DESIGN OF A CONTROL ARCHITECTURE
FOR RE-ENTRANT FLOW SHOPS

By

JAGANNATH T. GHARPURE

Bachelor of Engineering
The M.S. University of Baroda
Baroda, India
1978

Master of Engineering
National Institute for Industrial Engineering
Bombay, India
1981

Submitted to the Faculty of the
Graduate College of the
Oklahoma State University
in partial fulfillment of
the requirements for
the Degree of
DOCTOR OF PHILOSOPHY
July, 1994

DESIGN OF A CONTROL ARCHITECTURE FOR RE-ENTRANT FLOW SHOPS

Thesis Approved:

Joe H. Mize
Thesis Adviser

C. M. Bauer

P. L. Claypool

Manjunath Kamath

M. P. Terrell

Thomas C. Collins
Dean of the Graduate College

ACKNOWLEDGMENTS

"Let him who is taught the word share in all good things with him who teaches". Keeping in the true spirit of these holy words from the Bible, Galatians, Chapter 6, verse 6, I dedicate this work to the following teachers, who have played a great role in molding my life.

First and foremost I dedicate this work to God, the Almighty, who is the greatest teacher of all, who in the divine form of Lord Krishna taught Arjuna, the meaning of life, on the great battlefields of Kurukshetra. I pay my respect and lay this work at HIS feet in gratitude for all the good things that he has bestowed upon me. Next I dedicate this work to Dr. Joe H. Mize, my guru. I am very grateful to you for being not only my teacher but also a father-figure that was never wanting in providing me emotional support that I needed on several occasions. Without your support I could not have sailed through the most trying situations that I faced during the advent of this work. Lastly, I dedicate this work to my parents, Mrs. Leela and Mr. Trimbak Gharpure. I thank you both for the values that you have instilled in me.

Special thanks go to all the members on my doctoral committee for the way they have contributed in building and impacting my thought processes towards my research. Dr. Kamath has provided numerous suggestions during the proposal stages of this research. Dr. Terrell introduced me to the domain of multi-criteria decision making. Dr. Bacon laid foundations of Artificial Intelligence in my novice mind when I took a course in AI with him. Dr. Claypool has helped me not only by teaching me the art and

science of statistics and design of experimentation in the course STAT 5023, but also during the final phase of my dissertation when I was struggling to carry out the statistical analysis of the vast data.

Among friends and colleagues, I would like to thank my best friends Manoj Duse and Avinash Chatorikar. Thank you very much for all the support and the love that you two have given to me.

I would like to thank my son Ameya for the pure, wonderful and innocent love that he has shown towards me. Son, you may not know at this point in your life as to what a tremendous role you have played in keeping the zest for life alive in me.

Finally, I gratefully acknowledge the support provided by the Oklahoma Center for the Advancement of Science and Technology (OCAST) and the AT&T Foundation.

TABLE OF CONTENTS

Chapter Page

1. INTRODUCTION

1.1 Motivation for the Research 1
1.2 Focus of the Research 3
1.3 Overview of the Dissertation 4

2. STATEMENT OF THE PROBLEM

2.1 Introduction 5
2.2 Description of a General Re-entrant Flow Shop 5
2.3 Description of Semiconductor Wafer Fabrication RFSs 7
2.4 Complexities in Production Control of RFSs........ 8
2.5 Statement of the Problem 9
2.6 Proposed Approach 15
2.7 Proposed Methodology 16

3. BACKGROUND OF THE STUDY

3.1 Introduction 19
3.2 Review of Literature Related to Control of CIM Systems 19
3.3 Review of Literature Related to Re-Entrant Flow Shops 31
3.4 Multiple Criteria Decision Making (MCDM) Approach to Control 36

4. STATEMENT OF RESEARCH

4.1 Research Goal 38
4.2 Research Objectives 38
4.3 Research Questions 40
4.4 Research Assumptions 43

5. RESEARCH PLAN AND PROCEDURES

5.1 Research Phases 45

Chapter Page

6. DESIGN OF A CONTROL ARCHITECTURE FOR RE-ENTRANT FLOW SHOPS

6.1 Introduction 48
6.2 The Control Architecture 48
6.3 Object Oriented Implementation Details 62

7. EXPERIMENTAL DETAILS

7.1 Introduction 71
7.2 Classification of Complexities for the RFS 71
7.3 Measures of Merit 73
7.4 Experimentation for Study in the Context of Structural Complexities 74
7.5 Experimentation in the Context of Process Time Variability (Factor SV6) 84
7.6 Experimentation in the Context of Availability Variability (Factor SV5) 94
7.7 Effect of the Optimization Objective Used in the LP 95

8. COMPARISONS, POSSIBLE EXTENSIONS AND FUTURE RESEARCH

8.1 Introduction 99
8.2 Use of Multiple Objectives for Control 99
8.3 How the Architecture Performed 102
8.4 Possible Extensions 104
8.4 Place of the Architecture in the Big Picture 106
8.5 Directions for Future Research 109

9. SUMMARY AND CONCLUSIONS

9.1 Introduction 110
9.2 Research Summary 110
9.3 Research Contributions 112

BIBLIOGRAPHY 114

Chapter Page

APPENDIXES

APPENDIX 1 - RFS AS A DYNAMICAL SYSTEM 122
APPENDIX 2 - LINEAR PROGRAMMING FORMULATION 125

LIST OF TABLES

Table Page

1. Types of Complexities 71
2. Weekly Production Targets 74
3. Factor Combinations for Study in the Context of Structural Complexities 75
4. Routing Details (SV2-L1). 75
5. Results (SV2-L1) 76
6. Routing Details (SV2-L2) 77
7. Results (SV2-L2) 77
8. Routing Details (SV2-L3) 79
9. Results (SV2-L3) 79
10. Routing Details (SV3-L2) 80
11. Results (SV3-L2) 81
12. Routing Details (SV3-L3) 81
13. Results (SV3-L3) 81
14. Routing Details (SV4-L2) 82
15. Results (SV4-L2) 82
16. Routing Details (SV4-L3) 82
17. Results (SV4-L3) 83
18. Weekly Production Targets 84
19. Routing Details for Studying Effects of Randomness 84

Table Page

20. Treatment Combinations for Study in the Context of Process Time Variability 85

21. ANOVA for Flow Time Mean (FLOWM) 86

22. ANOVA for Flow Time Mean with COMSTGY = One-Way 87

23. ANOVA for Flow Time Mean with COMSTGY = Two-Way 87

24. Duncan's Test for Flow Time Mean with Independent Variable = OPTVAR 87

25. Duncan's Test for Flow Time Mean with Independent Variable = TPOFPA 88

26. ANOVA for Flow Time Sample Standard Deviation (FLOWSD) 89

27. ANOVA for FLOWSD with COMSTGY = One-Way 90

28. ANOVA for FLOWSD with COMSTGY = Two-Way 90

29. Duncan's Test for FLOWSD with Independent Variable = OPTVAR 90

30. Duncan's Test for FLOWSD with Independent Variable = TPOFPA 90

31. ANOVA for Percentage Production Not Made (PPNM) 91

32. ANOVA for PPNM with COMSTGY = One Way 91

33. ANOVA for PPNM with COMSTGY = Two Way 91

34. Duncan's Test for PPNM with Independent Variable = OPTVAR 92

35. Duncan's Test for PPNM with Independent Variable = TPOFPA 92

36. ANOVA for Number In System Mean (NISM) 92

37. ANOVA for NISM with COMSTGY = One-Way 93

38. ANOVA for NISM with COMSTGY = Two-Way 93

39. Duncan's Test for NISM with Independent Variable = OPTVAR 93

Table Page

40. Duncan's Test for NISM with Independent Variable = TPOFPA 93
41. Treatment Combinations - Variability in Machine Availability 94
42. Routing Details - Effect of Type of Objective Function 96
43. Comparison of Queue Lengths and Time In Queue 97
44. Work Centerwise Queue Lengths 97
45. Productwise Time In System and Queuing Time 97
46. Time In System and Number In System - Averaged Over Products 98
47. Routing for Product 1 100
48. Routing for Product 2 100
49. Effect of Multiple Objectives on RFS Performance 101
50. Details of the RFS 102
51. Comparison of Terminology 102
52. Product Routings 103

LIST OF FIGURES

Figure Page

1. Meaning of Various Time Intervals 6
2. Flow Chart of Proposed Methodology 17
3. The Control Architecture 49
4. Flow of Information from RFS to PCBrain 57
5. Complexity Due to Number of Places where Re-entrancy Occurs (SV2) 72
6. Complexity Due to Span of Re-entrancy (SV3) 73
7. Complexity Due to Number of Re-Entrant Paths (SV4) 73
8. RFS Performance (SV2-L1) 77
9. RFS Performance (SV2-L2) 78
10. RFS Performance (SV2-L3) 80
11. RFS Structure for Studying Effects of Randomness 84
12. RFS Structure for Studying Effect of Type of Objective Function Used in LP 95
13. Place of the Architecture in The Big Picture 107

NOMENCLATURE

AI	Artificial Intelligence.
ANOVA	Analysis Of Variance.
CA	Control Architecture.
CIM	Computer Integrated Manufacturing.
COMSTGY	Communication Strategy. Can be TW (Two-Way) or OW (One-Way).
DF	Degree of Freedom.
DG	Duncan Grouping.
d_i	Demand for part type i (NOT rate).
EDD	Earliest Due Date. It is a dispatching rule as per which the next part to be selected for processing is the one for which the due date for delivery is the earliest among all waiting to be processed.
EDDT	Earliest Due Date Truncated. Similar to EDD except that from time to time EDD is not followed but some truncation rule is used to choose from among those parts which have waited for "too long".
F	F statistics value.
FBFS	First Buffer First Served. This dispatching rule is the exact opposite of LBFS. Considering the example used for LBFS, the next part chosen for processing will be the one waiting for operation 2.
FCFS	First Come First Served.
FLOWM	Mean of Flow Time.
FLOWSD	Sample Standard Deviation of Flow Time.
FMS	Flexible Manufacturing System.

FT	Flow Time. The time elapsed from the instant the part is released into the RFS upto the instant it is completed. This is used as one of the measures of merit.
JIT	Just In Time. A Japanese philosophy of conducting a business.
KB	Knowledge Base.
LBFS	Last Buffer First Served. If a part visits a work center for say operation 2, operation 4 and operation 8, and if at a given instant there is one part each waiting for each of these operations then the part with the last visit (to this work center) is selected for processing, i.e. the part waiting for operation 8 gets priority over all the others.
LINDO	Linear Interactive Discrete Optimizer. A commercial software for solving Linear, Integer and Quadratic Programming problems.
LP	Linear Program or Linear Programming.
LPT	Longest Processing Time. A dispatching rule in which the next part to be selected for processing is the one with the longest processing time among all waiting to be processed.
LS	Least Slack. A dispatching rule in which priority of parts waiting to be processed is based on "slack" of a part, defined as the due date minus an estimate of the remaining delay.
LWait	Longest Waiting. This dispatching rule selects that part for processing which has waited longest in the queue till now.
$l_i(k)$	Loading rate for part type i in k^{th} small time interval (decision variable).
MCDM	Multi Criteria Decision Making.
MPS	Master Production Schedule.
MRP	Material Requirements Planning.
MS	Mean Square.

MSE	Mean Square Error.
MTBF	Mean Time Between Failures.
MTTR	Mean Time To Repair.
NIS	Number In System. Total number of parts in the RFS. It includes all the parts in the buffers and also on the machines getting processed. This is used as one of the measures of merit.
NISM	Mean of the Number In System.
$n_{ij}(k)$	Number of parts of type i waiting for operation j.
OPTVAR	Operation Time Variability.
OSU	Oklahoma State University.
OW	One-Way. Denotes the one-way communication strategy between the work center controller and the plant controller in which the communication is only from the plant controller to the work center controller.
PCBrain	Plant Controller Brain. The Knowledge Base (KB) which holds the rules reflecting knowledge about the plant. This knowledge base provides supervisory decisions to the Plant Controller Object.
PERT	Project Evaluation and Review Technique.
PPNM	Percentage Production Not Made. One of the measures of merit. If for product 1 the production target is say 100, and for product 2 it is 200 and if achieved production was 95 for product 1 and 190 for product 2, then PPNM is calculated as (15/300)*100 =5 %.
Pr	Probability.
RFS	Re-entrant Flow Shop.
RPT	Raw Processing Time. Processing time estimated for a part excluding the delays due to tool failures, queuing times, rework, and engineering holds.

SIMOLP	Simplified Interactive Multiple Objective Linear Programming Procedure [Reeves and Franz, 1985]. A procedure to arrive at desirable non-dominated points in an optimization problem with multiple objective functions.
SPT	Shortest Processing Time. A dispatching rule in which the next part to be selected for processing is the one with shortest processing time among all parts waiting to be processed.
SPTT	Similar to SPT but with some truncation rule superimposed to ensure that the variation in waiting involved is reduced.
SRPT	Shortest Remaining Processing Time. A dispatching rule in which the next part to be selected for processing is the one for which the sum of processing times of the remaining operations is the shortest among all parts waiting to be processed.
SS	Sum of Squares.
SSD	Sample Standard Deviation.
SV2	Denotes the complexity due to number of places where re-entrancy occurs. It is one of the factors in the design of experiments in this dissertation. It has three levels designated as SV2-L1, SV2-L2 and SV2-L3.
SV3	Denotes the complexity due to span of re-entrancy. It is one of the factors in the design of experiments in this dissertation. It has three levels designated as SV3-L1, SV3-L2 and SV3-L3.
SV4	Denotes the complexity due to number of re-entrant paths. It is one of the factors in the design of experiments in this dissertation. It has three levels designated as SV4-L1, SV4-L2 and SV4-L3.
SV5	Factor denoting complexity due to variability in machine availability. It has three levels denoted as SV5-L1, SV5-L2 and SV5-L3.
SV6	Factor denoting complexity due to process time variability. It has three levels denoted as SV6-L1, SV6-L2 and SV6-L3.

T	Big Time Period over which control is exercised. In this dissertation T is four weeks.
TAT	Turn Around Time. Another name for Flow Time (FT).
TPOFPA	Time Period OF Periodic Actions. Every TPOFPA certain periodic actions are exerted. It is one of the factors in the design of experiments in this dissertation.
TW	Two-Way. Denotes the two-way communication strategy in which communication is not only from the plant controller to the work center but also the other way.
t	Sub-period at end of which production targets are specified. In this dissertation t is one week.
t_{ij}	Operation time for operation i on part type j.
t_S	Small time interval over which the control variables are constant. In this dissertation it is one day.
$u_{ij}(k)$	Production rate of operation i on part type j (decision variable).
WCBrain	Work center Controller Brain. It is the Knowledge Base (KB) which holds the rules reflecting knowledge about a work center. This KB provides supervisory decisions to the Work center Controller Object.
WCj	Work Center j.
WIP	Work In Process. Total number of all the parts waiting in all the buffers in front of all the work centers. In this dissertation it does not include the number of parts on the machines getting processed. This is used in the linear programming formulation step.

CHAPTER 1

INTRODUCTION

1.1 Motivation for the Research

The high level of competition in modern industrial society has resulted in stringent demands on manufacturers to ensure customer satisfaction. This requires that high quality products be delivered at the right time and in the required amount. These delivery pressures from the sales division get translated in terms of pressure to meet the due dates and pressure to meet serviceability criteria for the manufacturing division. The desire to produce any quantity of any product, with minimal turnaround time requires a shop with high throughput rates, high flexibility with respect to different product types and volumes, and minimal changeover times. The designers of production equipment have therefore concentrated on designing equipment that are efficient, high speed, reliable, and highly automated. These equipment form the building blocks of flexible manufacturing systems (FMS). Simultaneously, advances in computer software and hardware have resulted in progress on the frontiers of production planning and monitoring software. Islands of automation started growing in the elusive moving target called Computer Integrated Manufacturing (CIM). However the fruits of advancements in technology can be reaped only if the technology is exploited effectively. This requires that the CIM environments be controlled effectively by exercising close control over shop floor activities.

The dynamic control and scheduling of CIM environments is currently an issue of concern to the manufacturing and academic communities [Harmonosky and Robohn, 1991]. Without an effective means of scheduling and controlling production in computer controlled manufacturing systems, it is clear that no reasonable returns can be expected from them.

The manufacturing shop can be classified as a flow-shop or a job shop. In a job shop, typically, one finds application of FMS. The literature on real time scheduling and control of manufacturing systems contains a large number of research papers on how to control an FMS. A class of manufacturing operation, not described previously in the operations management literature, has been reported in a paper [Graves et al., 1983]. This is a special type of flow shop called, re-entrant flow shop. Though there is no dearth of literature on scheduling for either job shops or flow shops, there is not much reported work on the scheduling and control of re-entrant flow shops.

The re-entrant flow shops are usually operated and scheduled as general job shops, ignoring the inherent structure of the shop flow. This could be one of the reasons for not finding substantial literature on scheduling and control under the heading of re-entrant flow shops. Examples of re-entrant flow shops are flexible machining systems and integrated circuit fabrication processes.

One of the motivating factors behind so much research effort in the control and scheduling of FMS, is the desire on the part of management to fully realize the benefits of high investments already incurred in the FMS. This requires that the FMS be utilized effectively and efficiently. The same factor, viz. high investment, is the motivating factor for desiring to control re-entrant flow shops.

This research is an attempt to fill the gap (at least partially) in the main field of control of manufacturing systems. The gap is specifically in the sub-field of control of re-entrant flow shops.

1.2 Focus of the Research

The focus of this research is to conduct investigations into the area of manufacturing control, specifically in the activities of shop control. The research proposes to develop an architecture to control a special class of manufacturing systems, viz. the re-entrant flow shop (RFS). The characterization of such a shop is given in the second chapter.

The control architecture is intended to be flexible enough so as to be applied to any re-entrant flow shop. It is intended to be applicable to any number of different part types, with their corresponding routings on several work centers of the RFS.

The scheduling rules for a standard flow shop are based on simplifying assumptions. The re-entrant flow shop does not lend itself to such assumptions. The complexities of controlling a re-entrant flow shop increase due to many other factors such as lot movements, yield effects, setup times, and machine failures. Hence in practice it is seen that a RFS (say a semiconductor wafer fab) is scheduled and controlled by a human scheduler [Economides and Cunningham, 1987]. The performance of the RFS thus depends on the quality of the decisions made by the human scheduler. Human decision makers are good in using judgment and experience in solving ill-defined problems. However, their abilities are highly limited when it comes to computational requirements. It is conjectured by the author of this research, that a control architecture that is designed to supplement the human capabilities rather than replace them will be a superior method to control the RFS. The architecture should exploit proven algorithms, heuristics and other mathematical programming techniques to make global decisions. Further the architecture should exploit the developments in the field of AI (Artificial Intelligence) to tackle the local issues/decisions. The control architecture should thus reduce the burden on the human scheduler as regards (1) computational efforts and (2) tracking of and responding to several simultaneous events.

