd: Faster time-to-market implies that the advantages inherent to short times-to-market are available to the manufacturer. Some of the direct benefits of reduced time to market include reduced costs, more sales and revenue. These are well documented in literature [50, 51, 121]. For a domain such as wafer fabrication, time-to-market assumes a different dimension. There exists scope of dictating the price of the product for a while after launch till the competition enters the market [25, 33, 143]. See also Section 2.6 for a detailed explanation of some of the benefits of reduced time-to-market.

edge bf: Cohen *et al.* [25] study another interesting aspect of an efficient development cycle, stressing that product improvement is more valuable than unnecessarily early introduction into the market. Reducing manufacturing cycle time provides more time for research and development while maintaining the same product introduction time. Figure 6.2 shows this added time for the same development time.

edge bi: Based on the recommendations from the study conducted by Cohen *et al.* [25], since the product development team has more time to improve the product before actual market introduction, the product released to the market

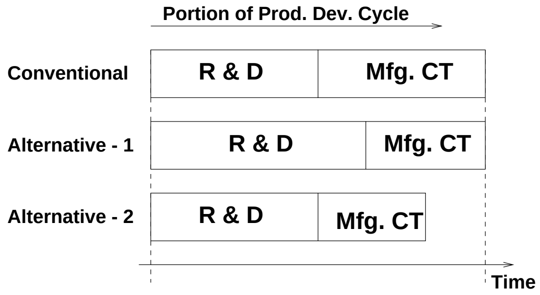

Figure 6.2: Lower manufacturing cycle time allows more time for research and development

performs better and is more desirable.

edge de: An improved product would result in higher product sales due to a better product being delivered to the customers in the same time frame [25]. (Note that this would also be influenced by the final price of the product).

Yoon and Kijewski [145] hypothesize that a manufacturer's price of a product is positively significantly affected by the customer's evaluations of the product's overall quality. Further, it is significantly associated with the availability of the product's main functional features. It follows from the hypothesis that the product pricing is a function of the product's main functional features and the advantages due to the product's auxiliary features.

edge di: With an improved product, depending on its quality relative to that of competitive products, the manufacturer can expect to charge a higher price, resulting in increased revenue [33, 96]. This in turn means that a new product in the market that provides both, the requisite and promised primary functional features and significant advantages of its auxiliary features arising from its use, promises to attract the target customer base at a higher price.

It is important to note another result from the paper by Yoon and Kijewski [145]. Based on regression analysis on the product-pricing data for a consumer product, the paper finds that this price-product feature association diminishes in a mature product/market environment. This may be attributed to the fact that, in time, all product brands have similar functional and convenience features and the price becomes a function of other factors.

The significance of Yoon and Kijewski's findings is that the presented hypotheses corroborate what this research proposes; reducing the manufacturing cycle time for a product leads to a dual benefit, higher gains due to more development time and higher revenue from synthesizing a better product before the competition. Alternative-1 in Figure 6.2 shows how a lower manufacturing cycle time for the product allows for more time to be spent on research and development. Thus, based on their hypothesis, this additional development time for a new product can command a higher price in the market as well as expect to sell more.

6.1.4 Better Order Fulfillment

Reducing manufacturing cycle time results in better order fulfillment. This subsection presents the associated sections of Figure 6.1.

edge ad: The cycle time depends on a number of manufacturing system and product variables some of which are random in nature. As the cycle time for the product increases, the variance of the manufacturing cycle time increases [65]. This is clearly seen in the case of a manufacturing system approximated by a M/M/1 queue wherein the distribution of the processing and arrival times for the jobs are exponential. For an exponential distribution, the mean and variance are both functions of the scale parameter only. Hence a low mean means a low variance as well. Similar arguments can be presented for other distributions such as gamma (Erlang), Poisson, and Weibull distributions.

edge bg: Increasing the variability of the cycle time increases the error in estimating the lead time for product delivery, making it harder to quote an accurate delivery date to the customer. By reducing the variability an accurate lead time can be quoted [65].

edge ci: Delivering the product to the customer in accordance with the quoted delivery dates ensures that any tardiness penalties that may be applicable are avoided.

edge da: The tardiness penalties translate into direct costs to the manufacturer. Hence, avoiding or minimizing these penalties results in a reduction in cost.

edge cf: Delivering goods as per the date promised (accurate lead time quotes) increases the confidence of the customer and also helps maintain customer loyalty [46].

edge af: Reducing the manufacturing cycle time means that the product can now be delivered sooner [33, 143]. In a make-to-order kind of product scenario, this means that the orders can be fulfilled and sent to the customer quickly, often before the due date.

This is especially important in industries such as personal computers and automobiles where customers order customized products that are made on high volume production lines. Such customized products may be variants of a basic platform product design.

edge ck: Early fulfillment of orders means that the customer has more confidence in the vendor to deliver on-time. This boosts the customer confidence and with it the probability of securing a repeat order.

edge df: Higher customer confidence leads to an increase in sales. Also customer loyalty ensures steady product sales and a somewhat assured market for any new product.

6.1.5 Higher Profitability

Thus, the manufacturing cycle time has a considerable influence on the revenue for the product. A reduction in the manufacturing cycle time has a threefold effect - reducing the product costs, increasing the product sales and increasing the revenue from increase in sales and higher prices. All these factors lead to an increase in the profits and the return on investment.

edges ea, eb, ec: The product profit or return on investment is usually defined to be the difference between the total revenue and the total costs. Reducing the costs and increasing the sales and hence the revenue amounts to increasing the profits or the return on investment for the manufacturer.

Figure 6.1 represents the overall benefits of reducing manufacturing cycle time for the product. Not all the benefits depicted here will hold true for all production scenarios or for all aspects of the product development process. For example, if the benefits of a reduced manufacturing cycle time during only the production phase are being considered, the portion of the map referring to the benefits of reduced pilot production time may not be relevant. Similarly, some benefits are valid for make-to-stock scenarios while some others are relevant only to make-to-order production systems.

The further sections of this chapter present mathematical models to quantify some of the relationships represented by edges in Figure 6.1. The overall impact may be modeled as a composition of these models.

6.2 Work In Process Inventory

This section addresses the cost associated with holding WIP which essentially accrues from,

1. investment costs for material and overheads for the WIP, and
2. space considerations for stocking the WIP on the shop-floor.

These costs are proportional to the quantity of WIP in the manufacturing system.

The relationship between cycle time and the work in process (WIP) inventory is given by Little's Law which states that: The fundamental long-term relationship between Work-In-Process, Throughput and Cycle Time of a production system in steady state is

$$\text{WIP} = \text{Throughput} \times \text{Cycle Time} \tag{6.1}$$

There are two general requirements for Little's Law to hold true [65],

1. inventory, throughput and flow time must represent long-term averages of a stable system, and

2. inventory, throughput and flow time must be measured in consistent units.

Little's Law applies to single stations, production lines, factories, and entire supply chains. It applies to systems with and without variability. The law holds true for single and multiple product systems. Thus, since WIP in the system is a direct function of the manufacturing cycle time, as manufacturing cycle time increases, WIP also increases.

As the WIP increases, the cost associated with holding WIP also increases. A product is normally a composition of various components or parts. The processing sequence of the product includes the processing of the components and some joining operations. For a major part of the product's processing sequence, these components are processed independently. The total WIP of the product is an aggregate of the WIP associated with various product constituents. Let

C_i^w = total cost of holding WIP for product i, (\$/time unit)

$\overline{W_{i_k}}$ = average WIP of component k of product i, (parts)

C_{i_k} = holding cost of component k of product i, (\$/[part - time unit])

The total cost of holding WIP for the product may hence be modeled as shown by Equation 6.2,

$$C_i^w = \sum_{k=1}^{m} \overline{W_{i_k}} C_{i_k} \tag{6.2}$$

Note that the holding cost for WIP incorporates costs associated with storing inventory on the shop floor, cost of wastage due to contamination of WIP (which is highly relevant for products such as semiconductors), storing the WIP in off-shopfloor locations and associated material movement costs. The proposed cost in Equation 6.2 models the cost represented by **edge ca** in Figure 6.1.

6.3 Finished Goods Inventory

Subsection 6.1.1 described the relationship between reducing manufacturing cycle time for a product and lower finished goods inventories. This section presents models to quantify this relationship and calculate the associated cost.

6.3.1 Q-R Policy and Shortages

The basic two decisions to be made for inventory management are

1. how much to order, and
2. when to order.

There are various inventory rules in literature discussing how to make these decisions. This research looks at the Q-R rule and the implications of cycle time length and variability on this policy. Under the Q-R rule, an order for a fixed quantity (Q) is placed whenever the stock reaches the reorder point (R^e). Until the order arrives, inventory continues to fall. Figure 6.3 shows a plot of the inventory levels versus time and indicates these quantities.

The algorithms for calculating the optimal values of the reorder quantity and time assume conditions of fixed demand rate and constant replenishment time.

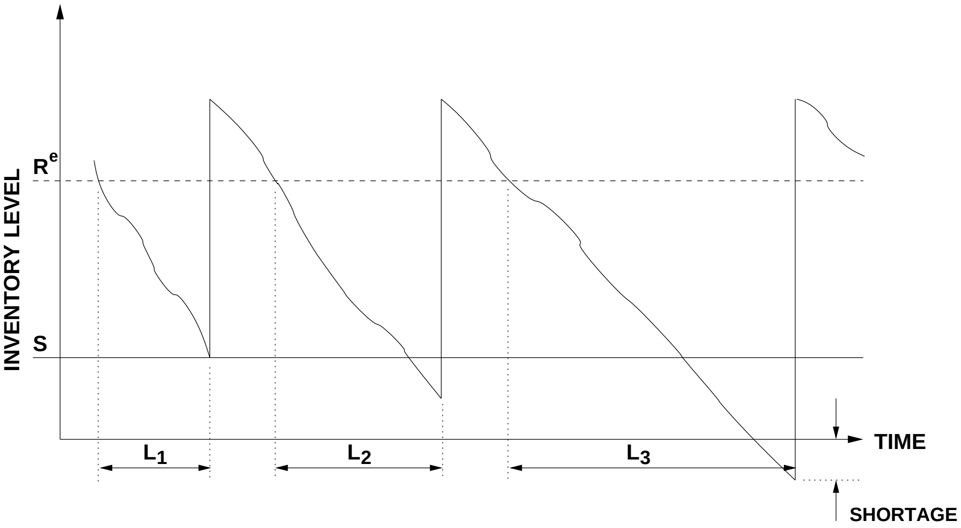

Figure 6.3: Q-R Policy and Cycle Time Variability

However, in actual operations, due to random fluctuations in demand for the product, this is rarely true. Significant inventory related costs may be incurred when the demand exceeds the available inventory for a product. The costs associated with such stock outs are called *shortage costs.* The effects of shortage costs are multi-fold:

1. There is an immediate loss of revenue due to the inability of the manufacturer to deliver the goods.

2. The void caused by this inability is most likely filled by a competitor who is able to deliver the needed products in the defined time frame, so sales may be lost.

3. Lack of promised delivery on time maligns the firm's reputation and customer goodwill may be lost.

Thus, in order to inoculate against such inventory shortages when there is uncertainty in demand, the reorder point must be greater than average demand during the replenishment lead time. The difference between the reorder point and the average demand in this period is called the *safety stock* S. The quantity of safety stock is determined by the magnitude of risk involved in the possibility of a stock-out.

The discussion thus far has focussed on the possibility of shortages and stock-outs due to fluctuations in the demand for the product during the replenishment lead time. However, in addition to this uncertainty, there is also the possibility of delay in the replenishment lead time owing to lead time variability. Figure 6.3 shows variability among the lead times L_1, L_2 and L_3. The lead time variability is dominated by the variability in manufacturing cycle time for the product. The influence of the manufacturing cycle time variability and the uncertainty in demand on inventory levels is analyzed in Subsection 6.3.2.

Causes of manufacturing cycle time variability, its implications and models to quantify this variability were presented in Chapter 3. One effect of this variability in the manufacturing cycle time is the need to maintain more safety stock in order to prevent shortages due to increased order fulfillment time. This in turn means that the firm now needs to maintain higher levels of finished goods inventory, bearing the associated costs.

6.3.2 Demand and Cycle Time Variability

Often, production models developed for shop-floors in a manufacturing environment consider the uncertainty in the demand forecast. As explained earlier, various mechanisms such as safety stock may be used to account for the lack of accuracy resulting from this variability. However, there is another uncertainty viz. that associated with the lead time of the product, which is often overlooked. A dominant part of the lead time uncertainty is the variability associated with the manufacturing cycle time for the product. These variabilities create a need for the company to maintain a larger finished goods inventory and in some cases may even lead to shortage situations. To calculate total product cost, it is necessary therefore for the product development team to consider costs associated with the effects of these variabilities as well. This issue is addressed by edges **ae** and **cj** of Figure 6.1.

Now, the expected value and variance in manufacturing cycle time for product i are given by

$$E[CT_i] = \sum_{j \in R_i} E[CT_j^\star] \tag{6.3}$$

$$\mathrm{Var}[CT_i] = \sum_{j \in R_i} \mathrm{Var}[CT_j^\star] \tag{6.4}$$

where

$$\mathrm{Var}[CT_j^\star] = \text{manufacturing cycle time variance at resource } j$$

The following discussion in this section will drop the subscript i since the discussion refers to only one product.

Considerable literature on product development and new product introduction assumes the demand per unit period (D) to follow a normal distribution. This work makes a similar assumption. For this distribution, let

$$E[D] = \mu_D \tag{6.5}$$

$$\mathrm{Var}[D] = \sigma_D^2 \tag{6.6}$$

The resultant distribution for the lead time demand H for a product has mean and variance given by [94]:

$$\hat{\mu} = E[H] \;=\; \mu_D \sum_{j \in R} E[CT_j^\star] \tag{6.7}$$

$$\hat{\sigma}^2 = \mathrm{Var}[H] \;=\; \left[\sigma_D^2 \sum_{j \in R} E[CT_j^\star] \;+\; \mu_D^2 \sum_{j \in R} \mathrm{Var}[CT_j^\star]\right] \tag{6.8}$$

Consider the case where the production process for the product under analysis is a Markov process (in order to use the well-defined relationship between the mean and variance of the distribution). The model can be extended to the more general distributions for the production process by using corresponding relationships between the mean and variance for the distributions. Thus, for each workstation in the manufacturing system the total cycle time for that workstation is exponentially distributed ($\mathrm{SCV}_j = 1$) [80]. Therefore,

$$\mathrm{Var}[CT_j^\star] = E[CT_j^\star]^2 \tag{6.9}$$

Hence, the resultant distribution for the lead time demand H has mean and variance:

$$E[H] \;=\; \mu_D \sum_{j \in R} E[CT_j^\star] \tag{6.10}$$

$$\mathrm{Var}[H] \;=\; \left[\sigma_D^2 \sum_{j \in R} E[CT_j^\star] \;+\; \mu_D^2 \sum_{j \in R} E[CT_j^\star]^2\right] \tag{6.11}$$

It is seen from Equations 6.10 and 6.11 that the variance of the lead time of the order release is a direct function of the mean and variance of the manufacturing cycle time (in this case a function of the expected manufacturing cycle time alone). This means that as the mean value of the cycle time increases, the variance of the lead time or in effect the order release increases. This creates the need to maintain higher safety stock levels in order to prevent shortages, increasing the required finished goods inventory.

6.3.3 Calculating Associated Costs

To calculate the costs involved in carrying this additional finished goods inventory, consider the expression for total annual cost of making a product as presented by Vollmann *et al.* [139]. This may be modified to calculate the finished goods holding and ordering cost over each production horizon,

$$C^{fg} = \frac{P}{Q}(C^p + C^s E[s]) + C^h \left[\frac{Q}{2} + (R^e - E[H])\right] \tag{6.12}$$

where,

C^{fg} = finished goods holding and ordering cost, (\$/time unit)

P = annual demand, (parts/time unit)

Q = order size, (parts)

C^p = fixed ordering cost, (\$)

C^h = finished goods inventory carrying cost (annual), (\$/part)

C^s = shortage cost per unit, (\$/part)

$E[s]$ = expected shortages in inventory per order, (parts)

$$
\begin{aligned}
E[H] &= \text{expected demand during the lead time, (parts)} \\
R^e &= \text{re-order level, (parts)}
\end{aligned}
$$

The term $(R^e - E[H])$ in Equation 6.12 is the difference between the reorder level and the expected demand during replenishment lead time. This term, as explained in Subsection 6.3.1, is the safety stock, S. The safety stock may be either a constant for the given product, fixed through a policy decision by the manufacturing firm, or a function of the demand. If the level of safety stock is fixed, it is usually calculated depending on the probability of a stock-out and the consequences of the stock-out. In other instances, the level of safety stock is calculated as a function of the demand by setting a reorder point and establishing the percentage of demand that can be supplied directly out of inventory [139].

Note that the shortage costs included here arise from not only penalties for late order fulfillment but also depleted customer confidence from the tardiness, represented by edges **ad, bg, bd, ci, cf, cd, da,** and **df** in Figure 6.1.

When the lead time demand is normally distributed, the expected number of shortages that occur may be computed as,

$$E[s] = \int_{R^e}^{\infty} (x - R^e)\, f(x)\, dx \tag{6.13}$$

$$\therefore E[s] = \int_{\frac{R^e-\hat{\mu}}{\hat{\sigma}}}^{\infty} \hat{\sigma} \left(x - \frac{R^e - \hat{\mu}}{\hat{\sigma}} \right) \phi(x)\, dx \tag{6.14}$$

where,

$$\begin{aligned} f(x) &= \text{probability density function of lead time demand } H\text{, and} \\ \phi(x) &= \text{standard normal probability density function.} \end{aligned}$$

Using the standardized loss function $L(z)$, defined as,

$$L(z) = \int_z^\infty (x - z)\,\phi(x)\,dx \tag{6.15}$$

the expected shortages may be calculated as follows [94]:

$$E[s] = \sigma\, L(z) \tag{6.16}$$

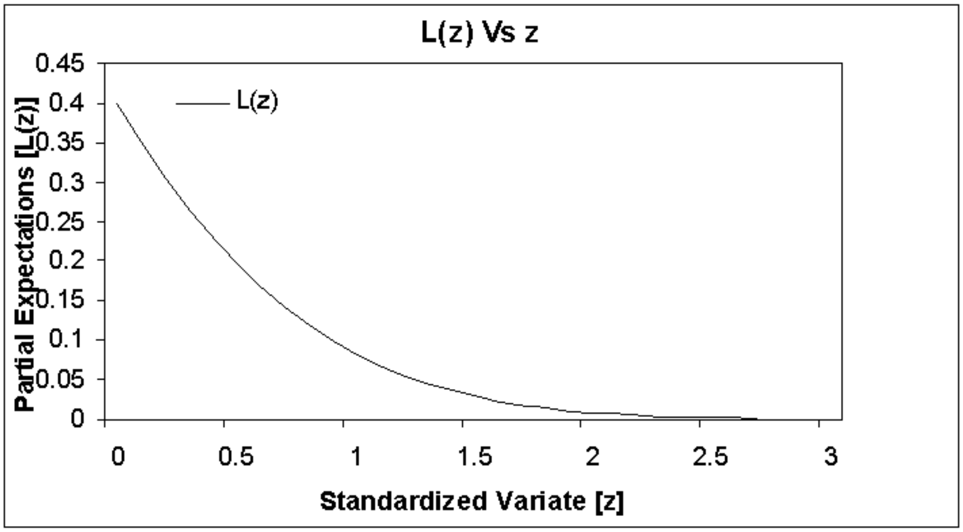

Figure 6.4: $L(z)$ as a function of z

Values for the standardized loss function $L(z)$ may be obtained from standard tables as a function of the standardized variate, z. Figure 6.4 shows the shape of the standardized loss function as a function of the standardized variate.

$$z = \frac{R^e - \hat{\mu}}{\hat{\sigma}} \tag{6.17}$$

$$= \frac{R^e - E[H]}{\sqrt{Var[H]}} \tag{6.18}$$

Therefore,

$$E[s] = \sqrt{Var[H]}\ \ L\left(\frac{R^e - E[H]}{\sqrt{Var[H]}}\right) \tag{6.19}$$

and,

$$C^{fg} = \frac{P}{Q}\left[C^p + C^s\sqrt{Var[H]}\ L\left(\frac{R^e - E[H]}{\sqrt{Var[H]}}\right)\right] + C^h\left[\frac{Q}{2} + S\right] \tag{6.20}$$

6.3.4 Illustrative Example

This example is designed to demonstrate the effects of increasing manufacturing cycle time on the costs explained in the earlier subsection. Consider a scenario similar to he scenario explained in Section 3.3.