1.3 Overview of the Dissertation

The dissertation is divided into nine chapters. The second chapter discusses the statement of the problem. The third chapter reviews the relevant literature on control of job shops and flow shops, and particularly re-entrant flow shops. Chapter four describes goals, specific objectives and assumptions in this research. Chapter five gives the details of the research plan and methodology. Chapter six presents the design of an architecture for control of the RFS. This chapter discusses the design philosophy and also discusses the implementation aspects in an object oriented framework. Chapter seven deals with the experimentation details. It presents the design of experiments carried out for evaluating the performance of the architecture. It also presents the results. Chapter eight presents the possible extensions that can be made to the architecture and discusses the scope for future research. Chapter nine summarizes the research efforts and brings out the contributions made by this research to the body of knowledge in the domain of control of manufacturing systems, specifically the re-entrant flow shop.

CHAPTER 2

STATEMENT OF THE PROBLEM

2.1 Introduction

This chapter is organized into seven sections. Many flexible machining systems and some of the automated assembly flow lines can be represented as a RFS (Re-entrant Flow Shop). The second section describes a general re-entrant flow shop. The third section describes the specific differences of the semiconductor wafer fab with respect to the general RFS. Various complexities involved in the control of a RFS are discussed in the fourth section, along with the performance measures used. The statement of the problem is given in the fifth section. The sixth section describes the approach used and, finally, the seventh section gives a brief overview of the research methodology that was employed.

2.2 Description of a General Re-Entrant Flow Shop

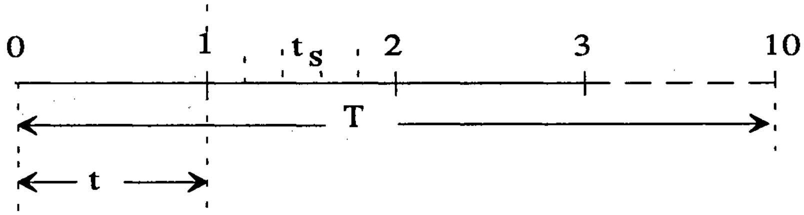
The description of a general re-entrant flow shop is divided into two subsections; the first describes the characterization and the second describes various assumptions.

2.2.1 Characterization of a RFS

There are W work centers. Each work center has identical parallel machines. There are M part types. Each part has a fixed sequence of operations over the work centers. That is the fixed routing case. The traditional flow shop assumes the flow to be unidirectional. We assume here that the part can make multiple visits to the same work center (not for rework, but as an essential step in routing). There are no alternate work centers for an operation of a part type. The transit time for a job from one work center to the next is negligible. The setup time for a part type at some machines cannot be neglected. There can be scrap in the general re-entrant flow shop. Also, there can be rework. The parts can be released into the flow shop in any quantity, not necessarily a multiple of some lot size. Thus, even one part can be released.

2.2.2 Assumptions

There is an infinite supply of raw material for all part types at the input of the flow shop. There is an infinite buffer at the output of the flow shop for storing the finished parts as they are made. Demand will be satisfied from these buffers. There is a physical buffer of limited capacity in front of each work center. There is no buffer in front of machines (workstations). Control is exercised keeping in view the next big-period (T), say the next 10 weeks. The demand for each part type occurs only at the end of each sub-period (t), say at the end of each week. Each sub-period is divided into small time intervals (t_s) say days. Thus a week is divided into 5 small time intervals. The values of the decision variables are to be determined for the small time intervals. The meaning of various time intervals becomes clear in Figure 1.



T = Big Time Period (say 10 weeks), t = Sub-Period at end of which demand is to be satisfied (say each week), t_s= Small time interval in which control variables are constant

Figure 1. Meaning of Various Time Intervals

The decision variables are, loading rates of each part type into the flow shop, and various processing rates for each part-operation combination. Even single parts can be moved from one work center to the next, in the general re-entrant shop. The processing times for each operation are known (a number for deterministic or distribution for random). There can be priority setting among jobs when they are released into the shop. There is no constraint on the stay (waiting time) of a part type at any particular operation. A part can wait for any amount of time without change in its quality, at any buffer. Machines can fail. Failure and repair distributions are known.

2.3 Description of Semiconductor Wafer Fabrication RFSs

The wafer fab is also a re-entrant flow shop. But it is more restrictive in the context of some of the assumptions given above. Specifically, part types move in lots. Thus we can release into the shop only a lot at a time. The lot's size (number of wafers in a lot) is determined by the technical considerations of the product type and the production line. The production process is characterized by some stages where the wafers may be scrapped or sent for rework. This factor is very dominant in the semiconductor wafer fab. Thus the size of the lot varies, as it progresses in the shop. The setup times for different part types cannot be neglected in some stages. Movement from one work center to the next can take place only in lots. At certain stages in the production, the wafers (lot) cannot wait for a long time as this would cause contamination resulting in reduction in yield. At the time of release into the wafer fab, the priority of a lot can be set very high so that it will be expedited at each stage. It is called a 'hot lot'. Thus setting of priority is involved at the time of release into the shop.

2.4 Complexities in Production Control of RFSs

The factors which cause complications in controlling production can be summarized as follows:

(1) Presence of a large number of work centers, through which jobs (lots) move, undergoing hundreds of operations.

(2) Typically many hundreds of jobs belonging to different product families are circulating within the line at any given time.

(3) The flow of jobs is highly re-entrant, meaning that jobs make multiple visits to the same tool group (work center) as successive operations are performed.

(4) Jobs or portions of jobs may be scrapped or sent for rework. Hence the number of parts in a job (lot) may change as the job moves through the line.

(5) Tool (work station) breakdowns, operator unavailability, major and minor setup times, unplanned maintenance and other factors combine to make the manufacturing environment highly stochastic.

The performance measures most often used to judge the productivity of a RFS can be grouped into two classes:

A. Those related to system performance:

Cycle Time (mean and variance): Also called turn around time or response time. It is the time from the instant the job is released into the shop up to the instant it leaves the shop.

Throughput Rate: The number of jobs (parts) made per unit of time.

Work In Process (WIP): A measure of all the parts/jobs waiting for processing in front of buffers of the various work centers.

Tool Group (work center) Utilization: The total productive minutes of all the work stations (machines) in a work center expressed as a percentage of the total available machine minutes in that work center.

B. Those related to serviceability:

These measures are the criteria designed to measure the degree to which the line meets the demands for the chips it produces. Obtaining robust measures of serviceability is not easy. A commonly used measure is volume serviceability. This measure is arrived at by aggregating the number of parts of all types produced by a certain due date and comparing this with the total number of parts (of all types) promised by that date. This is a very crude measure because meeting aggregated demand for parts says nothing about whether demand for a specific part type was satisfied. Another way to form a serviceability measure is to adopt a more detailed view of demand for each part type, and define a serviceability measure for each part type, and then take a weighted average of each of these measures to arrive at a single index of serviceability.

2.5 Statement of the Problem

Before giving the statement of the problem, definition of the term "control" in the context of a manufacturing system is given. Then the meaning of the term, "real time control" is discussed. After that the statement of the problem is presented.

2.5.1 Control of a Manufacturing System

Control means taking an action (or a series of actions) so as to achieve some desired goal. Defined this way, control can be either open loop control or closed loop control. In open loop control the control action is stated at the beginning of the control period, and monitoring of the system state does not take place. Closed loop control however monitors the state of the system and then takes the control action required to maintain the system state trajectory on the desired path, till the end of the control period when the desired final state is achieved.

In the context of a manufacturing system the meaning of the word control is no different. If one considers a manufacturing system in which machines never fail, and if one defines the state of the system at any time as the number of parts in each buffer, then the system can be represented by the following state equation [Refer to Appendix 1].

$$\vec{y}(k+1) = A\vec{y}(k) + B\vec{u}(k)$$

Where A is the n x n structural connectivity matrix, n = number of buffers, B is the n x m matrix that relates m input and output rates (decision variables) with n buffers, y is the state variable vector and u is the control vector.

Controlling the above dynamic system involves exerting the control actions necessary to attain a desired state. The control variables (decision variables) for the manufacturing system under consideration are, the rate at which parts are loaded (released) into the system and the rate at which different part-operations are performed at different work centers [for more details see Appendix 1].

Further, one can even decide to control the above system so as to maximize some performance measure. This is the statement of an optimal control problem, and it is shown [Tabak and Kuo, 1971] that, for dynamical systems whose state equations are of the form described above, the solution to the optimal control problem is the same as the solution to the linear program having the set of constraints described by the equation given above. Thus in the context of the manufacturing system an optimal control problem may be to find that set of controls (i.e. those loading rates and part-operation rates) which minimizes the WIP over the concerned control period.

The control view of the manufacturing system as described above is referred to as "flow control" in the literature. The control view as discussed is global in the sense that the state of the whole system (global information) is taken into consideration, and further, the control variables refer to the whole system.

In the manufacturing system under consideration, control is exercised in two ways; global control and local control. Global control specifies the decision variables such as the rate of release of parts into the system and the rate at which different part-operations are performed at every work center. The part release control is achieved by various release mechanisms which may be guided by simple rules (such as Shortest Processing Time, i.e. SPT) or by sophisticated heuristics. Generally the heuristics or the rules (such as SPT) used for the release do not explicitly consider the state of the whole system. Local control takes into account only the limited information available at the concerned work center. This type of control is exerted with the immediate objective of alleviating the problems that arise locally. One of the ways to exert such a control is by deciding the next part to be selected for processing for a given current state of the work center. Local controls are exerted by considering only the short term consequences, as against global controls which take into account the long term consequences. The local control mode allows one to use control policies which are distributed in nature. Global control can be used as the guiding beacon, in whose light distributed control can be applied.

2.5.2 Real Time Control of Manufacturing Systems

The speed at which a control system makes production decisions is a good measure of the effectiveness of the control system in controlling a manufacturing system. There is no point in having a control system whose response time is greater than the smallest time interval after which events of interest (those needing control decisions) take place, because in that case the control system will altogether miss such events. In order to maximize the performance of the manufacturing system, an effective and timely means of scheduling and control must be developed. Thus the control actions need to occur in real time. Traditionally, real time refers to immediate response to some event. The speed needed for a response may actually depend on the manufacturing system parameters such as the order of magnitude of part processing times and the flexibility of the system. For example, if the part processing times are of the order of an hour or more, then a response within say five minutes may be considered as real time response. Further if the manufacturing system has random events which occur with large periodicity, then the control can be real time.

2.5.3 Statement of the Problem

Most of the research efforts till now emphasize one aspect or the other. Thus researchers have concentrated on just "what's next" schedule using either flow rate or heuristic approaches. There are not any efforts (to the author's knowledge) which investigate the effect of job configuration decisions. Job configuration means deciding the lot size, and deciding the types of jobs in a lot. Job configuration and lot release decisions will have an impact on serviceability measures. All research efforts till now are mainly directed towards system performance improvements only, such as WIP, cycle time and throughput.

Further, the objective function is generally just one of the system performance measures. The treatment of a multiplicity of objectives is not seen in the literature. Most of the approaches do not include the effects of yield (necessitating rework or scrap). Limited buffer capacities are not considered in many approaches. Machine failures are treated by considering the MTBF (Mean Time Between Failures) and MTTR (Mean Time To Repair) in some approaches. Further there is no research effort that uses a comprehensive architecture for control. No efforts have been reported in the literature that identify the key design issues and features of such an architecture. Further, there is lack of literature on hybrid approaches which combine techiniques from operations research and expert systems in the context of RFS control. A real RFS is controlled by hierarchy of human 'controllers', typically a plant manager takes global decisions such as loading rates and lot priorities while at every work center the supervisors take local decisions such as next lot to be processed. These human controllers collaborate and interact to control the RFS. These aspects of intelleigent collaboaration and hierarchical interactions need to be captured in a research effort that investgates the issues in the design of a control architecture.

In the above context the problem can be stated as:

There is a need for a research effort to investigate the potential of a hybrid, hierarchical approach for control and scheduling of a re-entrant flow shop, in a manufacturing environment where a multiplicity of criteria of performance are important. A specific example of wafer fab cited in the literature [Glassey and Petrakian, 1989] will be used as a test case for comparing the performance of the proposed methodology with other research efforts.

The architecture must take into account multiple objectives at the global level, to arrive at global control policies. These global policies can then be used as the guiding beacon for implementing distributed control policies locally. These local controls can be a combination of traditional, and simple "what's next" scheduling rules (such as Shortest Processing Time and Last Buffer First Serve), with simple heuristics and procedures based on knowledge about the system.

The most appropriate distributed control policy at a given time and for a given state (local) can be selected under the guidance of the rules written in the knowledge base. The proposed approach must take into account realistic complexities arising due to limited buffer capacities, yield effects, machine failures, and process time variability.

Existing literature on scheduling/control of RFS is strewn with examples of control approaches with a limited view, which may be either specifying just the input sequence or release control (i.e. just global control) or which discuss just the distributed policies (i.e. just local view).

There are some approaches which are hierarchical in nature [Gershwin, 1989; Kimemia and Gershwin, 1983] but they are from a pure control theoretic view and hence do not include many of the realistic complexities cited above. The approach of the starvation avoidance policy [Glassey and Resende, 1988] takes into account only one view point, viz. that the bottleneck machines should not be starved, as the entire shop's productivity is dependent on these weakest links (bottlenecks). In one paper [Sharifnia, 1992], it is suggested that the global policies should be arrived at by a flow control approach and then local distributed control should be implemented as tracking policies that track the globally set goals. The research in this dissertation was inspired by this paper. However, this research approach will differ significantly from this paper in the following ways:

- This approach will arrive at global policies while considering the multiplicity of objectives at the global level, unlike the single objective formulation in the cited paper.
- Because of the above, a variety of objectives can be incorporated by the shop manager in an interactive manner to decide the global policies. Note that the objectives and the accent on these objectives may change from situation to situation.
- Thus, for example, the manager may be able to include not only the WIP minimization but also the maximization of the utilization of the bottleneck machines.
- Further, this research will differ markedly from that described in the paper by Sharifnia [1992] as it will include the effects of yield, limited buffer capacities and machine failures and yet arrive at effective distributed control policies guided by a rule base. Effects of yield and limited buffer capacities are included in the constraints in the LP formulation that calculates the loading rates and part-operation rates [Refer Appendix 2].

2.6 Proposed Approach

The proposed approach will attempt to devise an integrated architecture for scheduling and controlling the RFS. The main features of this approach will be:

1. Hybrid: It will be a combination of algorithmic methods from operations research and control theory with the knowledge-based techniques, employed by "controller" objects.

2. Hierarchical: The higher level of controller/scheduler (such as plant controller) will make global decisions, as in the fluid approach (also called flow control approach), taking into account the demand, capacities (available machine hours) and yields at various work centers and the limited buffer capacities. These will be the constraints while the objectives will be interactively given by the shop manager. The decisions of production rates and shop loading rates will be passed on to the lower level controllers such as the work center controller. The work center controller will then decide the rule(s) (or heuristics or algorithms as appropriate) by which each machine/work station (under that work center) should select the next job. Each machine controller then selects the jobs to closely track the goals. The hierarchy of controllers will act in a coordinated way to achieve the targeted performance measures.

3. Realistic: The approach will be realistic because it can take into account features such as yield, limited buffer capacity, machine failures, and process time variability, etc. At a higher level it can consider the multiplicity of criteria in the MCDM (Multiple Criteria Decision Making) style and not as a single objective optimization problem. At a lower level of controller hierarchy the rule base used by the controller objects can take into account multiple goals while making "what's next?" type of decisions.

4. Object Oriented: The OO paradigm will be used due to; (1) the availability of a OO simulator for discrete part manufacturing systems and (2) the ease with which the proposed complex hierarchical architecture of controllers can be implemented in the OO paradigm.

2.7 Proposed Methodology

The proposed methodology can be depicted as a flow chart given in Figure 2. As shown therein, global policy and global goal setting are performed at the beginning of every big period T, or upon initiation by the manager. The distributed control or local control is exerted on a continuous basis, in reaction to the occurrence of various events, and in response to levels of achievement of various goals. Figure 2 is only a flow chart and should not be perceived as a control architecture, as it does not depict the different "agents" that will act in order to exert the control. Further it does not show the communications that may take place between various control agents (controller objects) and the sequence of such communications. It also falls short of showing the important need of synchronization of various control actions. Obviously a linear flow chart can never show the simultaneity of occurrence of various events and supposedly process type actions on behalf of different controllers.

The portion of the architecture that is responsible for specifying the global goals and solving the vector optimization consists of a FORTRAN program which interacts with LINDO a general purpose linear programming software package [LINDO SYSTEMS, 1985]. The program interacts with the decision maker to get the requisite details about the RFS. For every objective function the program generates an LP. Then following the Simplified Interactive Multiple Objective Linear Programming Procedure (SIMOLP) [Reeves and Franz, 1985] a preferred non-dominated point can be arrived at. The resulting global policies are written in an ASCII file and read in to the object-oriented framework of controllers. The object-oriented framework of controllers is the second major part of the control architecture.