Consider a make to stock production scenario for this product. Let the demand over the production horizon for the product follow a normal distribution. The company follows a Q-R policy for order replenishment. Table 6.1 shows the values of inputs to the model for this example. Note that for this example, the reorder level is fixed at 120 units. The safety stock, which is equal to $R^e - E[H]$, changes as the value of $E[H]$ changes. Equation 6.12 is used to calculate the inventory holding and ordering cost, C^{fg}. Table 6.1 lists the values of the product and manufacturing system constants.

Quantity	Value
E[D]	500.282 units
Var[D]	115.69 units
Cost for holding inventory	\$0.15 /unit
Fixed ordering cost	\$10
Penalty for shortages	\$20
Q	10
R^e	120

Table 6.1: Values of input costs and system constants

In the example, the desired throughput for the three products is adjusted so that the mean manufacturing cycle time for each product increases from $E[CT] = 1021.48\,\text{min}$ to $E[CT] = 1730.13\,\text{min}$. The variance Var$[CT]$ increases from 3.155×10^5 min^2 to 1.404×10^6 min^2. Table 6.2 lists the values of quantities in the cost model that change.

Desired Throughput Product (parts/day)			$E[CT]$ (min)	Var$[CT]$ (min^2)	$E[H]$ (parts)	Var$[H]$ (parts^2)	$E[S]$ (parts)	C^{fg} (\$)
1	2	3						
16	16	15	1021.48	3.155 $\times 10^5$	42.59	558.187	0.009	1022.38
16	16	17	1149.54	4.425 $\times 10^5$	47.92	780.131	0.045	1056.84
15	15	20	1243.57	5.568 $\times 10^5$	51.84	979.708	0.165	1175.96
15	18	18	1332.78	6.899 $\times 10^5$	55.56	1211.915	0.439	1449.87
15	15	22	1509.74	9.636 $\times 10^5$	62.94	1689.294	1.550	2560.26
15	23	15	1659.11	1.296 $\times 10^6$	69.17	2268.483	3.501	4511.61
16	17	20	1730.13	1.404 $\times 10^6$	72.13	2456.555	4.510	5511.41

Table 6.2: Values of derived quantites

The cycle time thus increases 69.38%, while the inventory holding cost increases from \$1022.38 to \$5511.84, an increase of 439.11%. Figure 6.5 presents graphical results for cost of ordering and holding additional inventory based on the model in Subsection 6.3.3 as the manufacturing cycle time for the product increases. For the graph, the resultant cost accrued from holding additional inventory, reorder cost and cost from penalties due to shortages is plotted on the Y-axis. The manufacturing cycle time is the X-axis quantity. The increasing cost is attributive mainly to an increased probability of shortages due to increase in the mean and variance of the manufacturing cycle time. Severe penalties associated with an inability to fulfill a promised order drive up the cost of the product.

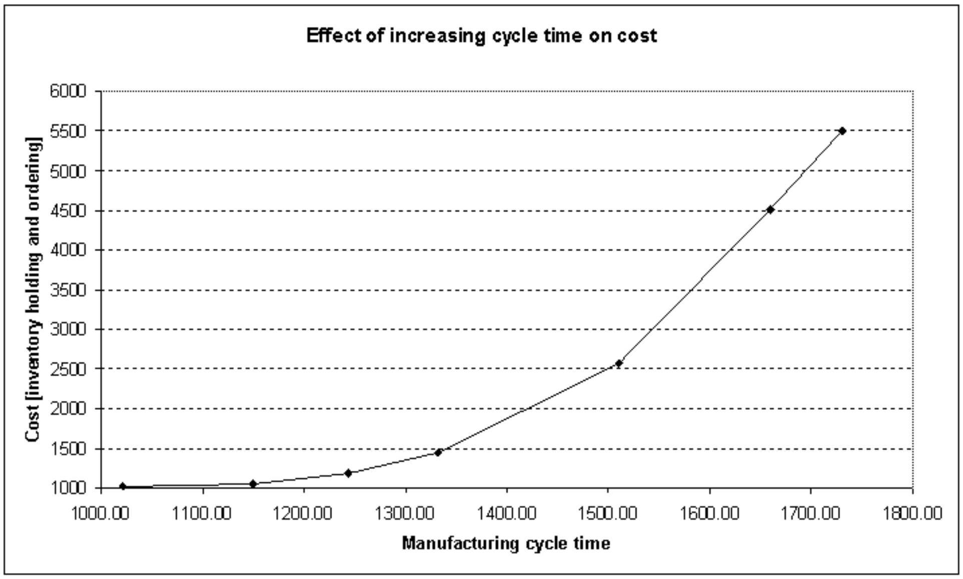

Figure 6.5: Cost affected by increased cycle time

6.4 Composite Model

The model for the overall cost and revenue associated with manufacturing cycle time for a product is a composition of the different models presented in this chapter. The profit from introducing a new product is the difference between the cumulative revenue and the total cost. This work has concentrated on the costs and revenue associated with the manufacturing cycle time for a product and this is reflected in the model presented below. It must be remembered that overall product cost and revenue comprises a number of costs associated with different portions of the product development cycle and revenue benefits from these phases. The manufacturing cycle time costs and revenue benefits modeled here must be aggregated with other costs and revenue contributors to realize the total profitability for the product.

$$Q_i = T_i^o \, P_i \tag{6.21}$$

$$C_i = C_i^{proc} + C_i^w + C_i^{fg} \tag{6.22}$$

$$P_i^r = Q_i - C_i \tag{6.23}$$

where,

$$C_i^{proc} = x_i B_{i0} C_i^m \; + \sum_{j \in R_i} x_i B_{ij} C_{ij}^p \tag{6.24}$$

Q_i = revenue from product i, (\$/time unit)

C_i = cost component for product i affected by the product manufacturing cycle time, (\$/time unit)

P_i^r = expected profit from product i, (\$/time unit)

P_i = expected price for product i, (\$/part)

T_i^o = actual throughput of product i at the end of R_i (calculated using Equation 5.31), (parts/time unit)

C_i^w = cost associated with WIP (Equation 6.2), (\$/time unit)

C_i^{fg} = finished goods holding and ordering cost (Equation 6.12), (\$/time unit)

C_i^{proc} = processing cost for product i, (\$/time unit)

C_{ij}^p = process cost of step j, (\$/time unit)

C_i^m = material cost for product i, (\$/part)

Here, the cost from additional finished goods, modeled by Equation 6.12 accounts for the additional costs associated with holding finished goods inventory, shortage costs and reorder costs.

6.5 Summary

This chapter explained the economic impact of reducing manufacturing cycle time for a product. Putting the advantages of minimizing manufacturing cycle time in perspective of economic gains in terms of reduced costs and increased sales helps the product development team understand the impact of reducing manufacturing cycle time and reinforces the need for the design for production approach advocated throughout this dissertation.

The latter part of the chapter focussed on developing mathematical models for quantifying some of the advantages of reducing manufacturing cycle time for a product. These include but are not limited to benefits resulting from reduced inventory, reduced variability in defect detection time, reduced time to market, increased time for other development activity. The overall benefit of reducing manufacturing cycle time for the product is a composition of these models.

Chapter 7

Summary and Conclusions

This chapter summarizes the work carried out as a part of this research effort. It then lists significant research contributions of the work and presents opportunities and directions for future work to improve upon and develop the work presented here towards improving the product development process. The chapter ends with some concluding remarks on the DFP methodology presented throughout this dissertation and its application to practical product design.

7.1 Summary

This research developed a set of approaches that determine how manufacturing a new product design affects the performance of a manufacturing system.

Along with other DFX techniques, product development teams need, early in the product development process, methods that can estimate the manufacturing cycle time of a given product design. If the predicted manufacturing cycle time is too large, the team can reduce the time by redesigning the product or modifying the production system. Estimating the manufacturing cycle time early helps reduce the total product development time (and time-to-market) by avoiding redesigns later in the process. Design for production (DFP) methods evaluate a product design by comparing its manufacturing requirements to available capacity and estimating manufacturing cycle time. DFP methods can be used concurrently with DFM. DFP during conceptual design can determine the capacity and manufacturing cycle time savings that result from reducing the part count. Design for production (DFP) includes design guidelines, capacity analysis, and estimating manufacturing cycle times. Performing these tasks, like DFM and other DFX techniques, early in the product development process can reduce product development time. This research classifies much of the relevant existing research on design for production methods and approaches and creates a comprehensive DFP approach which would help the designer understand the impact of the product design on the manufacturing system and aid in designing a better product.

As part of this research, a decision support tool that performs DFP analysis has been developed. Unlike previous approaches, the tool quantifies how introducing a new product increases congestion in the manufacturing system. This requires only the critical design information needed to create a process plan and estimate processing times, so it can be used early in the product development process. Determining the sensitivity of the manufacturing cycle time to various parameters is an important step to providing feedback on how the product design or the manufacturing system could be changed to improve cycle time performance. This research developed approaches that use this information intelligently and make suggestions on product redesign and manufacturing system improvements.

The tool employs an approximate queuing network based model, elaborated upon in Chapter 3, to estimate the manufacturing cycle time of the new product.

It calculates the capacity requirements and estimates the average work-in-process inventory. This provides feedback that the product development team can use to reduce manufacturing cycle time. The tool can quickly evaluate changes to the new product design or changes to the manufacturing system. The tool assumes that the manufacturing system is in steady state, i.e. it will complete a large number of batches of the new product, for a given production horizon. No batch visits a station more than once. This model assumes that the product mix and the workstation availability do not change significantly over a time horizon. Note that forecasts of product mix and workstation availability may change during the product development process. The DFP analysis should be updated when new information becomes available. The chapter also included examples of applications of the DFP approach to the domain of microwave module manufacturing. Plots of manufacturing cycle time for the products and station utilization illustrate the impact of the new product on the manufacturing system and the existing product set.

This work developed models for evaluating how embedding passives into a PCB affects not only the processing times at each step in the manufacturing process but also the overall manufacturing system behavior. The product processing time models, included in Chapter 4, are functions of the product design parameters and were developed based on literature on PCB manufacturing and discussions with experts in the field. The processing times change as the number of embedded passives changes. Values of certain other PCB design parameters are also affected by the number of embedded passives and these also affect the processing times at various processing stations.

The results presented in Section 4.4 illustrate how the model works and give some ideas of the types of tradeoffs that need to be considered while deciding the percentage of passives to be embedded in the PCB. Further, the chapter presents results that discuss the manufacturing system behavior i.e. changes in values of key manufacturing system metrics as manufacturing cycle times and station utilizations, as the percentage of embedded passives changes. A product development team should apply the models developed to the specific system that the team plans to use. The team should consider the product and manufacturing system performance results along with other information about cost, performance, reliability, and desirability of embedding passives.

The research also developed models for manufacturing systems with process drift. As explained in Chapter 5, process drift is a condition causing a process to deviate from expected processing parameters resulting in a reduced yield at that station. Such drift situations can be detected only when errant parts reach a subsequent inspection station. The time to detect drift depends on the manufacturing cycle times of intermediate processing stations. Thus, for a manufacturing system with process drift, the manufacturing cycle time can impact yield and throughput. Models for various production scenarios incorporating process drift were developed with numerical examples showing their utility, along with suitable verification schema. This research developed algorithms to apply these mathematical models to estimate performance measures for the flow shop manufacturing scenario in Chapter 5. Applying the models to the job-shop production scenario was shown to be more complicated due to interdependencies among system attributes. Methods to identify such interdependencies and algorithms to solve a certain sub-class of the general job-shop problem were presented.

The dissertation demonstrated the importance of the DFP approach by describing the economic impact of reducing manufacturing cycle time. The impacts, modeled in Chapter 6, include lower costs, increased sales, and greater revenue. The dissertation presented detailed relationships to support these claims. Further, it presented mathematical models that help quantify these relationships. The overall economic impact of reducing manufacturing cycle time is a composition of these individual impacts.

7.2 Contributions

This section discusses the significant research contributions of the work reported in this dissertation.

Understanding the product design - manufacturing cycle time relation.

This research has presented novel approaches to estimate the manufacturing cycle time for a product before the product designs are finalized. Manufacturing cycle time has significant economic implications and contributes to total product development time. Confirming feasibility of making the product in the present manufacturing system before the designs are released for production eliminates the need for expensive re-design efforts later in the development cycle. Having reasonably accurate estimates of the manufacturing cycle time aids the product development team in developing better products. The DFP approach gives the development team the option to modify a proposed product design using this manufacturing system performance information or compare alternatives in a set of product designs. Alternatively, the tool also presents the user with the least preferred option viz. increasing the number of resources for a highly utilized workstation.

Tools developed based on the DFP approach facilitate representation and analysis of existing as well as new products being processed in the manufacturing system. The existing product set serves as a reference against which the product development team can evaluate the new product performance. The tools combine the manufacturing system models into a unique performance evaluation system. The system combines product design characteristics, process plans and manufacturing information to evaluate different performance attributes. The system includes an innovative scheme to identify the critically loaded resources in the system. The scheme estimates capacity addition requirements and generates suggestions to the effect. Further, the DFP tools dispense advice to the user regarding potential design modifications. A novel combination of resource utilizations, workstation cycle times, and product cycle times is used as an evaluation metric to present such redesign advice.

Quick analysis mechanism.

The mathematical models developed to represent the product and processing parameters, though based on certain approximating assumptions, provide a mechanism to the product design team for real-time analysis of the product design performance. Previously proposed approaches employed more simplistic manufacturing system models. This research has presented an approach based on a more realistic queuing system based model for estimating product and manufacturing system parameters. These approximations used are acceptable for preliminary design analysis especially since the time taken is much lower than simulation-based procedures. The time taken for building the corresponding simulation model is much higher than for building an analytical model. The running time for the simulation models is also greater. (Simulation can be more accurate if sufficient time is allowed and care is taken to create a valid model.)

The low analysis time requirements mean that the development team can quickly analyze multiple product design alternatives. In addition, the team can easily analyze the product and process parameters for varying product mixes, multiple order release schema and dynamic production scenarios. The models also include sensitivity analysis for the manufacturing cycle time with respect to the processing times at the various stations. This analysis can be used by the development team to better understand how their designs affect the manufacturing system performance. Though queuing network models have long been used for various analyses including representing manufacturing systems, this research effort is the first to consider the relationship between product design and a queuing network based manufacturing system and to use the queuing network parameters to suggest improvements to the product design.

Process drift representation and throughput analysis.

This research studied the phenomenon of process drift and developed new mechanisms to model and quantify the effects of process drift in a manufacturing system. Further, it created new models for the relationships between manufacturing cycle time and process drift. Because process drift affects the throughput, these novel models show that a product design affects not only manufacturing cycle time, but also throughput.

This research proposed a system graph representation for the manufacturing system. The system graph represents the relationships between the batch size for a product, the hidden yields, and the manufacturing cycle time. This novel representation is useful for systematically calculating the values in the manufacturing system model. This research has developed algorithms to use the representation to calculate the system performance measures.

Manufacturing cycle time affects product profitability.

This research studied and clearly presented how reducing manufacturing cycle time for a product results in economic benefits for the manufacturer. This compilation supports the arguments for reducing manufacturing cycle time for a product. Relating manufacturing cycle time to greater profitability is a difficult but important undertaking. No comprehensive presentation for this relation previously existed. The map model presented as a part of this research effort draws upon different costing techniques to create an edge graph representation of the cost and revenue benefits. The mathematical models presented for various relationships combine to quantify the various cost and revenue benefits. Product development teams need to understand that costs associated with the manufacturing cycle time impact the overall product cost and need to include the economic impact of manufacturing cycle time when evaluating product designs, calculating product development cost and deciding which product design alternatives to develop. New product introduction literature and product cost analysis literature do not present comprehensive models to understand the economic implications of reducing manufacturing cycle time. The economic model map presented in Section 6.1 is a novel attempt to understand the implications of manufacturing cycle time reduction on overall product profitability. Further, the mathematical model developed in Chapter 6 to quantify the impact of reducing manufacturing cycle time on inventory holding cost, implicitly models the relationship between shortages and variability in manufacturing cycle time. It also understands the association between process drift and the need to start more jobs to minimize shortages.

Embedding passives changes PCB manufacturing cycle time.

This research has developed a tool for helping a PCB development team decide whether to embed passive components into the PCB substrate and how many to embed. As described in Chapter 4, the tool analyzes the performance of the given manufacturing system as the number of embedded passives in the PCB changes. With the increasing need for smaller PCBs and the short market life of consumer electronic devices, applying DFP for aiding the decision of embedding passives into a PCB can be very useful. Previous work on studying embedded passives technology focussed on helping the PCB designer understand other criteria influencing the choice of embedding passives without a detailed analysis of the manufacturing system performance.

Clearly, product design plays a very important role in product development, significantly influencing the product life cycle, including production. It is important that the product development team understand that their design decisions affect the manufacturing system performance. Having this feedback early in the design process avoids re-work loops needed to solve problems of manufacturing capacity or cycle time. This research has classified previous DFP methods and presented a comprehensive set of models. This will help researchers and manufacturers understand the issues involved, develop better DFP tools and design more profitable products.

7.3 Future Work

Manufacturing cycle time for a workstation or for a product is inherently a function of a set of random variables. Two such variables are the processing times for a product being processed at a station and the job arrival times at the station. This work presented mathematical approximations to facilitate estimation of the manufacturing cycle time for a station and the products being processed in the manufacturing system.

The variability associated with these parameters is estimated by calculating the respective squared coefficients of variation. The manufacturing cycle time is a function of these squared coefficients of variation. Though this work has presented such approximations to model the variability in the underlying processes, it does not explicitly provide means to model the uncertainties associated with different parameters of the product design and manufacturing system. These uncertainties arise from diverse quarters. Some of these are an inability to accurately measure or forecast the values of various system parameters. It could be useful to model these uncertainties using techniques such as Monte Carlo simulation and provide ranges for the estimates of manufacturing cycle time and other performance measures.

One of the principal assumptions in the manufacturing system models is the absence of re-entrant flow in the system. Such re-entrant flow may be associated with rework and in semiconductor manufacturing, among others. Due to this assumption, whenever a defect is located in a batch, the current model discards the part as scrap. This scenario could also be handled by setting up a parallel processing sequence that processes the parts to be reworked. However, in many manufacturing systems defective parts are salvageable with appropriate rework. Such rework batches are often processed on the same processing stations as those processing the non-defective parts. It is necessary to extend the models developed here to incorporate such re-entrant flow requirements. In addition, one could extend the models to overcome other approximations (e.g. those related to setups and breakdowns).