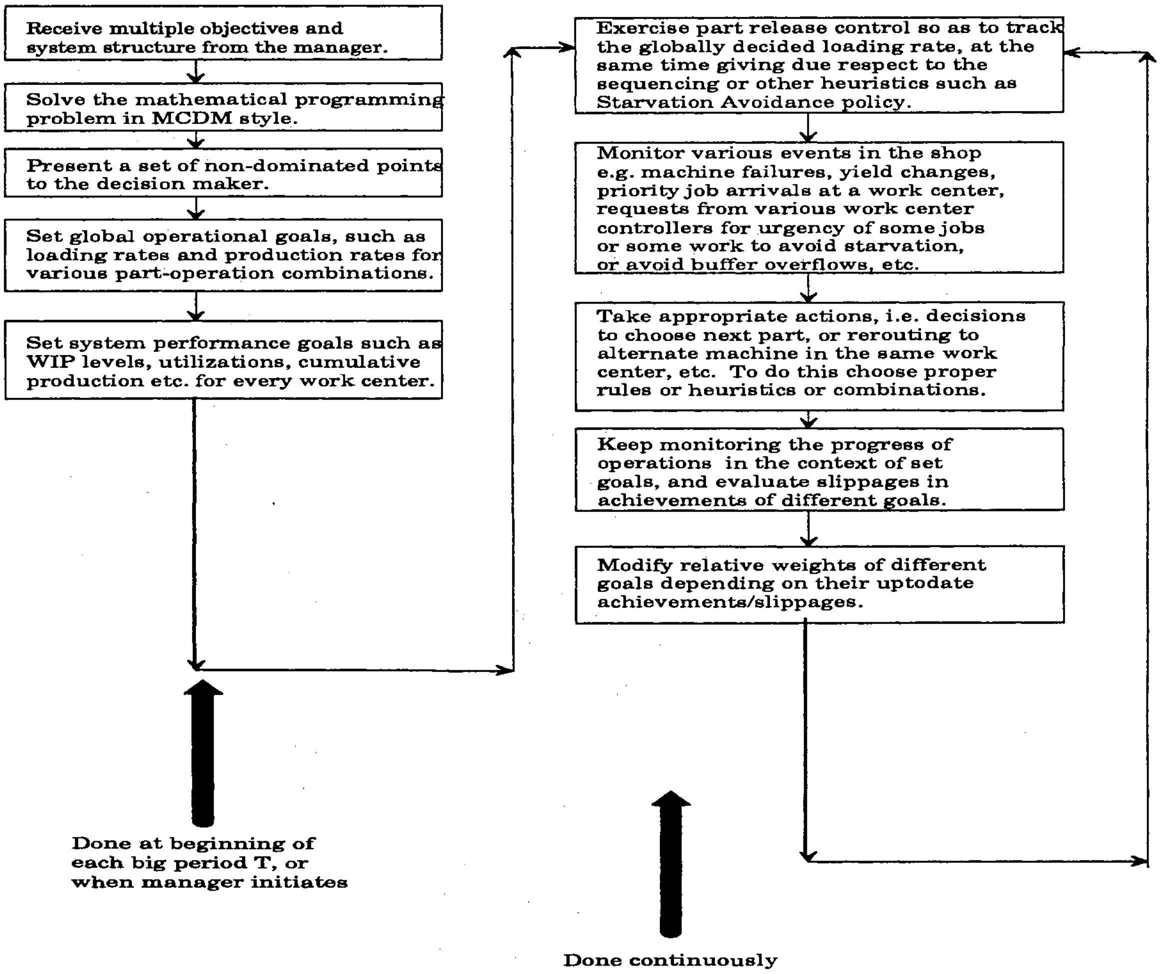


Figure 2. Flow Chart of Proposed Methodology

The Object Oriented (OO) framework was used due to the following reasons:

- The OO paradigm is well suited for modeling the controller objects. This is because in this paradigm, encapsulation of the data and methods can be easily done [Budd, 1991; Godlberg and Robson, 1985]. Further, polymorphism can be exploited to design standardized controllers at all hierarchical levels of the system.
- The simulation environment developed at the Center for Computer Integrated Manufacturing at OSU is highly reusable [Pratt et al., 1991; Bhuskute et al., 1992], and different structures of the flow shops can be created very easily along with specification for routings, setup and processing times, and limited buffer capacities.

- The control architecture uses a knowledge base and various rule bases for exercising local as well as global decisions. The knowledge base is created in the expert system shell, HUMBLE [XEROX, 1991], which is written in Smalltalk 80 [ParcPlace, 1992]. The integration of the knowledge base written in HUMBLE with the controller classes written in Smalltalk 80 is easy to achieve.
- Ease of designing interfaces in Smalltalk 80/VisualWorks.

CHAPTER 3

BACKGROUND OF THE STUDY

3.1 Introduction

This chapter reviews the literature in the domain of control of manufacturing systems. The review is divided into three sections. The second section deals with review of control approaches for CIM systems. This section will not only discuss the approaches for re-entrant shops, but also approaches for other types of manufacturing systems, such as a general job shop, FMS etc. Some of the ideas such as various sequencing rules and truncation of sequencing rules from these approaches were helpful, after some adaptation for use in this research. The third section of this chapter considers specifically the literature for re-entrant shops. Most of this work is found to be applied to semiconductor wafer fabs. The fourth section briefly reviews some approaches to the shop floor control from Multi Criteria Decision Making (MCDM) view point.

3.2 Review of Literature Related to the Control of CIM Systems

The scheduling and control literature for CIM systems can be classified into the following categories/approaches:

(1) Mathematical programming oriented
(2) Heuristics oriented
(3) Control theoretic approaches
(4) AI based approaches
(5) Simulation based approaches
(6) Interactive Approaches
(7) General approaches and combinations of above

Some important works in each category are given below. Mathematical programming oriented and heuristics oriented approaches are combined as it was observed in many important works that the researchers had formulated the mathematical programming problem, but then due to computational complexities had devised some heuristic approach.

3.2.1 Mathematical Programming and Heuristics Oriented Approaches

Several excellent review papers [Elmagharby, 1968; Bakshi and Arora, 1969; Panwalker and Iskandar, 1977; and Graves, 1981] in the field of scheduling are available. Extensive bibliographies are also available in some of the books on scheduling [Conway et al., 1967 and Rinnooy Kan, 1976].

Many researchers have been working on the flow shop sequencing problem for many years. Each one provides some heuristic to achieve good sequencing in the context of some objective function. An approach to compare the quality of the solution provided by different heuristics has been reported [Taillard, 1990]. As per this paper, the quality of the solutions provided by different heuristics can be compared by forming the distribution of the objective function and the distribution of the optima of the objective function. The paper then goes on to describe a heuristic to improve the mean quality of solutions based on the taboo search technique. In a recent paper [Cao and Bedworth, 1992], an effective heuristic algorithm for scheduling a set of different tasks to be processed on serial processors is presented that provides an approach towards minimizing the entire makespan and improving productivity. Flow shops with an inter-stage storage policy, non-zero transfer times and non-zero setup times are considered.

Joint lot sizing and scheduling for multi-stage, multi-product flow shops with capacity constraints was the focus of Pinto and Rao [1992]. The authors treat setup costs as fixed in the short run and thus independent of the number of setups. Loss of production capacity due to setup times is explicitly accounted for and the transfer of portions of a production lot between stages is permitted. The procedure is based on identifying the bottleneck work center and synchronizing the production schedules at all other work centers with the bottleneck work center such that the product throughput requirements are met with minimal inventory costs. A compact procedure for constructing the Gantt charts is also presented. Conversion of the lot sizes and transfer batches into an information control system with kanbans is another feature of this research.

In many flow shops there exists a constraint that once the processing of a job begins, subsequent processing must be carried out with no delay in the passage of the job from machine to machine except before the first machine, if necessary. Various optimizing and heuristic algorithms have been developed with the makespan objective. A recent paper reports a heuristic which results in near optimal solutions [Rajendran and Chaudhuri, 1990]. In this approach two heuristic preference relations are used as a basis for job insertion to build up a schedule by the heuristic.

Traditional production planning procedures, e.g. those used in MRP (Material Requirements Planning) systems follow a top-down hierarchical approach. They start with the generation of specific planned order releases for all final products, subassemblies and components produced. These order releases are subsequently translated into a set of tasks and due dates, and a detailed job shop scheduling problem is solved to satisfy these due dates. Since the production planning procedure ignores detailed job shop scheduling constraints, there is no guarantee that a feasible production schedule exists for the generated production plan [Lasserre, 1992]. After this argument, the author describes a modeling approach which succeeds in a systematic integration of the production planning and job shop scheduling problems. The author shows that for a fixed sequence of products on the machines, the makespan is easy to express in terms of variables in a PERT network. Using this fact the author then derives an integrated jobshop production planning and scheduling model with exact capacity constraints. Then following a (multi-pass) decomposition approach, the solution method alternates between (1) a planning problem for a fixed choice of sequence of products on the machines, and (2) a standard jobshop scheduling problem for a fixed choice of the production plan. The procedure converges to a local optimum. The procedure can be terminated at any time with a feasible plan, i.e. a plan which allows for a feasible schedule (but may fail to satisfy some of the demands).

The large-scale-event-driven nature of the modern day CIM systems requires revision of the techniques for designing a production planning and control architecture. A paper exploring this line of thought [Conterno et al., 1987] considers two manufacturing environments, batch and repetitive. The paper shows a unified approach to the production planning problem for batch and repetitive manufacturing. Starting from a common model and control architecture, specialized algorithms for the two cases are then derived. The paper considers the minimization of WIP as the production planning objective, while respecting the due dates, buffer and demand constraints. The production planning problem in a multi-stage system is then decomposed into a sequence of local minimization problems to be solved iteratively. The goal of each local optimization is to obtain the best local schedule, according to the objective function forwarded by the central coordination procedure.

3.2.2 Control Theoretic Approaches

In a significant work [Kimemia and Gershwin, 1983], the control theoretic approach to production control of a manufacturing system (particularly FMS) with unreliable machines is presented. It is a closed loop hierarchical formulation of the FMS scheduling problem. A framework for hierarchical flow control, for scheduling and planning discrete events in the manufacturing system was reported [Gershwin, 1989]. The central concept in these papers is to maintain a steady safety buffer of the parts produced in the FMS, as long as it is feasible to do so. Their hierarchy is based on the frequency of events. Decisions about events of higher frequency are made at a lower level of the hierarchy. Three levels of hierarchy are suggested. The frequency of events at a particular level is an order of magnitude smaller than that at a lower level. The top level of the hierarchy calculates the vector of safety buffer levels for each machine state. Several other modifications and/or extensions of this basic approach have been reported. Notably among them are;

[Violette and Gershwin, 1992], which specifies how to decompose the control structure for the proposed hierarchical framework for manufacturing systems,

[Maimon and Gershwin, 1988], which focuses on dynamic routing and scheduling in a manufacturing system where some of the operations can be performed by more than one machine, allowing for differences in operation times for the same operation on different machines,

[Akella et al., 1984], which suggests an alternate way to calculate the vector of safety buffer levels as an approximation.

Sharifnia [1988] derives the probability density function (PDF) of the surplus using sample path analysis and time averaging. This PDF is then used to arrive at the average surplus cost function in terms of the values of the hedging points (of the safety buffers). This average cost is then minimized to find the optimum hedging point. The problem is solved for a single product in a manufacturing system with multiple machine states.

Based on the research described in the previous paragraphs, design of a near optimal manufacturing flow controller has been reported [Caramanis and Sharifnia, 1991]. The design exploits the decomposition of the multiple part type problem into many analytically tractable one-part type problems.

A hierarchical flow control framework for the optimal flow control of manufacturing systems has been reported [Sousa and Pereira, 1992]. This framework does not use combinatorial optimization techniques as these tend to grow expensive with increasing dimensions of the problem. For this reason the authors address this problem within the framework of dynamic optimization where a realistic and yet tractable model of the manufacturing system is considered. The optimal control problem consists of finding time dependent production rates for all the operations to be performed in order to satisfy a certain demand profile while maximizing a certain performance index. The authors propose a two stage hierarchical control structure in which the first stage consists of solving an optimal control problem which defines optimal production rates for each activity type as a function of time. The second stage is required in order to fully specify a schedule which satisfies the technological constraints, i.e. flow and machine specifications.

3.2.3 AI Based Approaches

A closed loop control structure for the scheduling and control of a CIM system has been proposed [Maley et al., 1988]. Real-time feedback from the physical system monitors the performance of the current scheduling decision and updates a historical knowledge base used to make future decisions by providing initial starting solutions and guiding the search efforts. Scheduling decisions are made through the interaction of the historical knowledge base and the current system information. No application of this technology to real physical systems was presented.

An approach to scheduling and control that divides the scheduling task into four sub-tasks has been reported [Zhijun and Kai, 1990]. These are; system input control which determines the time when each part enters the system, work piece routing control which directs the parts along multiple possible routings, workstation input control which decides the sequence in which stations process the parts in their respective buffers, and vehicle control which determines the service and routes of the automated guided vehicles. The authors believe that the control of each of these sub-tasks is an event sequence control task and cannot be managed by traditional control theory. They propose a hierarchical closed loop control system composed of three levels used to control each sub-task. The first level functions as an expert system and maintains overall FMS control. The second level controls event sequences. The third level controls material flow and monitors the system status. This three-level control method is compared with the FCFS, SPT and WINQ (Least amount of work in queue) dispatching rules using an example FMS. The performance measures of interest were mean flow time, mean tardiness, mean utilization and mean queue length. It was found that the method did perform better than the dispatching rules for the measures listed.

A good survey of AI based scheduling systems can be found in [Steffen, 1986]. Steffen found that many AI approaches were currently used by the system builders but most approaches were rule based. Job shops were the most popular subject for AI approaches to scheduling. Another well documented survey can be found in [Kusiak and Chen, 1988].

The use of predicate calculus to solve planning and control problems is suggested in [Bullers et al., 1980]. It is argued that the traditional off-line analyses are too slow and may result in costly mistakes in real time environments. Hence, they advocate the use of automated controllers which have knowledge of the system and also the current status information. They have proposed predicate forms for representation of static and dynamic states of a production system.

ISIS is a knowledge based system to schedule production. Its main focus is on the constraints of the production system being modeled [Fox et al., 1982; Bourne and Fox, 1984]. The search space in the scheduling problems is very much curtailed by various constraints. The authors reported that the human schedulers spent about 80% to 90% of their time determining the current constraints and the remaining time for actually deriving the schedule. ISIS is constraint directed in the sense that constraints are used to identify the next state to go to and are also used to evaluate the current state. In case of severe constraining of search, some of the constraints are relaxed. ISIS follows a four level planning process, viz.; order selection, capacity analysis, resource analysis, and resource assignment. At each level the solution progresses in three phases; pre-search in which the current problem is constructed, constraint directed search phase, and the post-search phase where the acceptability of the solution is determined. It is not known how good the results obtained by ISIS were or how fast they were obtained. Further it tends to schedule with gaps, i.e. where a machine remains idle even if a job is available. This happens because it is waiting for a more important job which has yet not arrived [Vollman et al. 1992].

One other well known production scheduling system is OPAL [Bensana et al., 1988] which is also based on constraint directed search. There are many other such research efforts for job shops or FMS in the literature. There was no direct reference to re-entrant flow shop control architecture.

3.2.4 Simulation Based Approaches

Simulation can play a major role as a decision support tool for real-time control and scheduling of manufacturing systems. An example of this is provided in [Davis and Jones, 1988]. They present a framework for addressing real-time scheduling problems, using discrete-event simulation and mathematical decomposition to break down production scheduling problems into a hierarchical decision structure. A production planner provides input for an inter-process coordinator (IPC), which then directs the individual process controllers (PCs). The PCs contain more detailed information regarding direct process control than the IPC, resulting in distributed process control authority at the lowest levels and more aggregate system state information at the higher levels. A direct mathematical programming formulation of this decomposition approach is not feasible due to complex constraints, the stochastic nature of the process, and conflicts between multiple objectives. Therefore the authors suggested that a single processor be dedicated to the simulation of each potential job dispatching rule, such as SPT, EDD, FCFS etc. The simulation is integrated with the shop floor information in order to obtain the current system status at the time of execution. Each dispatching rule has an associated objective function that is statistically analyzed by generating an empirical PDF and calculating the minimum, maximum, mean, and variance of each objective function. Compromise analysis is then performed to determine the best rule. This is done by making an additional simulation pass with the current system status and the current best scheduling rule is run to produce an event list containing the jobs to be processed and their estimated total processing times is then generated for the IPC. The events concerning each individual PC are subsequently passed down for implementation. The success of this conceptual scheduling algorithm depends on the integration and development of several technologies, especially compromise analysis, conflict resolution, and concurrent simulation techniques. Also the authors identify a tradeoff between a guarantee of feasibility and operational efficiency due to stochastic process uncertainty. Hence optimality cannot be guaranteed.

An on-line scheduling and control framework for random FMS has been developed at the Center for CIM, at Oklahoma State University [Basnet, 1990]. This framework also uses the event driven architecture. The events as they unfold cause posting of their occurrence on an agenda. The processors of the events, in turn, post their needs on the agenda. A system supervisor takes up those requests and calls upon the relevant processors to handle the request. The methodology is based on the premise that discrete event simulation is the only analysis tool that will run on-line and at the same time ensure feasibility in the face of multiple constraints for a system as complex as FMS. The framework uses knowledge based simulation to evaluate the scheduling alternatives. In the knowledge based paradigm, the control (or decision) elements are separated from the physical elements in the discrete event simulation. This separation is advantageous from the viewpoint of modularity; changes can be made easily in the control architecture without disturbing the rest of the structure. Further it permits convenient testing of the decision alternatives.

FACTOR is an example of a commercial software product that exploits simulation for detailed scheduling. It carries out simulation using any of the sequencing rules selected by the user. It also generates Gantt charts and shop orders. FACTOR provides two standard interfaces; a modeler's interface for building and maintaining the model, and a scheduler's interface for using the model on a daily basis [Grant, 1989]. It can be used for rescheduling in case of occurrence of unforeseen events.

Expert systems have been integrated with FACTOR [Yancey and Peterson, 1989]. OAS (Output Analysis System) is an expert system that generates rulebases for analyzing a schedule. These rulebases then detect problems and suggest improvements to the schedule generated by FACTOR. Another expert system, SST (Site Specific Tailoring) is used to create rulebases for making decisions during simulation. The rulebases implement sequencing decisions, resource selection, etc.

Other simulation approaches to real time control using simulation that were reviewed, include the following: Gaffar and Cochran [1989] present a framework to facilitate shop floor decision support. Erickson et al. [1987] point out the advantages of using animation concurrently with simulation for shop floor control.

3.2.5 Interactive Approaches

The interactive approach is not a separate approach in its own right. It is in fact a feature which can be combined with any of the approaches discussed so far. This feature lends tremendous power to a control scheme. An interactive approach generally considers the human role in a supervisory capacity. The supervisor can override the controller actions. Basnet [1990] has implemented interactive features mainly to improve the quality of the schedules. The operator can fine tune the schedules created by the software. The operator can change the sequence of the jobs. Ammons et al. [1988]

outline the limitations of the algorithmic techniques and the knowledge-based techniques used for control of FMS. The authors advocate inclusion of the operator in a supervisory mode to ensure effective real time control of the FMS. The authors present a supervisory control paradigm which is based on the explicit engineering of human and automated control functions and system interfaces. "The paradigm demands two objectives from the design process. The first is that designers of automation, algorithms, and knowledge-based controls, do so with a clear understanding of how each piece relates to the human who will manage the whole system. Secondly, before an FMS control system is built, all the pieces of the system must be integrated into an efficiently functioning entity, making the best possible use of both human and equipment resources."