The present mathematical treatment does not efficiently model assembly situations. Many products involve situations where a set of parts with independent processing sequences are assembled together to form a different product. With the present model, this situation can be handled by treating each of these parts as a different product and then modeling the individual processing sequences. This solution though reasonable within the limits of accuracy defined by the governing approximations in the basic manufacturing system model, nevertheless can be improved by handling assembly operations as different from other processing operations and modeling them accordingly.

Presently the approaches use a combination of resource utilizations, workstation manufacturing cycle times, and product manufacturing cycle times to present the development team with redesign suggestions. This research did not conduct an extensive study into developing criteria for generating redesign suggestions. It might be useful to develop better heuristics based on alternative performance parameter analyses or mathematical basis. These heuristics may be used by the DFP tool to present redesign suggestions to the development team. Section 2.7 surveyed some systems dispensing redesign advice based essentially on manufacturing requirements. Similarly, there certainly exists scope for work towards developing a more rigorous framework for redesign and systems providing advice based on product and manufacturing system performance.

There is also a need for more work towards looking into specific cases of applying these models when substantial product design and process changes occur. Such architectural modifications may require development of better models and new tools that take into consideration these changes and the effects of these changes. Under such conditions, the process planning also assumes greater importance. Further, redesign becomes harder due to such architectural changes which may affect product or part functionality.

This research proposed algorithms to analyze flow-shop manufacturing systems with process drift. It presented techniques for identifying parameter interdependencies for the job shop processing scenarios. It also presented algorithms to estimate system parameters for job shops where such interdependencies are absent. Though an algorithm was constructed to analyze systems with parameter interdependencies, it has not been developed and analyzed sufficiently to merit inclusion in this dissertation. Further work is needed to complete a study of the algorithm.

The analysis of manufacturing systems with process drift could be used for optimal placement of inspection stations. The decision on the number and positions of inspection stations is part of the facility design problem. A large number of inspection stations (with an upper limit of an inspection station following every processing station in the sequence) results in a lower number of scrap parts but also means a much longer manufacturing cycle time for the products being processed. On the other hand, too few inspection stations (with a lower limit of one inspection station in the entire processing sequence) or their placement too far apart would mean that the manufacturing cycle time is minimized but would cause a large number of parts to be scrapped due to large defect detection times. The literature on optimal placement of inspection stations does not address the problem of process drift and the relationship between manufacturing cycle time and drift detection time (which is also a function of inspection station positioning).

The manufacturing system models presented in this dissertation have been validated for certain types of manufacturing system. More work is necessary towards validating the models over a wide range of manufacturing systems. Additionally, the models and tools have focussed on manufacturing operations located in one facility. It may be useful to extend these models to include outsourcing and contract manufacturing. Similar models could also be developed for supply chains and facilities spread over various locations.

The economic impact of manufacturing cycle time proposed here along with the models developed pertain only to one aspect of the product development process, viz. the production process. In order to understand the impact better, it is necessary to consider cost associated with manufacturing cycle time in conjunction with other product life cycle costs. This is presented in greater detail in the later portions of this section. Also, in addition to the cost and revenue improvement proposed in Chapter 6, the price a manufacturer may expect to charge for a product is a function of object quality and development time. It may be interesting to develop product pricing models based on this hypothesis. These pricing models may ultimately be combined with the cost models to create a profitability model for the new product.

Contributing to Decision-Based Design

This research studied the impact of a product design on manufacturing system performance and presented the economic impact of reducing manufacturing cycle time for a product. Overall product economics can be used to create better and

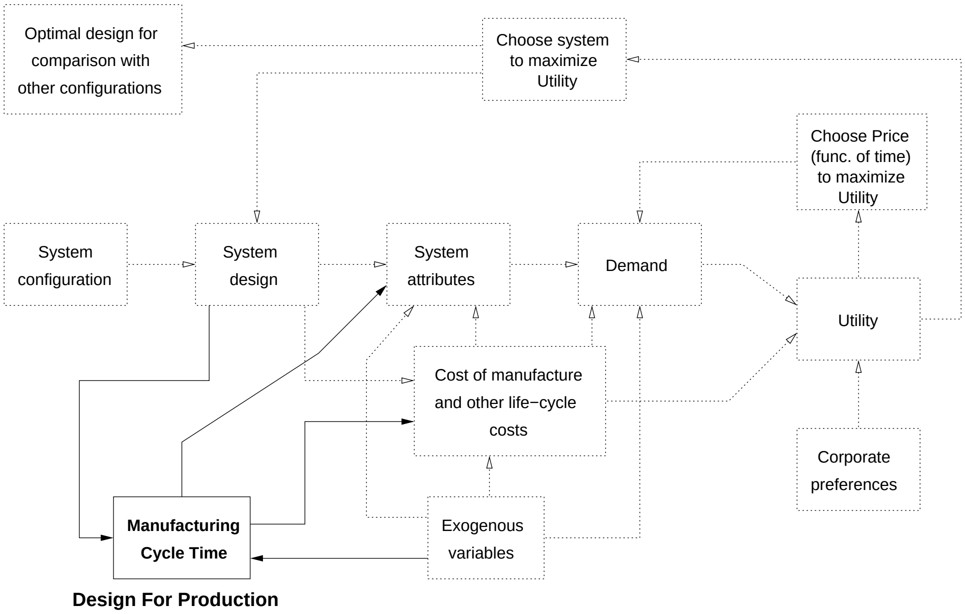
Optimal design for comparison with other configurations
Choose system to maximize Utility
Choose Price (func. of time) to maximize Utility
System configuration
System design
System attributes
Demand
Utility
Cost of manufacture and other life−cycle costs
Corporate preferences
Manufacturing Cycle Time
Exogenous variables
Design For Production

Figure 7.1: Contributions of DFP to Hazelrigg's decision-based design framework [59]

more profitable products using the ideas in decision-based design.

Hazelrigg [59] proposed a framework for decision-based design. The references to "system" in the figure will point to the product when applying the framework to product design. The purpose for the framework is to enable the assessment of a value for every design option in a design alternatives set so that options can be rationally compared and a preferred choice identified. The goal is to make a profit, considering costs and revenues associated with the product. The framework incorporates the effects of things that the designer can control (system design or product design) as well as those the designer cannot control (exogenous variables). These exogenous variables are usually random variables and can be estimated as distributions. All aspects of the design have associated costs. Demand and revenue are functions of the price. Thus the problem translates to deciding the price for the product while maximizing product utility. Since the revenues and costs are distributed over the entire product life cycle, the decision-based design framework views the design process from a systems context.

Figure 7.1 illustrates a modified version of this framework that makes explicit the role of manufacturing cycle time (indicated by the solid box and arrows in the figure) in this framework. Manufacturing cycle time is a product attribute and is affected by the product design. This was demonstrated by this research effort. Moreover, there are certain uncertainties associated with estimating the manufacturing cycle time. Some of these uncertainties were modeled here as approximations but more work needs to be done to include others. The economic impact of manufacturing cycle time and the models developed to quantify this impact can be added to the product life-cycle costs.

7.4 Concluding Remarks

The design for production methodology advocated throughout this dissertation has focussed on analyzing the relationship between a product design and a given manufacturing system using performance metrics such as manufacturing cycle time and throughput. There exist a wide variety of products and manufacturing systems and all can benefit from the design for production methodology. The requirements for the DFP approach will vary depending on the type of product being designed and the manufacturing system characteristics.

Tools based on the DFP approach must be designed based on the specific class of target products and manufacturing systems. This in turn requires understanding specific factory or supply chain performance metrics. The product development team must identify how different design decisions affect these performance metrics and to what extent. One design phase will have the largest impact on manufacturing system performance and should include the DFP methodology. This analysis will help the team develop and validate models that relate the critical design information for the associated design phase to these performance metrics. Identifying key product design characteristics would involve suitably decomposing the design into components and developing modular product architectures.

The development of DFP tools must also take into account the data available for the product and the manufacturing system, the effort involved in making that data accessible to the development team, and the time constraints that limit the amount of analysis that can be done.

It must be remembered, however, that the final goal for the product development team is to design a profitable product. Applying distinct, independent DFX methodologies targeting different aspects of the product life cycle would lead to potentially conflicting design improvement suggestions. Therefore, successfully applying the DFP approach requires coordination with other product design assessment measures, all of which finally contribute towards the ultimate aim of designing a more profitable product.

BIBLIOGRAPHY

[1] Ahmadi, R. and R.H.Wang, **"Managing Development Risk in Product Design Process,"** *Operations Research*, vol 47, no 2, pages 235-246, 1999.

[2] Aldakhilallah,K.A., and R.Ramesh, **"Computer-Integrated Process Planning and Scheduling (CIPPS): intelligent support for product design, process planning and control,"** *International Journal of Production Research*, vol 37, no 3, pages 481-500, 1999.

[3] Bacon, S.D. and D.C.Brown, **"Reasoning about mechanical devices: A top-down approach to deriving behavior from structure,"** *Computers in Engineering, Proceedings of the 1988 ASME International Computers in Engineering Conference and Exhibition*, vol 1, pages 467-472, 1988.

[4] Bakerjian, R., editor, **"Design for Manufacturability, Tool and Manufacturing Engineers Handbook,"** vol 6, Society of Manufacturing Engineers, 1992.

[5] Balachandra, R., **"Some Strategic Aspects of Faster New Product Introduction,"** *Engineering Management Conference*, pages 226-229, July, 1991.

[6] Ball, M., M.A.Fleischer and D.G.Church, **"A product design system for optimization-based tradeoff analysis,"** *2000 ASME Design Engineering Technical Conference"*, Baltimore, Maryland, Sept. 10-13, 2000.

[7] Banks, Jerry, ed., **Handbook of Simulation**, John Wiley and Sons, New York, 1998.

[8] Bardhan,T.K., V.N.Rajan and A.S.M.Masud **"Methodology and Implementation of Dynamic Design Advisor (DDA) in a Feature Based System,"** *ASME Design Engineering Technical Conference - Design Automation Conference*, Las Vegas, Nevada, Sept 1999.

[9] Bayyigit, A.C., O.Lane Inman and E.Dogus Kuran, **"New Product Introduction: Reducing Time to Market Using Internet and Intranet Technology,"** *IEEE Engineering Management Conference*, pages 454-459, .

[10] Bermon, S., G. Feigin, and S. Hood, **"Capacity analysis of complex manufacturing facilities,"** *Proceedings of the 34th Conference on Decision and Control*, pages 1935-1940, New Orleans, Louisiana, December, 1995.

[11] Bolz, Roger W. **Production Processes: Their Influence on Design**, The Penton Publishing Company, Cleveland, 1949.

[12] Boothroyd, G. **"Product Design for manufacture and assembly,"** *Computer Aided Design*, vol 26, no 7, pages 504-520, 1993.

[13] Boothroyd, G., P.Dewhurst and W.Knight **Product design for manufacture and assembly**, M. Dekker, New York, 1994.

[14] Bralla, J.G., **Design for Excellence**, McGraw-Hill Inc., USA, 1996.

[15] Bralla, J.G., Ed. **Handbook of Product Design for Manufacturing**, McGraw-Hill Book Company, New York, 1986.

[16] Brookes, N.J. and C.J.Backhouse, **"Understanding concurrent engineering implementation: a case study approach,"** *International Journal of Production Research*, vol 36, no 11, pages 3035-3054, 1998.

[17] Brookes, N.J. and C.J.Backhouse, **"Measuring the performance of product introduction,"** *Proceedings of the Institution of Mechanical Engineers, Part B*, vol 212, pages 1-11, 1998.

[18] Burhanuddin S. and S.U.Randhawa, **"A Framework for Integrating Manufacturing Process Design and Analysis,"** *Computers and Industrial Engineering*, vol 23, nos 1-4, pages 27-30, 1992.

[19] Buzacott, J.A., and J.G. Shanthikumar, **Stochastic Models of Manufacturing Systems**, Prentice-Hall, Englewood Cliffs, New Jersey, 1993.

[20] Candadai, A., J.W. Herrmann, I. Minis, **"A group technology-based variant approach for agile manufacturing,"** *Concurrent Product and Process Engineering, International Mechanical Engineering Congress and Exposition*, San Francisco, California, November 12-17, 1995.

[21] Candadai, A., J.W. Herrmann, and I. Minis, **"Applications of Group Technology in Distributed Manufacturing,"** *Journal of Intelligent Manufacturing*, vol 7, pages 271-291, 1996.

[22] Chincholkar,M.M. and J.W. Herrmann, **"Modeling the Impact of Embedding Passives on Manufacturing System Performance,"** paper DETC2002/DFM-34174 in CD-ROM Proceedings of *DETC 2002, 2002*

ASME Design Engineering Technical Conference, pp. 1-12, Montreal, Canada, Sept.29 - Oct.2, 2002.

[23] Chincholkar,M.M. and J.W. Herrmann, **"Estimating Manufacturing Cycle Time and Throughput in Flow Shops with Process Drift and Inspection,"** *Working Paper*, University of Maryland, College Park, 2002.

[24] Chun-Hsien, C. and L.G.Occena, **"A knowledge sorting process for a product design expert system,"** *Expert Systems*, vol 16, no 3, pages 170-182, Aug 1999.

[25] Cohen, M.A., J.Eliashberg and T-H.Ho, **"New Product Development: The Performance and Time-to-Market Tradeoff,"** *Management Science*, vol. 42, no. 2, pages 173-186, February 1996.

[26] Connors, D.P., G.E. Feigin, and D.D. Yao, **"A queueing network model for semiconductor manufacturing,"** *IEEE Transactions on Semiconductor Manufacturing*, vol 9, no 3, pages 412-427, 1996.

[27] Coombs, Clyde F., **Printed Circuits Handbook**, Third Edition, McGraw-Hill Book Company, New York, 1988.

[28] Cormen, T.H., C.L.Leiserson and R.L.Rivest, **Introduction to Algorithms**, The MIT Press, Cambridge, MA, USA, 1997.

[29] Crawford, R.H. and D.C.Anderson, **"A Computer Representation for Modeling Feedback in Design Processes,"** *Computers in Engineering, Proceedings of the ASME International Computers in Engineering Conference and Exhibition*, vol 1, pages 165-170, 1990.

[30] Cunningham, S.P. and J.G.Shanthikumar, **"Empirical results on the relationship between die yield and cycle time in semiconductor wafer fabrication,"** *IEEE Transactions on Semiconductor Manufacturing*, vol 9, no. 2, pages 273-277, May 1996.

[31] Cutkosky, M.R. and J.M.Tenenbaum, **"Providing Computational Support for Concurrent Engineering,"** *International Journal of Systems Automation: Research and Automation (SARA) 1*, pages 239-261, 1991.

[32] Darken, Rudolph and Christian J. Darken, **"VR + AI = intelligent environments: A synergistic approach to engineering design support,"** *SPIE*, vol 2653, pages 292-300, 1996.

[33] Datar, S., C.Jordan, S.Kekre, S.Rajiv and K.Srinivasan, **"New Product Development Structures and Time-to-Market,"** *Management Science*, vol. 43, no. 4, pages 452-464, April 1997.

[34] De Martino, Teresa, **"A Multiple-View CAD Representation for Product Modeling,"** *Proceedings of Shape Modeling International, '97*, International Conference on Shape Modeling and Applications, pages 78-85, Aizu-Wakamatsu, Japan, 1997.

[35] De Martino, T., B.Falcidieno, and S.Haβinger, **"Design and engineering process integration through a multiple view intermediate modeler in a distributed object-oriented system environment,"** *Computer-Aided Design*, vol 30, no 6, pages 437-452, 1998.

[36] De Martino, T. and F.Giannini, **"Modeling Product Information in Concurrent Engineering Environments,"** in P.Chedmail, J.-C.Bocquet and

D.Dornfield eds., *Integrated Design and Manufacturing in Mechanical Engineering, Proceedings of the* 1^{st} *IDMME Conference*, pages 437-444, Nantes, France, 15-17 April, 1996.

[37] De Martino, T. and F.Giannini, **"Feature-based Product Modeling in Concurrent Engineering,"** in G.Jacicci, G.J.Olling, K.Preiss and M.Wozny, eds., *Globalization of Manufacturing in the Digital Communications Era of the* 21^{st} *Century : Innovation, Agility, and the Virtual Enterprise*, Proceedings of the Tenth International IFIP WG 5.2/5.3 International Conference, pages 351-357, Trento, Italy, Sept 9-12, 1998.

[38] Deng, Y-M., S.B.Tor and G.A.Britton, **"A Computerized Design Environment for Functional Modeling of Mechanical Parts,"** *ACM Fifth Symposium on Solid Modeling*, pages 1-12, Ann Arbor, MI, 1999.

[39] Dixon, J.R., M.R.Duffey, R.K.Irani, K.L.Meunier and M.F.Orelup **"A Proposed Taxonomy of Mechanical Engineering Design Problems,"** *Computers in Engineering 1988, Proceedings of the 1988 ASME International Computers in Engineering Conference and Exhibition*, vol 1, pages 41-46, 1988.

[40] Doan, N.W., M.I.G.Bloor and M.J.Wilson, **"A Strategy for the Automated Design of Mechanical Parts,"** $2^{n}d$ *ACM Solid Modeling Conference*, pages 15-21, Montreal, Canada, 1993.

[41] Dong, Zuomin, **"Design for Automated Manufacturing,"** in Andrew Kusiak, ed. *Concurrent Engineering: Automation, Tools and Techniques*, pages 207-233, John Wiley and Sons, USA, 1993.

[42] Dossenbach, Tom, **"Manufacturing Cycle Time Reduction - A Must in Capital Project Analysis,"** *Wood and Wood Products*, vol 105, issue 11, pages 31-35, October 2000.

[43] Elhafsi, M., and E.Rolland, **"Negotiating price and delivery date in a stochastic manufacturing environment,"** *IIE Transactions*, vol 31, pages 255-270, 1999.

[44] Fillebrown, W.S and B.Mahler, **"Board-Embedded Planar Resistors for High-Density Circuits,"** *U.S.TECH*, January 1997.

[45] Flasinski, Mariusz, **"Use of graph grammars for the description of mechanical parts,"** *Computer-Aided Design*, vol 27, no. 6, pages 403-433, 1995.

[46] Garg, Amit, **"An application of designing products and processes for supply-chain management,"** *IIE Transactions*, vol. 31, pages 417-429, 1999.

[47] Gatenby, D.A. and G.Foo, **"Design for X (DFX): Key to Competitive, Profitable Products,"** *AT&T Technical Journal*, pages 2-13, May/June 1990.

[48] Giannini, Franca, **"Geometric Reasoning as a tool for Integrating Design and Production Activities in a Mechanical Environment,"** *International Journal of Shape Modeling*, vol 2, no. 2&3, pages 117-137, 1996.

[49] Gong, Dah-Chuan and Leon McGinnis, **"Towards a manufacturing metamodel,"** *International Journal of Computer Integrated Manufacturing*, vol. 9, no. 1, pages 32-47, 1996.

[50] Govil, Manish, **Integrating product design and production : Designing for time-to-market**, PhD Thesis, University of Maryland, College Park, USA, 1999.

[51] Govil, M., and E. Magrab, **"Incorporating production concerns in conceptual product design,"** *International Journal of Production Research*, vol. 38, no. 16, pages 3823-3843, 2000.

[52] Gupta, S.K., and D. S. Nau, **"A systematic approach for analyzing the manufacturability of machined parts,"** *Computer Aided Design,* 27(5), pages 323-342, 1995.