3.2.6 General Approaches

In this sub-section we review those approaches which do not fit easily into any of the categories previously described. There are a large number of such approaches but only representative cases are included.

Grant, Nof, and MacFarland [1989] propose an adaptive/predictive real-time scheduling and control tool. It consists of five specific modules:

(1) A scheduler which generates feasible schedules based on technology previously developed for FACTOR [Grant, 1989];

(2) A monitor which maintains the current system status, incorporating new demands and developments;

(3) A comparator, which compares the actual execution and demands from the monitor with the planned schedule produced by the scheduler. This comparison involves the use of performance tolerance fences, which define the normal performance region (i.e. an acceptable set of values for a set of variables) versus the performance region that requires a recovery strategy;

(4) A resolver, which, based on the results from the comparator, decides and selects how to respond to the system. These responses include the continuation of processing in the current manner, adaptation of the processing status to realign with the schedule (a recovery strategy), or to reschedule the entire system; and

(5) An adaptor which modifies the schedule if the resolver decides to enter recovery or reschedule mode. Decisions to recover or reschedule are based on the magnitude of the deviation of the system operation, defined by the tolerance fences. Deciding how quickly this schedule adaptation should be invoked is identified as an issue for further research. The scheduler is the only module of this proposed automated manufacturing control system which has been actually developed and implemented in the FACTOR production scheduling system commercial package [Grant, 1989; Harmonosky and Robohn, 1991].

A hybrid hierarchical scheduling and control system is reported [Bona et al., 1990], which combines operations research techniques and control theory to provide an algorithmic background for solution of the production scheduling problem. It further uses knowledge based techniques to fully take into account the complexities of the manufacturing world. The framework uses simulated annealing for schedule generation.

Hintz and Zimmermann [1989] present a hybrid framework for control of FMS. The framework solves a fuzzy linear program to arrive at a master schedule. The fuzziness is introduced in the possibility of violation of due dates to a certain degree. Then at lower levels job release and machine scheduling are performed. They derive a set of rules based on the criteria for decision making. These rules use principles of approximate reasoning for determination of priorities. The authors contend that, "By contrast to classical priority scheduling in which rather local priority rules favor strongly one or the other of the (conflicting) goals, the approximate reasoning approach uses a more global view. A large number of local rules can be taken into account and by different ways of aggregation the goals can be weighted differently. Hence solutions which prefer on a single objective can be computed as well as solutions balancing several objectives in a predetermined way. This can easily be achieved by calibrating the aggregating procedure accordingly."

3.3 Review of Literature Related to Re-Entrant Flow Shops

This section reviews literature that is directly related to re-entrant flow shops. It includes material related to semiconductor wafer fab control/scheduling and also material related to cyclic job shop scheduling.

Stecke and Kim [1991] present a flexible approach to scheduling job mixes in flow shops. An integer programming technique, used to balance machine workloads in flexible manufacturing systems is used to dynamically generate a schedule for the job classes of the mix with the aim of maximizing the utilization and minimizing makespan of the system. The approach minimizes the tool changeover time and the number of fixtures in the flow shop. Furthermore, breakdowns are handled by solving the integer program formulation of the problem subject to a new set of constraints. The size of the integer programs is dictated by the number of machines in the shop.

Graves et al. [1983] propose an algorithm for scheduling batches of identical jobs in re-entrant flow shops. Their re-entrant flow shops are equivalent to job shops, and their proposed algorithm performs cyclic job shop scheduling. Given the flow time of a job, the algorithm initiates the processing of this job as soon as this processing does not conflict with that of the current jobs. However this cyclic job shop scheduling strategy generally does not result in an optimal usage of the machines in the shop.

Shin and Zheng [1990] model an automated assembly line as a flow shop in which machines have no buffers, the constraints created by the presence of a material transport system are captured, and each batch of production is represented as a job mix. A job whose flow through the machines has n feedback loops is modeled as a job mix of n jobs of the flow shop and these jobs are scheduled individually. For an assembly line with two machines, the problem is formulated as an integer programming problem, and a solution that minimizes the cycle time of the schedule is derived. Heuristic rules are also provided for deriving suboptimal solutions to the problem of scheduling an assembly line with three or more machines, with the objective of minimizing the cycle time of the schedule.

Wein [1988] discusses the impact of scheduling on the performance of a semiconductor wafer fabrication facility. The performance measure considered by the author is the mean throughput time (some times called cycle time, turnaround time or manufacturing interval) for a lot of wafers. A variety of input control and sequencing rules are evaluated using a simulation model of a representative but fictitious semiconductor wafer fab. Certain of these rules are derived by restricting attention to the subset of stations that are heavily utilized. Three versions of the wafer fab model are studied, which differ only by the number of servers present at particular stations. The three versions have one, two, and four stations respectively which are heavily utilized (near 90% utilization). The simulation results indicate that scheduling has a significant impact on the average throughput time, with larger improvements coming from discretionary input control than from lot sequencing. The effects that specific sequencing rules have are highly dependent upon both the type of input control and the number of bottleneck stations in the fab. The author had tried combinations of four types of input control rules with 12 types of lot sequencing rules.

The scheduling of semiconductor lines can be approached by viewing the flow shop as a deterministic fluid network [Connors et al., 1992]. The fluid view was first described by Chen and Yao [1991]. This approach first allocates the work center capacity among competing job types by solving a series of linear and quadratic programming problems. Then the authors suggest the use of "what's next" scheduling algorithms designed to track these capacity allocations. The authors contend that the approach gives rise to a schedule which is based on global rather than local state information and which is responsive to the stochastic changes in the line. In this paper the authors consider only a single objective optimization problem. The authors have only alluded to tracking algorithms for what's next scheduling, but have not given any details. In a related paper [Roundy et al., 1992], the authors present the details of the "what's next" decision making. Periodically (weekly), data on the current and projected future demands are obtained, by part number. The data reflects both the quantity and the timing of the demands. Periodically (daily), the current state of the jobs in the shop is combined with the yield and estimated lead time information, to estimate the number of good chips for each part number that may emerge from each job. Also, an estimate of their likely completion date is made. For each part number a tentative assignment of the demands to the specific jobs that contain appropriate chips is made, based on the timing and quantity information. Associated with this assignment is a due date and a weight. At the end of this process, each job has a due date and a weight for each distinct part number that it currently contains. Then for each job, that set of operations is determined that has a reasonable probability of being performed during the next day. Then for every job and for every operation, a numerical measure of the urgency of performing each of these operations is determined. Finally, a constrained optimization problem is solved to compute the priorities for the operations that might be performed during the next shift.

Lou and Kager [1989] discuss a production control policy for VLSI wafer fabrication. The policy is designed to reduce the WIP in the shop floor and to follow the production plan as closely as possible. Basically it is also a flow rate control policy. This policy is formulated as a stochastic optimal control problem. The rules for lot releasing and lot dispatching are specified. Thus the policy is global in nature. The local effects of stochastic behavior, are guarded against by following the hedging point policy (similar to having safety stocks in buffers). The rules were then applied to a hypothetical two workstation flow shop. Simulation was carried out to compare the performance of flow rate control against the uniform loading policy (in which the shop is loaded uniformly by averaging the demand). It was found that the flow rate control outperformed the uniform loading policy.

A closed loop job release control for VLSI circuit manufacturing has been reported [Glassey and Resende, 1988]. As per this policy, the control is exercised by a particular closed loop job release control policy. The release policy adapts the concepts of the reorder point method of inventory control to the context of job shop (re-entrant flow shop) scheduling. The control mechanism, called starvation avoidance, is compared empirically with other input control mechanisms on several semiconductor wafer manufacturing job shops, with favorable results. In a related paper [Lozinski and Glassey, 1988], the authors present a graphical tool for inventory and production control. The tool supports a bottleneck starvation avoidance policy. Equations for calculating the bottleneck in clean room manufacturing environments are presented. A new constraint which must be satisfied to ensure starvation avoidance is introduced. This constraint relates the required amount of material within x hours of cycle time to load the bottleneck for x hours of operation. The constraint also considers the yield and safety stock. An equation is developed which describes how much work there must be in the flows to avoid starving the bottleneck.

Sharifnia [1992] develops a flow control approach for re-entrant flow shops, in which the global policies of loading rates of different part types into the RFS, and production rates for different part-operation combinations at each work center are calculated by solving a linear program. Then distributed control policies are used to make local decisions (what's next). Thus this approach advocates use of globally decided policies as guiding policies which are tracked in a distributed way by using local policies at each individual work center. The paper does not give enough details of the tracking local policies.

Stability issues of distributed policies are analyzed in a paper [Lu and Kumar, 1991]. The authors consider nonacyclic flows (re-entrant shop flow). Several distributed (local) policies are analyzed. It is shown that for a single nonacyclic flow line the first buffer first serve policy (FBFS), (which assigns priorities to the buffers in the order that they are visited) is stable, whenever the arrival rate, allowing for some burstiness, is less than the system capacity. Similarly the Last Buffer First Serve policy (LBFS) (where the priority ordering is reversed) is also stable. However, not all buffer priority policies are stable. This is shown by a counter example. The well known Earliest Due Date (EDD) policy (where priority is based on the due date of a part) as well as another due date based policy of interest called the Least Slack (LS) policy (where priority is based on the "slack" of a part, defined as the due date minus the estimate of the remaining delay) also proved to be stable. Simulation was used to provide empirical confirmation to the authors' intuition that the LBFS policy may well be the best policy for minimizing the mean delay at high load factors, while LS may well be the best policy for minimizing the variance of the delay. The authors neglect randomness due to machine failures and yield effects.

An excellent survey of developments in the domain of control of re-entrant lines is given by Kumar [1993]. The paper presents a tutorial account of some recent results in this field. Several scheduling policies are discussed, along with their stability and performance issues. Several open problems in this field are also given.

Narhari [1993], presents a Mean Value Analysis (MVA) approach to the study of the performance of distributed policies in re-entrant lines. "The approach is efficient and approximate, but promises to be accurate." The author shows how to formulate the MVA equations for a re-entrant line for studying the buffer priority-based scheduling policies. Effects of high priority jobs (hot lots) on the cycle time of other jobs in the system are examined. The author proposes that other performance related issues can be studied using this MVA technique. These include computation of variance of delay, optimization of system performance, and sensitivity analysis.

3.4 Multiple Criteria Decision Making (MCDM) Approach to Control

Any decision making situation involves choosing among alternatives. While choosing, various alternatives are judged in the context of various criteria. Traditional operations research treats this problem by optimizing only one criterion. However, multiple objectives are around us everywhere. MCDM is the subject that deals with decision making in the world of multiple criteria for choosing among alternatives. This section briefly presents the works that have considered the multiplicity of objectives for control of manufacturing systems.

While most researchers have concentrated on single objective function optimization for flow shop problems, there is a recent paper on multi-objective flow shop scheduling [Daniels and Chambers, 1990]. This research considers the sequencing of jobs through a multimachine flow shop, where the quality of the resulting schedule is evaluated according to the associated levels of two scheduling criteria, schedule makespan and maximum job tardiness. A constructive procedure is presented that qualifies the trade off between the two criteria. The significance of this tradeoff is that the optimal solution for any preference function involving only one of the criteria must be contained among the set of efficient schedules that comprise the trade-off curve. For the special case of a two machine flow shop, an algorithm is presented that identifies the exact set of efficient schedules. Heuristic procedures for approximating the efficient set are also provided for problems involving many jobs or larger flow shops.

MADEMA (MAnufacturing DEcision MAking) [Chryssolouris, 1987; Chryssolouris et al., 1988] is a framework that attempts to model the decision making process at the work center level by determining feasible alternatives, determining relevant criteria, determining consequences of the alternatives, applying decision-making rules and then selecting the best alternative. It views the FMS scheduling problem as a multicriteria decision making problem. The framework also considers the speed at which it needs to operate and is sufficiently responsive to change to make it appropriate for real-time control. It consists of several software modules written in LISP that implement the five step process described above. In the related literature cited above, two simulated test cases were presented that compared the dispatching rules FCFS, LCFS, GPT (greatest processing time first), and SPT with MADEMA using mean flowtime and mean tardiness as performance measures. For a work center with a single resource, SPT outperformed all other dispatching rules and MADEMA performed about as well as SPT. In the case of a work center with five resources, SPT performed better than all other dispatching rules, while MADEMA outperformed SPT by 10% to 20%.

CHAPTER 4

STATEMENT OF RESEARCH

4.1 Research Goal

The main goal of this research is to investigate the potential of a comprehensive architecture for controlling a re-entrant flow shop (such as a semiconductor chip fabrication facility) driven by a multiplicity of objectives, implemented in a hybrid and hierarchical manner and consisting of collaborating objects. The proposed architecture will be evaluated in the context of several complexities of the RFS in terms of certain measures of performance, such as WIP, cycle time, and percentage production not made as against target production.

The second section describes the research objectives. The steps in achieving the objectives are mentioned under each objective. The plan and procedures for achieving each of these objectives are given in Chapter 5. The third section then lists various research questions that will be addressed by the proposed research. The implemented control architecture will be used as the vehicle for answering the research questions. The fourth section gives some of the assumptions.

4.2 Research Objectives

To accomplish the research goal, the following research objectives are identified:

Objective 1 - Develop Control Architecture:

This includes the development of the main outline of the architecture for controlling and scheduling of the RFS. This will involve; identifying the main components in the controller architecture, identifying the precise manufacturing system boundary over which the architecture exercises its control (i.e. what is the domain of control of the controller?), identifying the vertical (multilevel) and horizontal connectivity requirements of various components of the controller architecture, and identifying the interactions among the components (control, information passing , goal setting, etc.).

Objective 2 - Develop Main Components Identified in Objective 1:

This will involve the following decisions or steps:

(1) Decision regarding the kind of multicriteria algorithm to be pursued for establishing the goals for plant level controllers. In particular; (a) the simultaneous objectives to be pursued, (b) the different types of constraints to be used, (c) the frequency with which the algorithm will be executed for goal setting of plant controller, and (d) the information to be used.

(2) Design of the structure of the plant level controller.

(3) Design of the structure of lower level controllers, i.e. work center controller and machine level controller.

(4) Design the interactions protocols between higher level and lower level controllers.

(5) Inclusion of knowledge about the problem domain in control decision making and the organization of the knowledge base.

Objective 3 - Develop Object Oriented Framework for Hybrid-Hierarchical Controller:

The framework will include the architecture of the controllers at each level. Thus it will describe the controller architecture in terms of, say, class hierarchy and class composition hierarchy. It will also specify the structure of each class of controller. The framework will also include interactions (among controllers) in terms of messages. The framework will also specify any synchronization routines that might be needed for coordinating the actions of various controllers. The framework will specify integration of the controller architecture with the rule base written in Humble. The rule base can contain rules which are (1) from the more general domain of a flow shop, (2) from the more restricted domain of a re-entrant flow shop and finally (3) from the specific domain of a wafer fabrication facility. The framework will also specify integration of decision making algorithms/heuristics that may be used by the controllers (for example the higher level controller might use the MCDM optimization algorithm while some of the lower level controllers might use some of the heuristics in deciding the next job to be selected).

Objective 4 - Performance Measures:

The research will address the question of the selection of appropriate performance measures for comparison. It will survey performance measures that are widely used in industry and academic circles and will specify the measures that will be used in this dissertation.

Objective 5 - Evaluation of the Architecture:

The performance of the control architecture will be evaluated in the context of several levels of different types of RFS complexities. The performance will be measured in terms of measures accepted in industry/academic circles. Further, as a test example the performance of the architecture will be compared with the performance reported in existing research publications.

Objective 6 - Further Research:

Identify further work that needs to be done to extend the results obtained in this research.

4.3 Research Questions

This section presents the research questions that will be addressed by the proposed research. The answers to these questions will be the contribution of knowledge to the domain of control of re-entrant flow shops. The research questions are divided into two groups. The first group includes the research questions that investigate the relationship of shop structure and its character to the structure of the control architecture. The second group of questions relate to the investigations that compare the performance of the controller architecture with other methods.

4.3.1 Questions Relating RFS Structure and Characterization to the Structure of the Control Architecture :

(1) What is the relationship between the complexity of the RFS and the coupling between various components of the architecture? The coupling between the components of the control architecture is described by the frequency and type of interactions between various components. The interactions are in terms of control, information passing, and goal setting. This question can be answered in the following two stages:

Stage 1: Fix the level of coupling between various components. That is, fix the frequency and type of interactions. Then vary the complexity of the flow shop to be controlled. Thus at one end of the spectrum, take a simple flow shop, which is not re-entrant at all. Then increase the level of 're-entrancy' and measure the performance of the shop, for the given architecture. Also increase the number of work centers and work stations and measure the performance of the architecture.

Stage 2: Fix the instance of the re-entrant flow shop. Then increase the coupling level between various components of the control architecture. Measure the performance for each level of coupling.

In both the above stages, experimentation is conducted on a shop in which machines do not fail and yield is 100%. Further the connectivity of different work center controllers in this stage will be only vertical, i.e. no work center controller can request any other work center controller directly.

(2) How crucial is the role of horizontal connectivity of different work center controllers in the context of increasing complexity of the RFS (complexity can be increased by increasing the number of work centers and work stations, and by increasing the level of re-entrancy)? Again, the shop used for experimentation will have no stochastic events.

This research question should answer; (a) whether horizontal coupling is required between various work centers for a re-entrant shop, and (b) how does the horizontal coupling level (this is a design decision) depend on the complexity of the shop floor?

(3) Investigate the same research questions as above, i.e. question number 1, and 2, but in the context of a shop with stochastic events. That is, investigate the performance of a given instance of the architecture by varying the stochasticity on the shop. This will be done by varying only one factor at a time (for example changing the variability in machine availability only, increasing the process time variability only, etc.).

(4) Will the type of objective function used in the MCDM block of the control architecture significantly affect the performance of the shop? Note that the achievement of shop performance measures is an indicator of the performance of the control architecture. Thus we say that if the shop has performed well for a given instance of the proposed architecture, then that instance of the architecture has performed well.

(5) Global control can be set based on a single objective (say minimization of WIP), then distributed control (local) can be used to make the "what's next" decisions under the guidance of the global policy. This has already been proposed in the literature. This research advocates that the global policies be arrived at by using multiple objectives, rather than a single objective. This will determine whether any improvement in performance occurs due to use of multiple objectives in arriving at global policies.

(6) What are the interaction effects between global control and local control? Thus for example, what is the best compatible "what's next" control rule (or heuristic) for a given lot sequencing rule (or heuristic), for a given structure of the RFS?

Answering all the above questions will lead to the body of knowledge that forms the guiding principles for designing an effective controller architecture for a given re-entrant flow shop.