[53] Gupta, S.K., D. S. Nau, W. C. Regli, and G. Zhang, **"A methodology for systematic generation and evaluation of alternative operation plans,"** in Jami Shah, Martti Mantyla, and Dana Nau, editors, *Advances in Feature Based Manufacturing*, pages 161–184. Elsevier/North Holland, 1994.

[54] Hauptman, O. and K.K.Hirji, **"The Influence of Process Concurrency on Project Outcomes in Product Development: An Empirical Study of Cross-Functional Teams,"** *IEEE Transactions on Engineering Management*, vol 43, no 2, pages 153-164, May 1996.

[55] Hayes, C.C. **"Plan-based manufacturability analysis and generation of shape-changing redesign suggestions,"** *Journal of Intelligent Manufacturing*, vol 7, pages 121-132, London, 1996.

[56] Hayes, C.C., S. Desa, and P. K. Wright, **"Using process planning knowledge to make design suggestions concurrently,"** in N. H. Chao and S.

C. Y. Lu, editors, *Concurrent Product and Process Design*, ASME Winter Annual Meeting, pages 87-92. ASME, 1989.

[57] Hayes, C.C. and D.Gaines, **"Using Near-Misses from Feature Recognition to Generate Redesign Suggestions for Increased manufacturability,"** *Design for Manufacturing and Assembly - ASME*, DE-Vol. 89, pages 67-77, Atlanta, Georgia, 1996.

[58] Hayes, C.C., and H. C. Sun, **"Plan-based generation of design suggestions,"** in R. Gadh, editor, *Concurrent Product Design*, ASME Winter Annual Meeting, pages 59–69, Design Engineering Division, ASME, New York, 1994.

[59] Hazelriggg, G.A., **"A Framework for Decision-Based Engineering Design,"** *Journal of Mechanical Design*, vol 120, pages 653-658, December 1998.

[60] Henderson, Mark, **"Representing Functionality and Design Intent in Product Models,"** 2^{nd} *ACM Solid Modeling*, pages 387-396, Montreal, Canada, May 1993.

[61] Henderson, Mark, and LeRoy Taylor, **"A Meta-Model for Mechanical Products based upon the Mechanical Design Process,"** *Research in Engineering*, vol 5, pages 140-160, London, 1993.

[62] Herrmann, J.W., and M. Chincholkar, **"Design for production: a tool for reducing manufacturing cycle time,"** paper DETC2000/DFM-14002 in CD-ROM Proceedings of *DETC 2000, 2000 ASME Design Engineering Technical Conference*, pp. 1-10, Baltimore, September 10-13, 2000.

[63] Herrmann, J.W., G. Lam, and I. Minis, **"Manufacturability analysis using high-level process planning,"** 1996 ASME Design for Manufacturing Conference, University of California, Irvine, August 18-22, 1996.

[64] Hewitt, Sara, **Comparing Analytical and Discrete-Event Simulation Models of Manufacturing Systems**, MS Thesis, University of Maryland, College Park, USA, 2002.

[65] Hopp, Wallace J., and Mark L. Spearman, **Factory Physics**, Irwin/McGraw-Hill, Boston, 1996.

[66] Hsu, Wynne, C. S. George Lee, and S. F. Su, **"Feedback approach to design for assembly by evaluation of assembly plan,"** *Computer Aided Design*, 25(7):395–410, July 1993.

[67] Hu, W. and C.Poli, **"To Injection Mold, to Stamp, or to Assemble? A DFM Cost Perspective,"** *Journal of Mechanical Design*, vol 121, pages 461-471, Dec 1999.

[68] Huang, G.Q and K.L.Mak, **"The DFX shell: A generic framework for applying 'Design for X' (DFX) tools,"** *International Journal of Computer Integrated Manufacturing*, vol. 11, no. 6, pages 475-484, 1998.

[69] Huang, G.Q and K.L.Mak, **"Redesigning the product development process with design for 'X' ,"** *Proceedings Institution Mechanical Engineers*, vol 212 Part B, pages 259-268, 1998.

[70] Huang, G.Q and K.L.Mak, **"Design for manufacture and assembly on the Internet,"** *Computers in Industry*, vol 38, pages 17-30, 1999.

[71] Ishii, Kosuke, **"Modeling of concurrent engineering design,"** in Andrew Kusiak, editor, *Concurrent Engineering: Automation, Tools and Techniques*, pages 19-39. John Wiley & Sons, Inc., 1993.

[72] Izuchukwu, John, I., **"Intelligent Foundation for Product Design Reduces Costs, Time-To-Market: Part-1,"** *Industrial Engineering*, pages 29-34, July, 1991.

[73] Jakiela, M.J., and Papalambros, P.Y., **"Design and Implementation of a Prototype 'Intelligent' CAD System,"** *Transactions of the ASME - Journal of Mechanisms, Transmission, and Automation in Design*, vol 111, pages 252-258, 1989.

[74] Kalpakjian, S., **Manufacturing Processes for Engineering Materials**, Addison Wesley, Reading, Massachusetts, 1984.

[75] Kanai,S., R.Sasaki, and T.Kishinami, **"Graph-Based Information Modeling of Product-Process Interactions for Disassembly and Recycle Planning,"** *IEEE*, pages 772-777, 1999.

[76] Kimura, F. and H. Suzuki, **"A CAD System for Efficient Product Design Based on Design Intent,"** *Annals of the CIRP*, vol.38, no.1, pages 149-152, 1989.

[77] Koo, P.-H., C.L. Moodie, and J.J. Talavage, **"A spreadsheet model approach for integrating static capacity planning and stochastic queuing models,"** *International Journal of Production Research*, vol 33, no. 5, pages 1369-1385, 1995.

[78] Krishnan, S and Edward Magrab **"Performing DFM Analysis in Multiple Domains using Interchangeable Manufacturable Entities,"** paper DETC1997/CIE-4282 in CD-ROM Proceedings of *2002 ASME Design Engineering Technical Conference*, Sacramento, California, Sept 1997.

[79] Kusiak, A., and Weihua He, **"Design of components for schedulability,"** *European Journal of Operational Research*, vol 76, pages 49-59, 1994.

[80] Law, Averill M., and W. David Kelton, **Simulation Modeling and Analysis**, 2nd edition, McGraw-Hill, New York, 1991.

[81] Lee, Hau L., C.Billington, and B.Carter, **"Hewlett-Packard gains control of inventory and service through design for localization,"** *Interfaces*, vol 23, no 4, pages 1-11, 1993.

[82] Lee, Tae-Eog, and Marc Posner, **"Performance measures and schedules in periodic job shops,"** *Operations Research*, vol 45, no 1, pages 72-91, 1997.

[83] Lenau, T. and L.Alting, **"Intelligent Support Systems for Product Design,"** *Annals of the CIRP*, vol 38, no 1, pages 153-156, 1989.

[84] Li, Chung-Lun, and T.C.E. Cheng, **"Due-date determination with resequencing,"** *IIE Transactions*, vol 31, pages 183-188, 1999.

[85] Lu, S.C.H., D.Ramaswamy and P.R.Kumar, **"Efficient scheduling policies to reduce mean and variance of cycle-time in semiconductor manufacturing plants,"** *IEEE Transactions on Semiconductor Manufacturing*, vol. 7, no. 3, pages 374-388, August 1994.

[86] Magrab, Edward B., **Integrated Product and Process Design and Development : The Product Realization Process**, CRC Press, Boca Raton, Florida, 1997.

[87] Marcanti, L., *et al.*, **NEMI Technology Roadmaps**, December 2000.

[88] McGrath, Michael E., **Setting the PACE in Product Development - A Guide to Product And Cycle-time Excellence**, Butterworth-Heinemann, Boston, USA, 1997.

[89] McKay, A., F.Erens and M.S.Bloor, **"Relating Product Definition and Product Variety,"** *Research in Engineering Design*, vol 2, pages 63-80, 1996.

[90] Meyer, J. *et al.*, **"Process Planning in Microwave Module Production,"** in *1998 Artificial Intelligence and Manufacturing: State of the Art and State of Practice*, September, 1998.

[91] Minis, I., J.W. Herrmann, G. Lam, and E. Lin, **"A Generative Approach for Concurrent Manufacturability Evaluation and Subcontractor Selection,"** *Journal of Manufacturing Systems*, vol 18, no 6, pages 383-395, 1999.

[92] Minis, I., J.W. Herrmann, and G. Lam, **"A Generative Approach for Design Evaluation and Partner Selection for Agile Manufacturing,"** Technical Report 96-81, Institute for Systems Research, University of Maryland, College Park, 1996.

[93] Murayama Takeshi, Kentaro Kagawa and Fuminori Oba **"Computer Aided Redesign for Improving Recyclability,"** *IEEE*, pages 746-751, Feb 1999.

[94] Nahmios, Steven, **Production and Operations Analysis**, Richard dIrwin, Boston, 1993.

[95] Narhari, Y., and L.M. Khan, **“Modeling Reentrant Manufacturing Systems with Inspection Stations,”** *Journal of Manufacturing Systems*,vol. 15, no. 6, pages 367-378, 1996.

[96] Nemoto, K., E.Akcali and R.M.Uzsoy, **“Quantifying the Benefits of Cycle Time Reduction in Semiconductor Wafer Fabrication,”** *IEEE Transactions on Electronics Packaging Manufacturing*, vol. 23, no. 1, pages 39-47, Jan 2000.

[97] Nielsen, N., P.Heeser and J.Holmstrom, **“Design for speed: a supply chain perspective on design for manufacturability,”** *Computer Integrated Manufacturing Systems*, vol 8, no 3, pages 223-228, 1995.

[98] Nieminen, J. and J. Tuomi, **“Design with features for manufacturing cost analysis,”** in J.Turner *et al.* eds. *Product Modeling for Computer Aided Design and Manufacturing*, pages 317-330, 1991.

[99] O’Grady Peter *et al.*, **“An advice system for concurrent engineering,”** *International Journal of Computer Integrated Manufacturing*, vol. 4, no. 2, pages 63-70, 1991.