The questions stated above are the major unanswered questions in this area of research. A subset of this list will be answered as the research progresses depending on the time frame and the extent of work involved.

4.3.2 Questions that Compare the Performance of the Control Architecture with Other Control Methods:

In this phase, the author proposes to employ a specific instance of the proposed control architecture that has been tuned properly (using the guiding principles arrived at as a result of answers to the research questions listed above).

(1) What should be the performance measure for comparing the proposed architecture versus other methods? Should it be a vector of different performance measures of the shop and then should a vector comparison be made? Or, should it be a combination of the performance measures of the shop and should the composite measure be compared?

(2) How does the performance of the architecture compare with other methods? A specific example of a fab given in the literature [Glassey and Petrakian, 1989] will be used. The performance of the architecture will be compared with their approach.

4.4 Research Assumptions

The research is aimed at developing a control architecture for control of re-entrant flow shops. The characterization of such a line was presented in section 2.2. The main assumptions are as follows:

The flow line will be preceded by a buffer of infinite capacity that can hold any amount of raw material for each part type being manufactured by the line. There is also a buffer of infinite capacity after the line for receiving the finished parts. The demand is satisfied from this buffer. The buffer in front of each work center has infinite capacity. There are no buffers in front of machines.

There is no random arrival of orders into the shop. Rather, the demand for every week is known in advance. Thus in a control period over say the next 10 weeks, the demand for every week is given at the beginning.

There is a higher level planning process (outside the purview of this research) which interacts with the environment (perhaps through the marketing division) and carries out the planning for each subsystem in the factory. The wafer fabrication facility is just one such subsystem, and so the load (demand) on this facility is already decided by the planning process. Further it is assumed that there are no "hot lots" introduced into the RFS. Further, the effects of yield will not be studied in this dissertation.

Further, it is assumed that the planning process is carried out such as to ensure that the wafer fab is not loaded consistently beyond its capacity.

The control architecture described here does not communicate directly with any of the machines on the floor. It is a software that will reside in a supervisory computer. So the only way to make it workable on the shop floor is to establish the links with the machines via some other hardware equipment and interfaces.

The software and hardware connections are not the issues in this research. The communication speeds and the data transfer speeds from the supervisory computer and from the other equipment are not taken into consideration in this research.

CHAPTER 5

RESEARCH PLAN AND PROCEDURES

5.1 Research Phases

The following plan is proposed to achieve the objectives mentioned in Chapter 4.

Phase 1 - Develop Control Architecture

This phase will involve development of the main outline of the architecture for control and scheduling of the fab. This phase will identify the main components in the architecture, their interactions, and frequency of interactions. To ensure that the process of architecture design is based on sound scientific footing, some methodology will have to be followed that helps guide this critical activity. Design is both an art and a science. One should not close on a design in an ad-hoc manner. There should be some way to judge the value of a design and every component of the design. What may be the systematic way of arriving at a sound design? As a response to this question, the author feels that systematic design methodologies such as value engineering and/or value analysis may be explored for possible use in this phase. This methodology has been successfully used in design of engineering products. The author also intends to look into the literature related to the generic design process.

Phase 2 - Develop Main Blocks Identified in Phase 1

Again, value engineering techniques and some of the generic design processes cited in the relevant literature will be reviewed for applicability in guiding the design process of each individual component. Further, this phase also requires the development of a knowledge base as one of the components. The author feels that the Knowledge Base (KB) will consist of knowledge from the domain of the general job shop (on which ample literature is available in the form of algorithms, heuristics, rules etc.), the domain of the re-entrant flow shops (on which enough literature is available on the issues of release control, dispatching and distributed control), and finally the domain of semiconductor manufacturing. In this final domain not much information is available as regards special practices and rules followed on the shop floor, due to the proprietary nature of the data and information. However some limited material is available from published articles and papers. Best possible use will be made of the same.

Phase 3 - Developing Object Oriented Framework For Hybrid-Hierarchical Controller

This phase is intensive from the viewpoint of programming and implementation. The simulation environment for modeling and simulating discrete part manufacturing systems developed at the Center for CIM, OSU, [Bhuskute et al., 1992], will be used. It will have to be suitably modified so as to make it compatible with the control architecture. The MCDM style of optimization algorithm will be written in FORTRAN. The algorithm will interact with LINDO available on the VAX system at the University Computing Center, OSU. The algorithm will write out the global policies in an ASCII file and these will then be read into the control architecture.

Phase 4- Performance Measures

The most obvious performance measure is perhaps the speed of the control architecture in responding to the decision making on the shop floor. This is not the only criterion. Other criteria could be those related to the shop performance. Thus if a shop controlled by the control architecture performs better than one not controlled by the control architecture then it is indicative of the better performance of the control architecture itself. Hence, shop performance criteria will be reviewed and those criteria which are most widely accepted in industry and academic circles will be selected.

Phase 5 - Evaluation of the Architecture

This phase will include two major tasks. The first relates to design of experiments to measure the performance of the shop for different scenarios, such as varying the number of places where re-entrancy occurs, varying the span of re-entrancy, increasing the number of re-entrant paths, and increasing the randomness in the shop by increasing the variability in machine availability, and increasing the processing time variability. The second stage involves a comparison with other work. In this stage, for some set of performance criteria already selected in the previous phase, the performance of the shop controlled by the control architecture will be compared with the performance reported in the literature for a specific case [Glassey and Petrakian, 1989].

Phase 6 - Further Research

In this phase the research will be critically analyzed in the context of the results obtained and the experience gained. The weak points and the gaps in the research will be brought forth, and possible remedial research directions will be mentioned. Open problems will be stated.

CHAPTER 6

DESIGN OF A CONTROL ARCHITECTURE FOR RE-ENTRANT FLOW SHOPS

6.1 Introduction

This chapter discusses the concepts which form the basis for design of a control architecture for on-line control of a Re-Entrant Flow Shop (RFS). The controller architecture consists of a hierarchy of controllers which maps into the organizational structure of the RFS. The individual controllers in this hierarchy interact with two knowledge bases to receive supervisory control decisions. The controllers make decisions pertaining to release control and dispatching. Release control is concerned with decisions on what jobs (lots) are to be released, and when and in what quantity they should be released. The dispatching decisions are concerned with deciding the next job (lot) to accept for processing at a work center when one of the work stations in the work center becomes idle. The control actions thus occur at the global level via release control and at the local level via dispatching decisions.

6.2 The Control Architecture

The general scheme of the control architecture is shown in Figure 3. Each of the blocks in the control architecture is described in the following subsections.

6.2.1 The MCDM block

This is responsible for specifying the loading rates and the rates of processing different part-operation combinations and consists of a FORTRAN program.

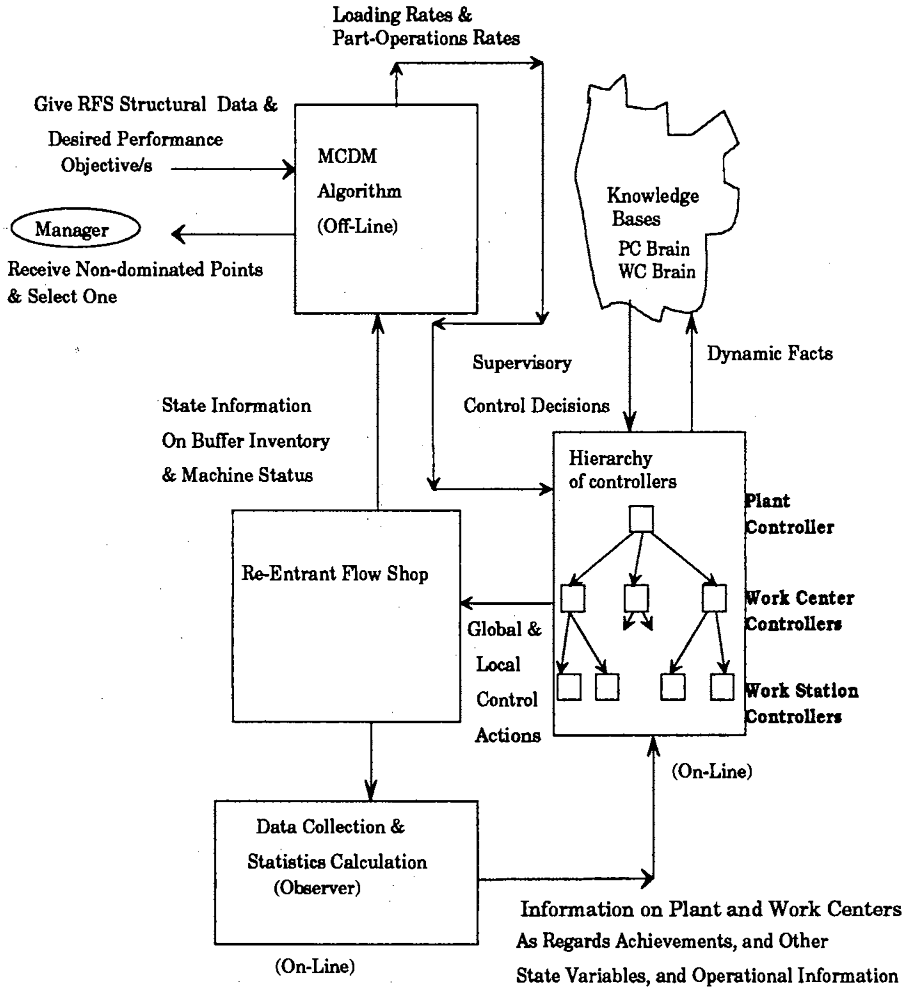


Figure 3. The Control Architecture

The MCDM program first interacts with the decision maker to get information pertaining to the structure of the RFS and desired production goals (demands for different product types and when they occur) and what objectives are to be optimized. It then generates a linear programming problem formulation. The constraints are expressed in terms of constraints for material flow balance (See Appendix 2), available machine hours for each work center as well as the initial conditions of the parts in every buffer. The LP is then solved using LINDO. LINDO (Linear Interactive Discrete Optimizer 1985, LINDO SYSTEMS) is a commercial software package used for solving Linear, Integer and Quadratic programming problems. Thus the decision maker can specify different objectives, and for each a LP is solved. Then by following the algorithm presented in

Simplified Interactive Multiple Objective Linear Programming Procedure (SIMOLP) [Reeves and Franz, 1985] a non-dominated point as preferred by the decision maker can be selected. The resulting global policies (values of loading rates and part-operation rates at each work center) are written to an ASCII file and read into the object-oriented framework of controllers. Typically the algorithm is run offline at the beginning of each big time period T. However the decision maker can choose to run the algorithm at any time when the conditions in the shop have changed significantly, for example, when many machines have failed, or production targets have changed significantly, etc. Every time the algorithm is run the constraints related to the initial buffer contents should reflect the actual number of parts waiting in each buffer. Note that the decision maker does not necessarily have to specify multiple objectives. A single objective can also be used to arrive at the loading rates and the part-operations rates. All the experimentation in this dissertation was carried out with the single objective of minimizing WIP throughout the RFS. However, in section 8.2 an example is presented which compares the performance of a RFS using multiple objectives as against a single objective in arriving at loading rates and part-operation rates.

The object-oriented framework of controllers is the second major part of the control architecture. The object-oriented framework is suggested due to the following reasons:

- The OO paradigm is well suited for modeling the controller objects. This is because in this paradigm, encapsulation of the data and methods can be easily done. Further, polymorphism can be exploited to design standardized controllers at all hierarchical levels of the organization.
- The control architecture uses the knowledge base for getting the supervisory decisions affecting the local as well as global control actions. The knowledge base can be created in the expert system shell, HUMBLE [XEROX, 1991], which is written in Smalltalk 80

[ParcPlace, 1992]. The integration of the knowledge base written in HUMBLE with the controller classes written in Smalltalk 80 can be easily achieved.

6.2.2 The Plant Controller

The controllers are arranged hierarchically. The hierarchy corresponds to the organizational hierarchy of the RFS plant. At the top the Plant Controller is responsible for global control actions to be exerted at the plant level. The Plant Controller gets the rates of loading and rates of producing different part-operation combinations at different work centers from the MCDM block of the architecture. These rates calculated by the MCDM architecture do not take into account the delays that occur due to waiting at different queues. So at the plant controller level, allowance is given for flow time to be about three times the total process time. This multiplier is in keeping with the stress on JIT (Just In Time) procedures. Pravin Johari in his paper on "Engineering a Circuit Board Assembly Line for a Desired Capacity and Flowtime" [1991] has put forth this idea. The author states that "With the stress on just-in-time (JIT) procedures, it was felt that the desired average flowtime should be no more than three times the average raw processing time...". David J. Miller [1989] states that, "A realistic target for a development line was determined to be 3 × RPT". David Miller is referring to the target for TAT (Turn Around Time, i.e. Flowtime). In the industry TAT is expressed as a multiple of RPT (Raw Processing Time). RPT does not include tool failures, queuing times, waiting, rework and engineering holds. In the paper by David J. Miller though the target for TAT was planned to be 3 times the RPT, at the end of the study TAT could be reduced from 6 times the RPT to 4.5 times the RPT. Such existing practices in the industry lead the author of this dissertation to use a multiplier of 3 for targeting the flowtime based on the RPT (Raw Process Time). This helps in deciding the appropriate release time for a lot. Further, the rates for the final operations of a product determine the due dates for different lots waiting to be released. Because an infinite supply of raw material is assumed, all due dates can be set at the beginning of a big time period.

Typically the Plant Controller carries out the following duties:

1. Consults a knowledge base called Plant Controller Brain to get the supervisory decisions pertaining to;
    - maximum number of parts allowed in the RFS (i.e. increase, decrease or no change),
    - whether to fire a message to arrange the part-type names as per achievement in cumulative loading against cumulative target in loading,
    - whether to fire a message to arrange the part-type names as per cumulative achievement in their completion as against the cumulative completion targets.
2. Gets information on the state of the plant from the Observer Objects (Data Collection and Statistical Analysis Objects), converts this quantitative information into suitable form as dynamic facts about the plant under control, and passes these on to the Plant Controller Brain during the consultation.
3. Calculates the cumulative loading goals for each product type.
4. Sets priorities for release among products depending on the gap in actual loading versus cumulative goals in loading.
5. Sets the maximum number of parts allowed in the plant.
6. Exercises the release control decisions pertaining to the number of lots to be released, the product for which the release is to be made, and the timing of the release.
7. Curtails the release of lots for a particular product type if so required depending on the over-achievement in production of that product.
8. Communicates to each work center controller; the beginning of the small time interval, the required rates for processing of different part-operation combinations

performed at the work center, and priorities among part types for processing at the work center.

6.2.3 The Work Center Controller

The second level of controllers consists of work center controllers. For each work center (physical grouping of similar work stations) there is a corresponding Work Center Controller. Each Work Center Controller has Plant Controller as its super controller. The Work Center Controller acts as the super controller for the Work Station Controllers below it.

The Work Center Controller typically performs the following duties:

1. Consults a knowledge base called Work Center Controller Brain pertaining to the dispatching rule to be used for selecting the next part from those waiting for processing. <u>This is done periodically.</u>
2. Interacts with the Observer Objects (Data Collection and Statistical Analysis Objects) to get the statistical information regarding the state of the work center being controlled, i.e. such information as the queue length statistics, achievement in part-operation completion as against required completion targets, coefficient of variation of waiting involved for different parts, etc. <u>This is done periodically.</u>
3. Converts the above information into suitable form to create dynamic facts about the work center under control and passes these facts to the Work Center Controller Brain during consultation. <u>This is done periodically.</u>
4. Arranges the names of different buffers for its work center as per SPT (Shortest Processing Time), LPT (Longest Processing Time), SRPT (Shortest Remaining Processing Time), etc. at the beginning of the big time period (i.e. once only). Note that in front of the work center there might be just one physical buffer, but it can be viewed as several logical buffers depending on the part-operation combination. Thus for a certain work center, product 1 might visit for operations

1 and 7, and product 2 might visit for operations 2 and 5, resulting in 4 logical buffers which can be arranged as per the buffer priorities mentioned above.

5. Arranges logical buffers (pointers to them) as per the gap in the achievement in the processing rates for them with respect to the target processing rates. This is done periodically.
6. Communicates to the Work Station Controller the next lot to be processed by the work station. This is done whenever the Work Station Controller requests for the next lot to be processed.

6.2.4 Plant Controller and Work Center Controller Coupling

The Plant Controller and the Work Center Controller are coupled vertically. The coupling comes due to the communication strategy (command from the plant controller and requests from the work center controllers), the time period of periodic actions and the passing of information between the two. In the control architecture, two strategies of communication between the Plant Controller and the Work Center Controller were tested. These two strategies are described next.

(1) If the work center under control of a Work Center Controller has the first operation for any part, then the Work Center Controller requests the Plant Controller for release, if required, of a lot. The timing of such requests (lot release requests) is tied with two alternative strategies of coupling between the Plant Controller and the Work Center Controller. One strategy is to request the Plant Controller every time the first operation on any part is completed in the concerned work center, and also make such requests periodically. Thus in this strategy the work center controller has an upward communication link with the Plant Controller. In addition, the Plant Controller also can command periodically any lot release, if required. This is the usual downward communication. Thus the first strategy is to have two-way communication in which the lot release decisions are initiated by either Plant

Controller (command) or in response to the requests from the Work Center Controller. This is called the two-way communication strategy.

(2) In the second vertical communication strategy, communication is one-way, that is from the Plant Controller to the Work Center Controller (downward command only). This is done only periodically. This is called the one-way communication strategy.

Since the release decisions depend on the previous supervisory decisions such as the maximum number of parts allowed in the system, the chosen release rule and the priorities among the parts as regards loading, it is felt that the two-way communication strategy should function better. It is timely, uses the latest state information and the latest supervisory decision regarding the maximum number of parts allowed in the RFS and hence there are opportunities for the controller to take control actions that are timely and more refined as regards the number of parts allowed in the system, priorities among the parts, etc. In the next chapter, the communication strategy is one of the factors in the factorial design of experiments.

6.2.5 The Work Station Controller

At the lowest level in the controller hierarchy, we have the Work Station Controller. For each work station (or machine) one Work Station Controller exists. The Work Station Controller has a Work Center Controller as its super controller. Work Station Controllers do not take any part in the decision making process pertaining to selection of the next lot to be processed. A Work Station Controller is intended to be an object that communicates with the work station and keeps up-to-date information pertaining to the status of the work station (i.e. idle or busy and down or up). It communicates this to the work center controller. The Observer Objects (Data Collection and Statistical Objects) tap information from the work station, via the work station controller. In the future, if this controller architecture has to be integrated with the hardware components, then the sensors on the work station will send signals to the Work

Station Controller regarding status of the machine. Thus the Work Station Controller will act as the hook where integration (connection) of the software controller architecture will take place with the hardware employed on the factory floor. The focus of this research was not this kind of system integration and hence the Work Station Controller has not been developed further. In the future however it could be developed to include communication facilities.

6.2.6 The Plant Controller Brain

The remaining two parts of the control architecture are the two knowledge bases, viz. the Plant Controller Brain, and the Work Center Controller Brain. Both of these knowledge bases are written in HUMBLE, a XEROX product written in Smalltalk-80. The knowledge bases written in HUMBLE can be easily integrated with the objects in a Smalltalk-80 environment. One can invoke consultation with the knowledge base from any object in the Smalltalk-80 environment. In this sense the Smalltalk objects are the entities requiring the supervisory decisions, the knowledge base is the expert and the invocation of consultation requires that the two-way communication be established. This is done by including interrogator behavior in the Smalltalk-80 objects. The knowledge base can ask the object for any piece of information (facts) for use in the rules. A powerful facility in HUMBLE is that while in the midst of a consultation, as part of the action one can send a regular Smalltalk-80 message to the Smalltalk objects. This feature can be used in a variety of ways. For example, one can use it to change the state of the object requiring consultation or for doing complicated mathematical calculations (which HUMBLE is not so good at doing), etc.

The Plant Controller Brain (PCBrain) is the Knowledge Base that contains the knowledge or the rules about the Plant. Figure 4 shows the flow of necessary state and performance information from the RFS to the PCBrain. The PCBrain uses these dynamic facts when the Plant Controller invokes the consultation with the PCBrain, to arrive at the supervisory control decisions at the plant level (global level).

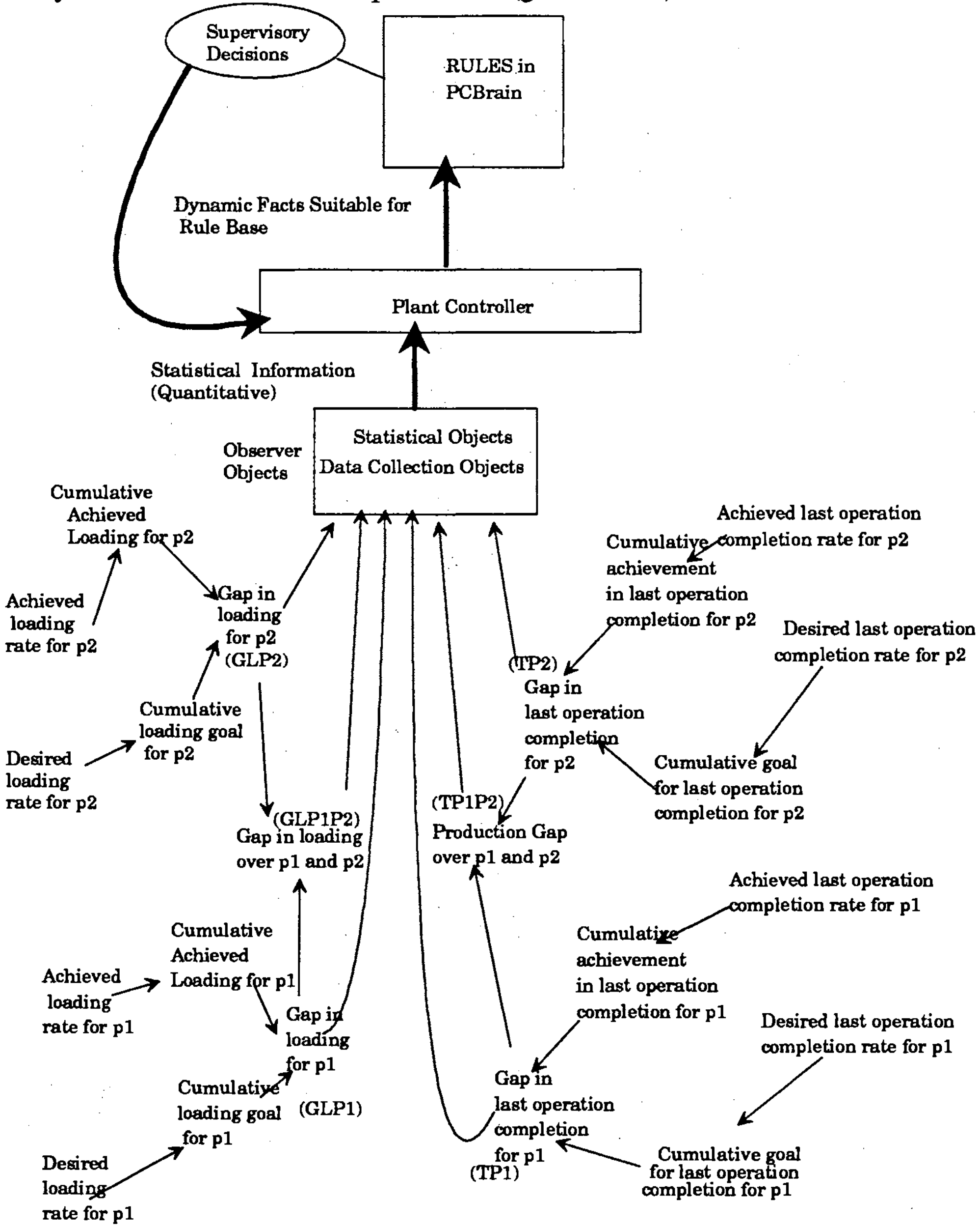


Figure 4. Flow of Information from RFS to PCBrain

The rules can be of a very general nature related to the dynamics of the plant or they can be site specific, that is specific to the plant to be controlled. Presently the rules in the Plant Controller Brain are very simple and general, i.e. pertaining to only the dynamic aspects of release control and to the assignment of priorities for the loading of different parts. There are eight such rules [Gharpure, 1994]. One of the rules in the Plant Controller Brain is;

IncreaseIfHighTardinessAndHighGapInCumLoading

if: (((avgMeanTardiness = 'high') | (avgMeanTardiness = 'medium')) & (gapInCumLoading = 'highPlus'))

then: [decisionOnAllowedNoOfPartsInPlant is: 'increase' withCertainty: 0.9]

else: [decisionOnAllowedNoOfPartsInPlant is: 'noChange' withCertainty: 0.5].

In the context of Figure 4, we can interpret this rule as

If (((TP1P2 = 'high') | (TP1P2 = 'medium')) & (GLP1P2 = 'highPlus'))

then: [decisionOnAllowedNoOfPartsInRFS is: 'increase' withCertainty: 0.9]

else: [decisionOnAllowedNoOfPartsInRFS is: ‘noChange’ withCertainty: 0.9].

Thus if there is no gap in cumulative loading goals over all products and if there is no gap in the cumulative goals for completion of final operations on each part type then no action is needed. But if the required number of parts cannot be loaded into the system (indicated by GLP1P2 to be 'highPlus') and also if the desired production rate for the final operations on the part cannot be achieved, then the maximum number of parts allowed in the plant is too few. If one were to continue to work with this number then the utilization of the machines would be low, as there is not enough work. Thus it is important that a decision be taken to increase the maximum allowed number of parts in the system (Plant). Note that this supervisory decision directly tells one to change the policy parameter viz. the maximum number of parts allowed in the system. A similar rule can cause the number allowed in plant to be reduced or the decision can be to not cause any change in this number.

Other rules in the rule base are concerned with the achievement in the loading goals of the individual part types and will cause a message to be sent to plant controller class instance if individual part type loading goals are not achieved though the cumulative goal might have been achieved. Achievement in the cumulative goal but not of individual part types means that some part types are being over-achieved while some other are being under-achieved in loading goals. This situation causes the rule base to tell the plant controller object to send a message to "self" that arranges the part types in order of loading achieved and then some release rules that use this ordering can be used while deciding which part to be released at the time of the next release decision. Similarly, other sets of rules in the knowledge base seek the state of affairs as regards the achievement in the cumulative targets on completion of final processing steps and may cause a message to be sent that will cause the plant controller to arrange the part types as per this achievement in the final processing step. In this way the Plant Controller Brain causes the Plant Controller to update the list only when needed. This takes away a lot of unnecessary burden from the Plant Controller and speeds up its response time.

It should be noted that the Plant Controller is the mediator between the Observer Objects and the PCBrain. That is, the Plant Controller taps the requisite information from the Observer Objects and post-processes this quantitative information in a suitable format for conversion into the dynamic facts. During consultation these facts are then used by the PCBrain to arrive at the supervisory decisions.

In summary, supervisory decisions from the PCBrain pertain to:

(1) increasing or decreasing the maximum number of parts allowed in the RFS

(2) telling Plant Controller to arrange the part type names in their priority, based on the gap in production which is reflected in under achievement of their production by the desired due date

(3) telling Plant Controller to arrange the part type names in the order of the gap in cumulative loading

(4) telling Plant Controller what release rule is to be used.

<u>When are these decisions taken</u>? Every time (after every CDT (Control Dt)) Plant Controller asks the PCBrain for these decisions. This consultation is invoked by the message **exertPeriodicControlAction**. This message is like the gateway from the Plant

Controller to the PCBrain for consultation. Note that the supervisory decisions are taken periodically because this message is fired periodically. One can change the frequency of this decision making by changing CDT. Also every time the **exertPeriodicControlAction** is fired at the plant level, it is also fired correspondingly at the work center level. The message selector is the same only that this time it is understood by the work center controller objects (a case of polymorphism). This message is the gateway for the Work Center Controller onto the WCBrain.

6.2.7 The Work Center Controller Brain

The second knowledge base is the Work Center Controller Brain. This knowledge base at present contains the rules that are not specific to any characteristics of a work center. These are very general rules and pertain to knowledge from the scheduling field. The supervisory decisions made by this knowledge base pertain to the dispatching rule to be used. This dispatching rule will be used by the Work Center Controller for dispatching purposes until the next consultation.

The Work Center Controller is the mediator between the Work Center Controller Brain and the Observer Objects. The Work Center Controller converts the quantitative information from the Observer Objects into dynamic facts and these facts are then used by the Work Center Controller brain for determining the scheduling rules to be used. The quantitative information is on such aspects as the waiting involved at the work center, the coefficient of variation of waiting times, queue lengths, and achievement in the cumulative production for part-operation as compared to the specified goals.

In addition to general rules one can use site specific rules to capture the characteristics of a work center. For example assume that at a certain work center, if the waiting becomes excessive then the yield of that lot will be affected drastically. Under such a situation one can write the rules that will capture the desire on the part of the management to enforce the discipline that the lot waiting longest and having minimum slack with respect to ceiling on waiting at the work center will be selected first. If the longest waiting lot has slack with respect to ceiling on waiting (ceiling on waiting minus actual waiting) that is larger than a pre-specified number, then this rule might not be fired. Such a work center is very common in the wafer fab, particularly a clean room where excessive waiting time might reduce the yield. Such rules are not included for the present in the knowledge base. The scheduling rules that might be selected at the end of a typical consultation are from EDD (Earliest Due Date), EDDT (Earliest Due Date Truncated), SPTT (Shortest Processing Time Truncated), or LWait (Longest Waiting lots to reduce the standard deviation of waiting time in queue). Presently there are no rules in the knowledge base that will tune the parameters of the scheduling rules themselves. Thus the truncation parameters themselves are fixed.

It should be noted that though there is only one Work Center Controller Brain knowledge base, there can be several work centers which would consult the knowledge base. And further, depending on the state of the work center (local information) the dispatching rule chosen by the knowledge base for that work center can be different from the dispatching rule at some other work center.

6.2.8 Periodic Actions

Throughout the above discussion there has been reference to periodic control action or periodic consultation or periodic updating of state information. The following paragraphs explain the periodicity (or when) of each periodic action.

Updating of information

This is always done only periodically, irrespective of the communication strategy employed, i.e. one-way or two-way.

Consultation

- In one-way and in two-way communication strategies, the Work Center Controllers consult WCBrain only periodically.
- In one-way, the Plant Controller consults PCBrain only periodically.
- In two-way, the Plant Controller consults the PCBrain periodically as well as when requested by the Work Center Controller of the work center with first operation.

One of the factors in experimentation is the time period after which the updating of information and other periodic actions take place. It is denoted as the TPOFPA (Time Period Of Periodic Actions) in the design of experiments in the next chapter. It is one of the factors that is varied in the experimentation stage (next chapter) to see how the performance of the controller varies for a given level of structural complexity or complexity due to randomness.

The important message that is fired periodically is, **exertPeriodicControlAction**. This message is understood by both the Plant Controller and the Work Center Controller classes. The message causes a host of things to be done. Details are presented in the next section in which implementation details are discussed. The period after which this message is fired is the time called control dt or CDT in the Smalltalk code. The next section presents some implementation details in Smalltalk-80 as regards classes, etc.

6.3. Object Oriented Implementation Details

The previous section presented the architectural details and the philosophy behind the controller actions. In this section the implementation details are presented to facilitate the understanding of the detailed code given in Gharpure [1994]. Several classes were designed and implemented in VisualWorks Release 1.0. These can be broadly categorized as classes responsible for modeling of the RFS, classes responsible for creation of a simulation model and model execution, classes responsible for collection of data and statistical processing, and classes responsible for control. The last category of classes is of primary importance. However the other categories are described first. A complete listing of all the classes is given in Gharpure [1994].

6.3.1 Classes Responsible for Modeling of RFS

These classes are modified versions of the classes designed in the Advanced Modeling Methodologies project being conducted in the Center for Computer Integrated Manufacturing at Oklahoma State University. The major classes are:

Plant: This class represents the whole RFS. It has infinite capacity storage at its input where raw materials can wait. It has infinite capacity storage at its output where finished parts can wait. The plant can have work centers within which there can be work stations. The plant has product information as regards the weekly production targets for each product type. The plant has a pointer to the plant controller.

WorkCenter: An instance of this class is contained within a plant. A work center has buffers at the input and at its output. These can be of finite capacity. However in this dissertation the buffer capacities are considered infinite. The buffers are for each part-operation combination. The work center has a pointer to its controller.

WorkStation: An instance of the class WorkStation is a work station (machine). It is contained within a work center. It has no buffers. It can fail. It has setup and processing times, and these can be from any probability distribution. It has a pointer to its controller. The state of the work station is identified as up/down coupled with busy/idle. A work station can be down while holding a part. It can be up and idle or up and busy. It can also be down and with no part. In this dissertation it is assumed that when a work station is processing a part it can fail but the part is not rejected. Further when the machine is up again, the processing of the part starts from where it stopped.

Buffer: This is a class, an instance of which is attached to an instance of the class work center. It can hold instances of the class Lot.

Lot: The instance of this class is a lot. A lot can have several instances of the class WorkFlowItem. A lot moves from a work center to the next work center. A lot moves only when it is completed. A lot starts with a certain number of work flow items in it. As it proceeds, this number may be reduced as some work flow items are scrapped during processing. Though this capability is provided in the class design, it is not used, as in the dissertation the scrapping of parts is ignored for all the experimental runs.

WorkFlowItem: The instance of this class represents a work flow item, i.e. a part. This moves from one work center to the next only in a lot.

6.3.2 Classes Responsible for Simulation and Creation of Simulation Model

These classes are modified versions of the classes designed in the Advanced Modeling Methodologies project. The classes are:

SimModel: An instance of this class is the model that contains a complete representation of the Re-entrant Flow Shop. It has an instance of Plant, which contains instances of WorkCenter, which has instances of WorkStation. It also has instances of Lot for each part type.

Simulation: An instance of this class holds an instance of the class SimModel which is to be simulated. This class also has information as regards the number of runs and the ending time of the simulation. It has an event calendar on which instances of the class Event are posted. This class is responsible for execution of the SimModel.

Event: The instance of this class represents an event which is posted on the event calendar of the Simulation class instance. Each event is characterized by an effector, a selector and arguments. Thus when an instance of the Event class is created, the effector, the selector, and the arguments have to be specified. The effector is an instance of a class such as Plant, or WorkCenter or WorkStation. The selector is the name of the message that is sent to the effector when the event is removed from the event calendar and executed. The arguments are the instances of the objects that may be required when the message is sent to the effector. There may or may not be any argument for an event.

6.3.3 Classes Responsible for Data Collection and Statistical Processing

There are several classes in this category. These classes were designed as part of the Advanced Modeling Methodologies project at Center for CIM, OSU. The classes can be categorized into classes for data collection and classes for statistics. The instances of classes for data collection can be plugged into the object from which the data is to be collected. Thus a data collecting class can be plugged into say an instance of a Buffer class for collecting the state related data, i.e. number in queue. These observations are then sent to a statistics collection class which will arrive at such information as queue length and queue delay and this can be represented in the form of detailed numbers, i.e. mean, sample standard deviation, minimum value, maximum value, etc. or it can be represented in the form of a histogram.

6.3.4 Classes Responsible for Control

These are the controller classes developed in this research explicitly for the control of Re-entrant Flow Shops. There are three classes, viz. PlantController class, WorkCenterController class and the WorkStationController class. They have "has-a" hierarchy. Thus, an instance of PlantController has one or many instances of the WorkCenterController class, and an instance of the WorkCenterController class has one or many instances of the WorkStationController class. Each controller class instance has a pointer to its super controller and also pointers to its subordinate controllers. The instance of PlantController has no super controller, while an instance of the WorkStationController has no subordinate controllers. Each controller instance also has a pointer to the controlled subject that it is controlling. Thus an instance of the class Plant is the controlled subject of an instance of the class PlantController. Each class is described in more detail below.