[100] Technical Bulletin F-1A, **Processing Recommendations - Ohmega-Ply© Resistor-Conductor Material**, *Ohmega Technologies, Inc.*, Culver City, CA, USA, June 1999.

[101] Mahler, Bruce P.,**Integral Resistors in High Frequency Printed Wiring Boards**, *TECHNICAL FEATURE, Ohmega Technologies, Inc.*, Culver City, CA, USA.

[102] Pahl, G. and W. Beitz. **Engineering Design**, Design Council, London, 1984.

[103] Papadopoulos, H.T., C. Heavey, and J. Browne, **Queuing Theory in Manufacturing Systems Analysis and Design**, Chapman and Hall, London, 1993.

[104] Pnueli, Y. and E.Zussman, **"Evaluating the end-of-life value of a product and improving it by redesign,"** *International Journal of Production Research*, vol 35, no. 4, pages 921-942, 1997.

[105] Power, C., M.Realff, and S.Bhattacharya, **"A Decision Tool for Design of Manufacturing Systems for Integrated Passive Substrates,"** *Proceedings of the IMAPS* 4th *Advanced Technology Workshop on Integrated Passives Technology*, Denver, CO, April, 1999.

[106] Rajagopalan, S., **"Make to Order or Make to Stock: Model and Application,"** *Management Science*, vol 48, no 2, pages 241-256, February 2002.

[107] Realff, M., C.Power, T.Bankhead, B.Oates and D.Baldwin, **"Technical Cost Modeling for Decisions in Integrated vs Surface Mounted Passives,"**, *Proceedings of the IMAPS 3rd Advanced Technology Workshop on Integrated Passives Technology*, Denver, April 1998.

[108] Rector, J., **"Economic and Technical Viability of Integral Passives,"** *Proceedings: 4th International Symposium on Advanced Packaging Materials: processes, properties and interfaces*, Braselton, GA, USA, 15-18 March, 1998.

[109] Rinderle, J.R., **"Implications of Function-Form-Fabrication Relations in Design Decomposition Strategies,"** *Computers in Engineering 1986, Proceedings of the 1986 ASME International Computers in Engineering Conference and Exhibition*, vol 1, pages 193-198, 1986.

[110] Rosen, David W., John R. Dixon, Corrado Poli, and Xin Dong. **"Features and algorithms for tooling cost evaluation in injection molding and die casting,"** *Proceedings of the ASME International Computers in Engineering Conference*, pages 1-8. ASME, 1992.

[111] Sandborn, P.A, B.Etienne, and G.Subramanium, **"Application-Specific Economic Analysis of Integral Passives,"** *IEEE Trans. on Electronics Packaging Manufacturing*, vol 24, No. 3, pages 203-213, July 2001.

[112] Sandborn, Peter, *"Electronic Product and System Cost Analysis,"* Lecture Notes, University of Maryland, College Park, 2000.

[113] Sandborn, Dr.Peter A., Personal Discussions, University of Maryland, College Park, Feb.2001 - Dec.2001.

[114] Scheffler, M., G.Tröster, J.L.Contreras, J.Hartung, and M.Menard, **"Assessing the cost-effectiveness of integrated passives,"** *Microelectronics Intenational*, vol 17, no 3, pages 11-15, 2000.

[115] Schilling, M.A., and Charles W.L. Hill, **"Managing the new product development process: strategic imperatives,"** *IEEE Engineering Management Review*, pages 55-68, Winter, 1998.

[116] Schmidt, L. and J. Cagan, **"Grammars for Machine Design,"** *Artificial Intelligence in Design '96, ed. J.S.Gero and F. Sudweeks*, pages 325-344, Kluwer Academic Publishers, Netherlands, 1996.

[117] Signer, R.M., **"Incorporating Planar Resistors in PCB Designs,"** *Electronic Manufacturing*, January 1989.

[118] Seepersad, C.C., G.Hernandez and J.K.Allen, **"A quantitative approach to determining product platform extent,"** paper DETC2000/DAC-14288 in CD-ROM Proceedings of *DETC 2000, 2000 ASME Design Engineering Technical Conference*, pages 1-10, Baltimore, September 10-13, 2000.

[119] Singh, Nanua, **Systems Approach to Computer-Integrated Design and Manufacturing**, John Wiley and Sons, New York, 1996.

[120] Smith, C.S., and P.K.Wright, **"CyberCut: A World-Wide Web Based Design-to-Fabrication Tool,"** *Journal of Manufacturing Systems*, vol 15, no 6, 1996.

[121] Smith, P.G., and D.G. Reinertsen, **"Developing products in half the time: new rules, new tools,"** *Van Nostrand Reinhold (International Thomson Publishing Company)*, New York, 1998.

[122] Soundar, P., and Han P. Bao, **"Concurrent design of products for manufacturing system performance,"** *Proceedings of the IEEE 1994 Interna-*

tional Engineering Management Conference, pages 233-240, Dayton, Ohio, October 17-19, 1994.

[123] Srinivasan, K., R. Sandell, and S. Brown, **"Correlation Between Yield and Waiting Time: A Quantitative Study,"** *Transactions of 1995 IEEE/CPMT International Electronics Manufacturing Technology Symposium*, 1995.

[124] Suh, N.P., **Principles of Design**, Oxford University Press, New York, 1990.

[125] Suri, Rajan, **"Lead time reduction through rapid modeling,"** *Manufacturing Systems*, pages 66-68, July, 1989.

[126] **"1999 Simulation Software Survey,"** *OR/MS Today*, vol 26, no 1, pages 42-51, 1999.

[127] Syan, C.S, **"Introduction to concurrent engineering,"** in C.S.Syan and U. Menon, eds. *Concurrent Engineering - Concepts, implementation and practice*, pages 3-24, Chapman and Hall, London, UK, 1994.

[128] Syan, C.S and K.G.Swift, **"Design for manufacture,"** in C.S.Syan and U. Menon, eds. *Concurrent Engineering - Concepts, implementation and practice*, pages 101-116, Chapman and Hall, London, UK, 1994.

[129] Syan, C.S and K.G.Swift, **"Design for assembly,"** in C.S.Syan and U. Menon, eds. *Concurrent Engineering - Concepts, implementation and practice*, pages 117-136, Chapman and Hall, London, UK, 1994.

[130] Tan, Gek Woo, Caroline C. Hayes and Michael Shaw, **"An Intelligent-Agent Framework for Concurrent Product Design and Planning,"**

IEEE Transactions on Engineering Management, vol 43, no 3, pages 297-306, August 1996.

[131] Tan, Gek Woo, Caroline C. Hayes and Michael Shaw, **"Concurrent product design,"** *IEEE Potentials*, pages 9-12, April/May 1997.

[132] Taylor, D.G., J.R.English, and R.J.Graves, **"Designing new products: Compatibility with existing product facilities and anticipated product mix,"** *Integrated Manufacturing Systems*, vol 5, no 4/5, pages 13-21, 1994.

[133] Taylor, LeRoy, **Meta-Physical Product Modeling**, PhD Thesis, Arizona State University, Arizona, USA, 1993.

[134] Trichur, Vinai S., **Integer programming models for product design**, PhD Thesis, College of Business and Management, University of Maryland, College Park, 1999.

[135] Trucks, H.E., **Designing for Economic Production**, Society of Manufacturing Engineers, 1987.

[136] van Vliet, J.W., C.A.vanLuttervelt and H.J.J.Kals **"State-of-the-art report on Design for Manufacturing,"** paper DETC1999/DFM-8970 in CD-ROM Proceedings of *1999 ASME Design Engineering Technical Conference*, Las Vegas, Nevada, Sept 1999.

[137] Veeramani, D., and P.Joshi, **"Methodologies for rapid and effective response to requests for quotation,"** *IIE Transactions*, vol 29, pages 825-838, 1997.

[138] Veeramani, D., and T.Mehendale, **"Online design and price quotations for complex product assemblies: the case of overhead cranes,"** *1999 ASME Design for Manufacturing Conference*, Las Vegas, Nevada, September 12-15, 1999.

[139] Vollmann, T.E., W.L. Berry, and D. Clay Whybark, **Manufacturing Planning and Control Systems**, 4th edition, Irwin/McGraw-Hill, New York, 1997.

[140] Wei, Yu-Feng and A.C.Thornton **"Concurrent Design for Optimal Production Performance,"** paper DETC2002/DFM-34163 in CD-ROM Proceedings of *2002 ASME Design Engineering Technical Conference*, Montreal, Canada Sept. 29 - Oct. 2, 2002.

[141] Wei, Yu-Feng, **Concurrent Design for Optimal Quality and Cycle Time**, PhD Thesis, Department of Mechanical Engineering, Massachusetts Institute of Technology, Cambridge, Massachusetts, February, 2001.

[142] Wei, Yui and P.J.Egbelu, **"A framework for estimating manufacturing cost from geometric design data,"** *International Journal of Computer Integrated Manufacturing*, vol. 13, no. 1, pages 50-63, 2000.

[143] Weng, Z.K., **"Strategies for integrating lead time and customer-order decisions,"** *IIE Transactions*, vol. 31, pages 161-171, 1999.

[144] Wu, Tong and Peter O'Grady, **"A Concurrent Engineering Approach to Design for Assembly,"** *CONCURRENT ENGINEERING: Research and Applications*, vol. 7, no. 3, pages 231-243, September 1999.

[145] Yoon, E. and V.Kijewski, **"Dynamics of the relationship between product features, quality evaluation, and pricing,"** *Pricing Strategy and Practice*, vol 5, no. 2, pages 45-60, 1997.

[146] Zhou, K., *et al.*, **"A Problem Solving Architecture for Virtual Prototyping in Metal to Polymer Composite Redesign,"** *ASME Design Engineering Technical Conference*, Las Vegas, September 12-15, 1999.

[147] http://www.circuitree.com/CDA/ArticleInformation/coverstory/
BNPCoverStoryItem/0,2135,21373,00.html.