PlantController: Salient methods in this class are described.

simInitialize This message is sent to the plant controller at the beginning of the big time period T, i.e. at time t = 0.0. This message causes;

- sending the simInitialize message to each of the work center controllers
- placing of an event called begnOfSTIEvent on the event calendar
- setting up of daily loading required for each part type
- creation and posting of the monitoring event mEV, whose effector is self, selector is exertPeriodicControlAction and the argument is time at which the next periodic action is to be exerted
- assigning of due dates to all parts

executeBeginningOfSmallTimeInterval The event **begnOfSTIEvent** (placed by the simInitialize message at the beginning) has effector as PlantController instance and the selector as this message. When the event **begnOfSTIEvent** is removed from the event calendar and executed, this message is sent to the effector, i.e. the instance of the class PlantController. The message causes;

- transmittal of current STI (small time interval) to each subordinate controller
- arrangement of buffers as per achievement in processing rates
- arrangement of buffers as per percentage gap in processing
- arrangement of buffers as per required processing rates
- sending of the message ***releaseIfRequiredLotForAnyParts*** to self
- creation of another **begnOfSTIEvent** and placement of the same on event calendar

releaseIfRequiredLotForAnyParts This message causes release of lot(s) for part(s) if required. In doing so it fires a message corresponding to the current release rule as decided by the knowledge base PCBrain.

exertPeriodicControlAction This message is the selector of the event **mEV** which is placed for the first time when simInitialize is sent to the PlantController. This message is sent to the effector of the event **mEV**, i.e. PlantController. It causes;

- consultation with the PCBrain (the knowledge base for Plant Controller)
- gets two supervisory decisions, viz. the decision on maximum number of parts allowed in the plant, and the release rule to be used for release control
- causes the message ***releaseIfRequiredLotForAnyParts*** to be sent to self
- causes a new **mEV** event (monitoring and control event) to be posted

All of the above messages are fired in both the communication strategies, i.e. one-way as well as two-way. Note that in the above messages there is no message sent from WorkCenterController to the PlantController. The PlantController initiates the action. The message ***releaseIfRquiredLotForAnyParts*** is sent by itself to self.

In addition to the above messages there are several categories of messages as follows;

Messages related to different release rules: These are different algorithms for releasing lots to the plant. One can add to these algorithms depending on the site specific algorithms used in a real life RFS. The main two release rules are ;

- ***releaseAsPerLargestGapInLoadingAndWLAndJobPrioirty*** and
- ***releaseAsPerLargestGapsAndNoOfPartsInRFS.***

The first release control rule considers the parts with largest gap in loading (desired minus achieved), the priority for jobs and the work load present in the RFS. The job priorities are set by the PCBrain periodically. The AI rules for setting job priority can be extended from those presently implemented depending on the practices followed in the organization. The second rule considers just the number of parts currently present in the RFS and the gap in loading for each part type.

Messages related to updating of performance details: These messages cause updating of the achievement in loading on a given day, updating of cumulative loading achieved till date, updating of the gap in loading, arranging of part type names as per gap in loading, etc.

Messages related to Interrogator Mimicry: The PlantController receives consultation from the PCBrain. For this, HUMBLE requires that the PlantController be able to answer queries posed by the PCBrain during a consultation. The messages in this category are the messages which answer such questions. Some examples are avgMeanTardiness, gapInCumLoading, etc. Throughout the implementation of the controller architecture the word "tardiness" has been improperly used in the Smalltalk code. Actually the author means percentage production not made. Thus 5% tardiness actually means that 5% of the production could not be achieved by the required due date. Tardiness is actually derived from minimum of 0 and required due date minus actual achieved delivery date. This is not meant in the code wherever the word tardiness is used in the message selectors.

WorkCenterController: The important messages are;

simInitialize This message is sent from the ***simInitialize*** method for PlantController.

It causes;

- setting up of daily required production
- setting up of daily required cumulative production
- arranging of buffers as per SPT (Shortest Processing Time)
- sending of the message ***simInitialize*** to each subordinate controller
- creation and posting of the monitoring event **mEV**, whose effector is self, selector is **exertPeriodicControlAction** and the argument is time at which the next periodic action is to be exerted.

exertPeriodicControlAction This message causes;

- arranging of buffer names as per processing gap and as per queue length
- initiation of consultation with the WCBrain to get the scheduling rule to be used till next periodic control action is taken
- sending of message ***allotALotToEachIdleWSTN***

- posting of the next **mEV** event

allotALotToEachIdleWSTN This message causes the allotment of a lot (if an unallotted lot is available) to each of the idle work stations in this work center.

arrivedALot: aLot This message is sent to the WorkCenterController every time a lot arrives from the previous work center to this work center. This message then sends to self the message ***allotALotToEachIdleWSTN.***

<u>All the above messages are present in both communication strategies. However in two-way communication strategy, the following happens:</u>

When a lot is finished at any work center, the WorkCenterController requests a routing from the plant routing dictionary. Now when the lot is to be routed it is checked to determine if the stage at which the lot was completed was stage 1, i.e. first operation. If it was the first operation then it is checked to determine if the control type is CL (two-way). If it is CL, then the WorkCenterController sends the message ***releaseIfRequiredLotForAPart: pn*** to the PlantController. Here **pn** is the name of the part type for which the lot was completed. Upon receipt of this message, the PlantController first checks if it is necessary to release a lot for part named pn. If not, then the PlantController sends to itself the message ***releaseIfRequiredLotForAnyParts***. This message now decides if there are any other parts for which the cumulative loading goal is not satisfied.

The remaining messages can be classified in various categories as follows:

<u>*Messages that cause updating of information:*</u> These messages cause the updating of achievement in processing goals for each buffer in that work center, updating of ordered list of buffer names as per current queue lengths, updating of ordered list of buffer names as per the required processing rates, etc.

<u>*Messages that cause setting and getting of information and instance variables:*</u> These messages set and get the values of instance variables or information. For example setting

of the controlled subject or accessing it, accessing of current Small Time Interval (STI), etc.

Messages that select a lot based on some rule: Some examples are **selectALotWithLongestWaiting, selectALotWithLeastSlack, selectALotWithEDD,** etc.

Messages that mimic the Interrogator: When the WorkCenterController approaches the WCBrain for getting supervisory decisions pertaining to the dispatching rule to be used for selection of lots for processing, the WCBrain queries the WorkCenterController about various parameters. For answering these questions posed by the knowledge base, certain methods have to be implemented in the class WorkCenterController. All these messages are grouped in the category Interrogator Mimicry.

WorkStationController: This class has mainly the methods that are used in setting and getting the instance variables such as bigTimePeriod, current value of Small Time Interval (STI), superController, etc. Further it has the following messages which it uses for communication with the super controller:

allotMeALot, **finishedAWFI: aWFI**, and **whatShouldBeNextLotToBeProcessed.**

It also understands the message **processALot: aLot**. This message will be sent to the WorkStationController by the WorkCenterController. Upon receipt of this message the WorkStationController sends the same message to the WorkStation (that is being controlled) for processing this lot.

CHAPTER 7

EXPERIMENTAL DETAILS

7.1 Introduction

This chapter discusses the performance of the control architecture in the context of several complexities of the RFS. The performance is studied for both types of communication strategies, i.e. one-way and two-way, and for several different Time Period of Periodic Actions (TPOFPA). Also investigated is whether the objective function in the LP for arriving at the loading rates and the part-operation rates plays any role in effective control of RFS.

7.2 Classification of Complexities for the RFS

The RFS complexities can be grouped into two classes; structural complexities and complexities due to randomness as shown in Table 1. Each class contains further sub-classes of complexities.

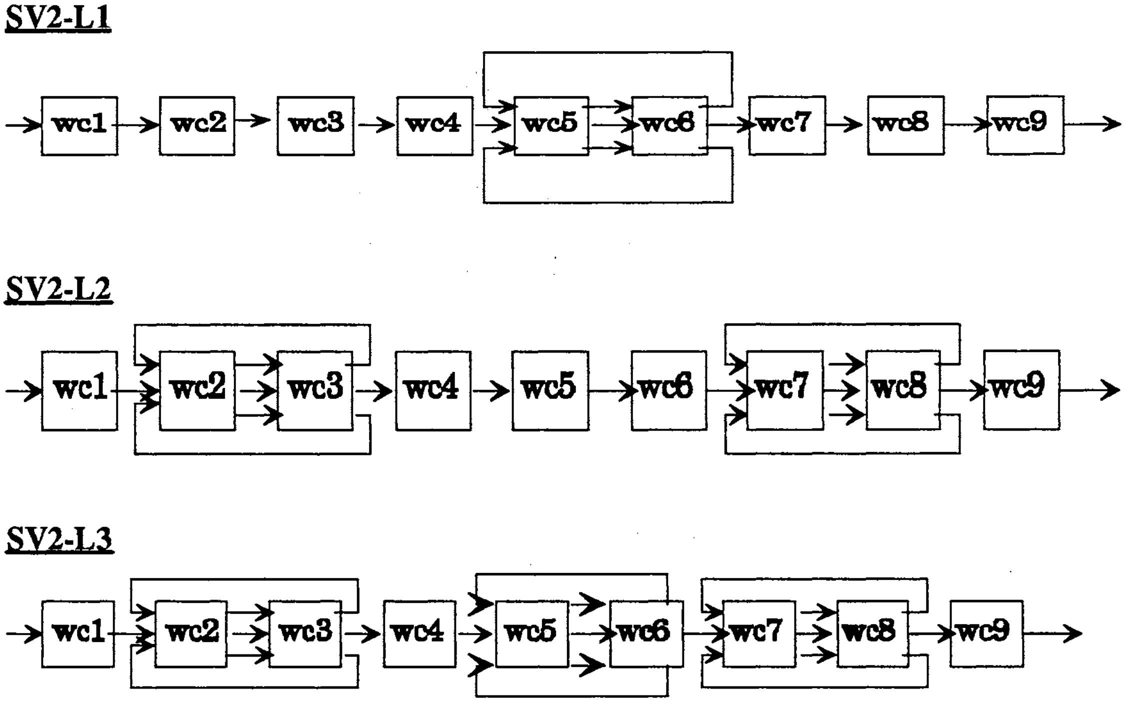
Table 1. Types of Complexities

Structural complexities	**Complexities due to randomness**
Number of places where re-entrancy occurs	Due to variability in availability of machines
Span of re-entrancy	Due to variability in processing time
Number of re-entrant paths	

Table 1 lists only those complexities which are studied in this research. It does not include for example, complexity due to yield effects, i.e.scrapping of processed parts. The above complexities make the task of control of the RFS more and more difficult as the level of complexity is increased. Each of these complexities are diagrammed in the following subsection.

7.2.1 Structural Complexities and Complexities Due to Randomness

The complexity of RFS increases with randomness that can occur due to failure of work stations (referred to as SV5), due to variablity in processing times (referred to as SV6) and due to variable yield (not studied in this research). This chapter compares the performance of the control architecture with both strategies of communication. TPOFPA (Time Period of Periodic Actions) has been varied to see how the performance of the controllers varied for both coupling strategies. The details of various experiments are given in the subsequent sections. The structural complexities are depicted in Figures 5, 6 and 7.


SV2-L1

SV2-L2

SV2-L3

Figure 5. Complexity Due to Number of Places where Re-entrancy Occurs (SV2)

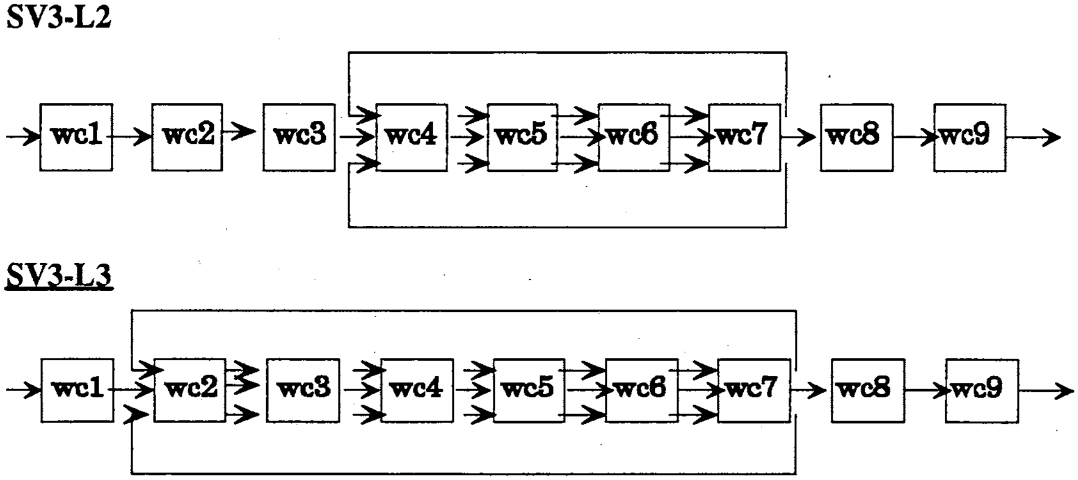


Figure 6. Complexity Due to Span of Re-entrancy (SV3)

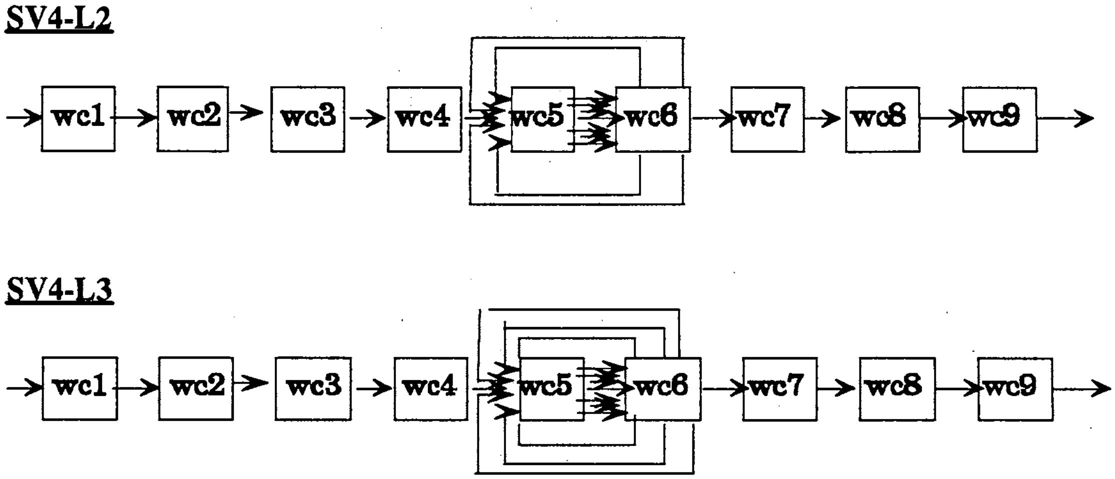


Figure 7. Complexity Due to Number of Re-entrant Paths (SV4)

7.3 Measures of Merit

Before delving into the experimental set up and discussion of results, the measures of merit that were used in assessing the performance of the two communication strategies are listed. The following measures of merit were chosen:

Flow Time (FT): This is the time spent by the part in the RFS, from the instance it is released into the RFS up to the instance it is finished and delivered to the finished goods buffer, at the output of the RFS. There are two dimensions to this measure, the mean flow time and standard deviation of the flow time.

Percentage Production Not Made (PPNM): At the end of every week production targets have to be satisfied. Thus suppose the plant was required to produce 75, 50, 75 and 100 units at the end of the first, second, third and fourth weeks for product 1, and for product 2 these figures are, say, 50 at the end of each week. Further suppose for product 1, the actual production was 60, 50, 70 and 90 while for product 2, it was 40, 50, 45, and 50. Then the percentage production not made is calculated as (45/500)* 100% = 9%.

Number In System (NIS): This is counted as number of parts in the RFS (i.e. physical count) which includes parts in all the buffers in front of the work centers and the parts being processed on all the work stations. Both the mean and the standard deviation of number in system are important.

7.4 Experimentation for Study in the Context of Structural Complexities

In all the experiments described in this section, the RFS produces two products and the production targets for four weeks are given in Table 2.

Table 2. Weekly Production Targets

Product	Week 1	Week 2	Week 3	Week 4
Product 1	30	120	50	60
Product 2	120	100	100	110

Table 3 presents different treatment combinations for study of the performance of the control architecture in the context of structural complexities. In this table SV2 represents the complexity due to number of places where re-entrancy occurs, SV3, the complexity due to the span of re-entrancy, and SV4, the complexity due to the number of re-entrant paths. For each cell in the table one simulation run (which is one experiment) is carried out as the complexity factors due to randomness are not introduced at this stage. Each simulation starts with no parts in the RFS and ends at the end of big time period T, i.e. each simulation is a terminating simulation. The experiments can be performed in any order at random without any effect on the results obtained. Thus the experimental design is that of completely randomized design (CRD) with factorial arrangement of several types of factors.

Table 3. Factor Combinations for Study in the Context of Structural Complexities

Time Period Of Periodic Actions	20 min.		80 min.		160 min.		320 min.		480 min.	
Communication Strategy	OW	TW	OW	TW	OW	TW	OW	TW	OW	TW
SV2-L1										
SV2-L2										
SV2-L3										
SV3-L1										
SV3-L2										
SV3-L3										
SV4-L1										
SV4-L2										
SV4-L3										

The following subsections describe the details of each experiment and the results.

7.4.1 Number of Places Where Re-entrancy Occurs - Level 1 (SV2-L1)

Here the re-entrancy occurs only at one place in the RFS. Table 4 presents the routing details along with the processing times. All the processing times are deterministic. Table 5 presents the experimental results.

Table 4. Routing Details (SV2-L1)

Operation No.	1	2	3	4	5	6	7	8	9	10	11	12	13
Work Center No.	1	2	3	4	5	6	5	6	5	6	7	8	9
Operation Time Prod1	30	30	30	30	5	5	10	10	15	15	30	30	30
Operation Time Prod2	20	20	20	20	5	5	5	5	10	10	20	20	20

Table 5. Results (SV2-L1)

TPOFPA & Strategy of Communication	Flow Time Mean	Flow Time SSD	Number in System Mean	Number in System SSD	% Lost Demand
Two-Way (20)	581.6	262.2	42.35	22.9	4.49
One-Way (20)	592.8	269.8	43.2	23.1	4.35
Two-Way (80)	599.9	296.9	43.34	22.4	6.38
One-Way (80)	650.11	311.9	47.7	27.00	5.65
Two-Way (160)	591.7	282.4	42.58	20.3	7.39
One-Way (160)	638.3	300.1	46.14	24.1	7.39
Two-Way (320)	604.6	286.3	43.4	20.2	6.96
One-Way (320)	594.68	247.5	43.9	21.15	8.55
Two-Way (480)	624.6	316	45.52	19.73	8.55
One-Way (480)	627.23	272.39	45.32	23.0	10.58

Figure 8 presents the results graphically.

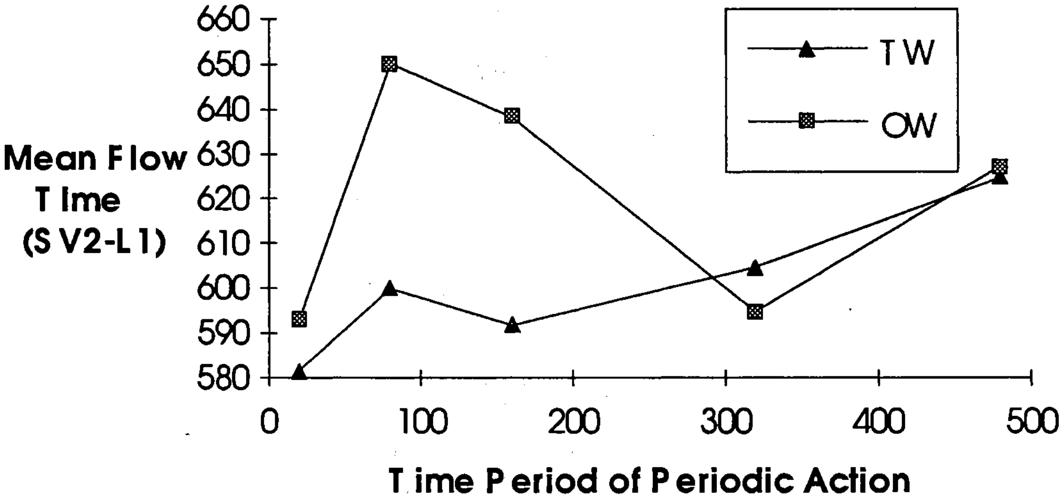


Figure 8 (a)

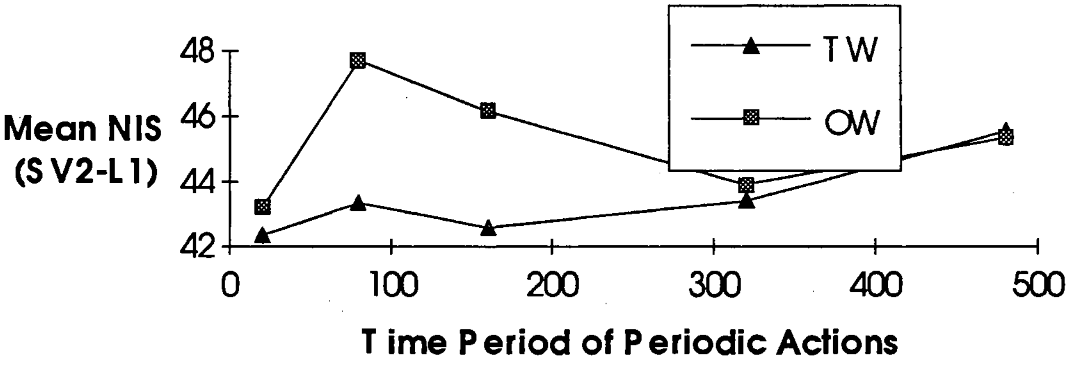


Figure 8 (b)

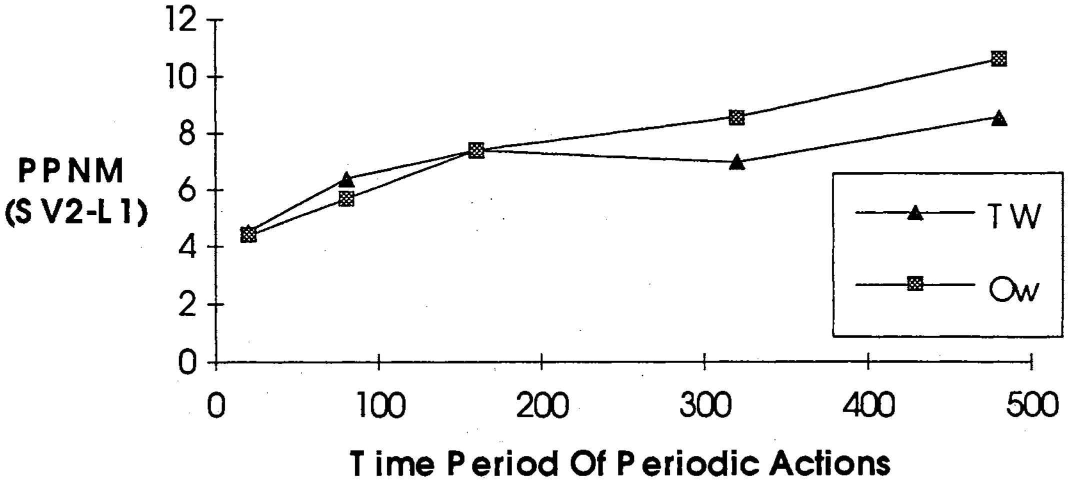


Figure 8 (c)

Figure 8. RFS Performance (SV2-L1)

7.4.2 Number of Places Where Re-entrancy Occurs - Level 2 (SV2-L2)

Table 6. Routing Details (SV2-L2)

Operation No.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
Work Center No.	1	2	3	2	3	2	3	4	5	6	5	6	5	6	7	8	9
Operation Time Prod1	30	5	5	10	10	15	15	30	5	5	10	10	15	15	30	30	30
Operation Time Prod2	20	5	5	5	5	10	10	20	5	5	5	5	10	10	20	20	20

Table 7. Results (SV2-L2)

TPOFPA & Strategy of Communication	Flow Time Mean	Flow Time SSD	Number in System Mean	Number in System SSD	% Lost Demand
Two-Way (20)	611.1	279.2	44.4	23.3	4.49
One-Way (20)	661.2	324.1	48.4	27.9	4.93
Two-Way (80)	624.56	304.9	45.1	22.8	6.52
One-Way (80)	687.4	341.7	51.1	30.7	5.94
Two-Way (160)	644.2	304.9	46.3	22.8	7.1
One-Way (160)	685.8	326.9	49.8	26.6	7.68
Two-Way (320)	682.2	328.6	49.0	24.2	7.25
One-Way (3200	638.2	300.8	47.1	22.5	9.6
Two-Way (480)	670.0	344.5	50.0	22.2	9.27
One-Way (480)	700.9	299.5	51.75	25.8	12.6

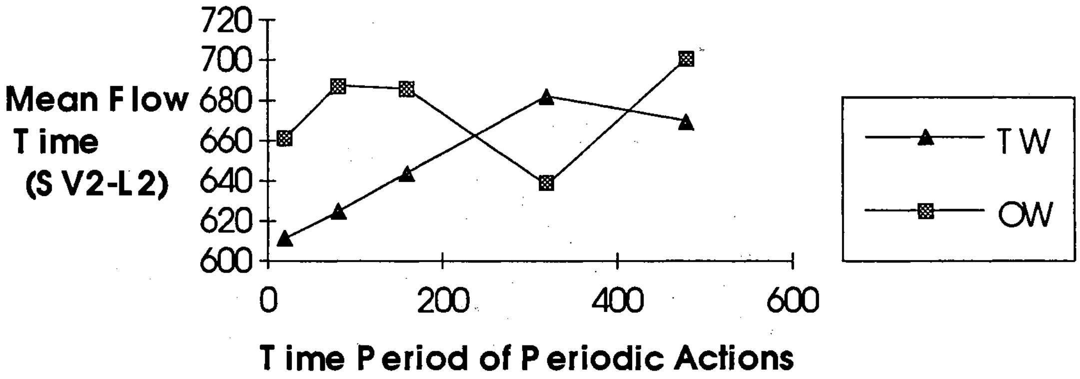


Figure 9 (a)

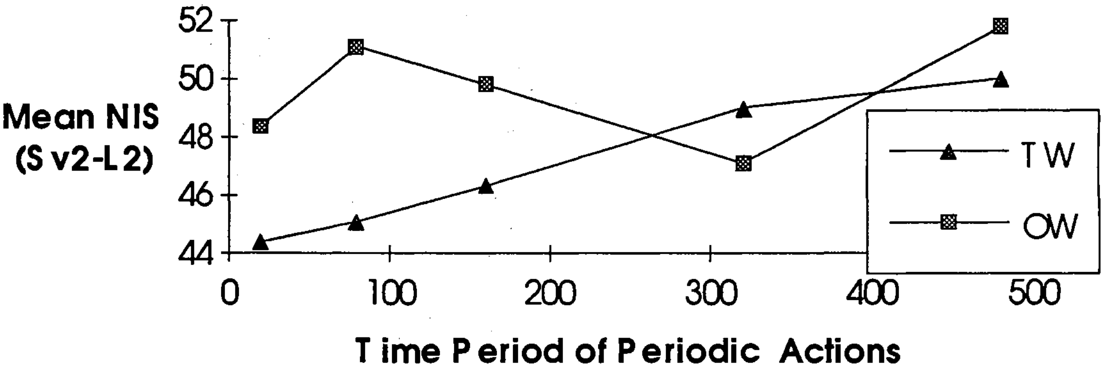


Figure 9 (b)

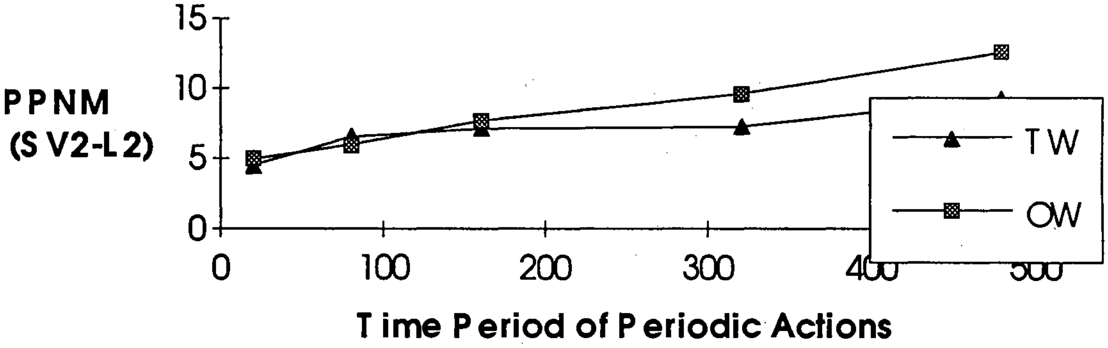


Figure 9 (c)

Figure 9. RFS Performance (SV2-L2)

7.4.3 Number of Places Where Re-entrancy Occurs - Level 3 (SV2-L3)

Table 8. Routing Details (SV2-L3)

Operation No.	1	2	3	4	5	6	7	8	9	10	11
Work Center No.	1	2	3	2	3	2	3	4	5	6	5
Operation Time Prod1	30	5	5	10	10	15	15	30	5	5	10
Operation Time Prod2	20	5	5	5	5	10	10	20	5	5	5

Operation No.	12	13	14	15	16	17	18	19	20	21
Work Center No.	6	5	6	7	8	7	8	7	8	9
Operation Time Prod1	10	15	15	5	5	10	10	15	15	30
Operation Time Prod2	5	10	10	5	5	5	5	10	10	20

Table 9. Results (SV2-L3)

TPOFPA & Strategy of Communication	Flow Time Mean	Flow Time SSD	Number in System Mean	Number in System SSD	% Lost Demand
Two-Way (20)	640.34	294.4	46.4	24.4	4.93
One-Way (20)	696.0	344.64	51.5	31.0	4.8
Two-Way (80)	657.3	304.6	47.5	22.8	5.94
One-Way (80)	712.3	348.8	52.9	31.7	6.23
Two-Way (160)	671.9	316.1	48.6	22.41	6.81
One-Way (160)	731.1	342.5	53.53	29.43	8.41
Two-Way (320)	714.6	333.5	51.4	25.4	8.12
One-Way (320)	717.9	327.6	53.1	28.8	12.03
Two-Way (480)	744.8	379.6	55.3	26.5	9.56
One-Way (480)	775.1	341.9	57.6	30.1	13.04

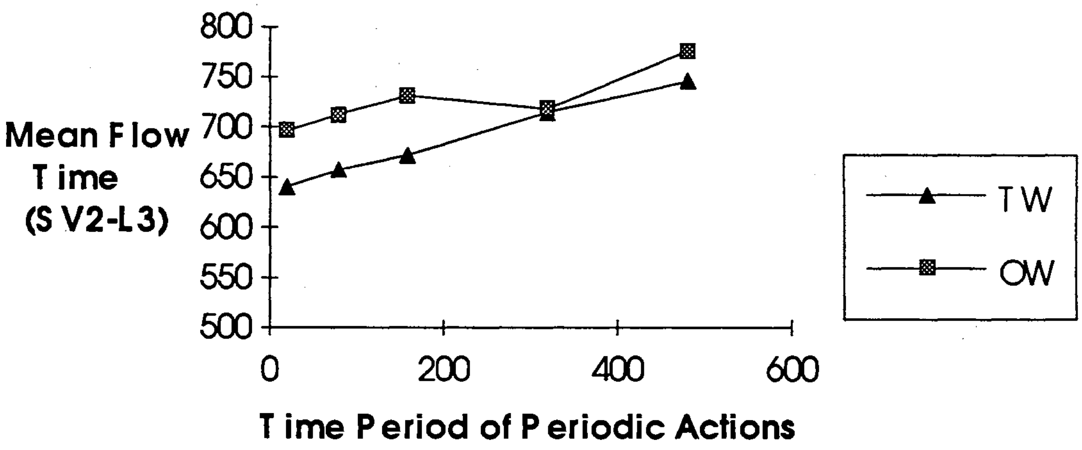


Figure 10 (a)

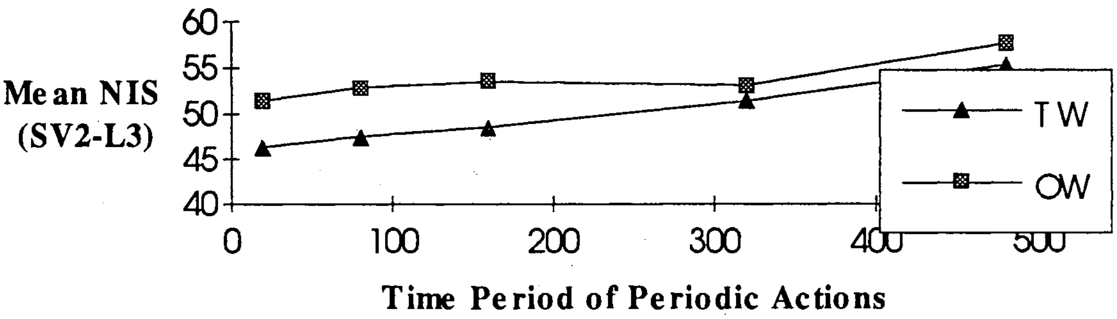


Figure 10 (b)

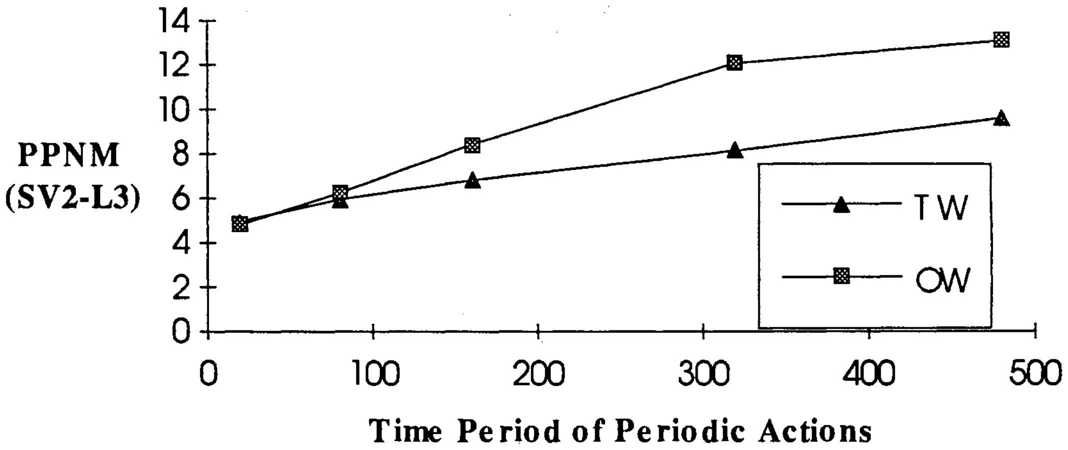


Figure 10 (c)

Figure 10. RFS Performance (SV2-L3)

In the subsequent subsections graphs are not shown as the patterns are generally similar to those shown earlier.

7.4.4 Span of Re-Entrancy Complexity - Level 2 (SV3-L2)

Since SV3-L1 is same as SV2-L1, only SV3-L2 and L3 are shown in the following.

Table 10. Routing Details (SV3-L2)

Operation No.	1	2	3	4	5	6	7	8	9
Work Center No.	1	2	3	4	5	6	7	4	5
Operation Time Prod1	30	30	30	5	5	5	5	10	10
Operation Time Prod2	20	20	20	5	5	5	5	5	5

Operation No.	10	11	12	13	14	15	16	17
Work Center No.	6	7	4	5	6	7	8	9
Operation Time Prod1	10	10	15	15	15	15	30	30
Operation Time Prod2	5	5	10	10	10	10	20	20

Table 11. Results (SV3-L2)

TPOFPA & Strategy of Communication	Flow Time Mean	Flow Time SSD	Number in System Mean	Number in System SSD	% Lost Demand
Two-Way (20)	617.7	279.5	45.1	24.2	5.07
One-Way (20)	635.6	279.8	45.4	24.4	4.64
Two-Way (80)	621.2	302.5	45.9	23.6	7.10
One-Way (80)	676.7	335.8	49.1	26.8	6.38
Two-Way (160)	625.7	286.5	44.9	21.5	7.97
One-Way (160)	649.9	304.3	46.9	23.7	8.12
Two-Way (320)	620.8	279.2	44.4	19.7	8.41
One-Way (320)	638.9	294.6	46.2	23.3	11.45
Two-Way (480)	665.6	271.2	47.2	18.9	8.7
One-Way (480)	653.6	277.3	47.1	22.8	11.6

7.4.5 Span of Re-entrancy Complexity Level 3 (SV3-L3)

Table 12. Routing Details (SV3-L3)

Operation No.	1	2	3	4	5	6	7	8	9	10
Work Center No.	1	2	3	4	5	6	7	2	3	4
Operation Time Prod1	5	5	5	5	5	5	10	10	10	10
Operation Time Prod2	5	5	5	5	5	5	5	5	5	5

Operation No.	11	12	13	14	15	16	17	18	19	20	21
Work Center No.	5	6	7	2	3	4	5	6	7	8	9
Operation Time Prod1	10	10	10	15	15	15	15	15	15	30	30
Operation Time Prod2	5	5	5	10	10	10	10	10	10	20	20

Table 13. Results (SV3-L3)

TPOFPA & Strategy of Communication	Flow Time Mean	Flow Time SSD	Number in System Mean	Number in System SSD	% Lost Demand
Two-Way (20)	633.5	275.1	45.9	23.9	5.79
One-Way (20)	637.9	278.8	46.3	24.1	5.36
Two-Way (80)	653.3	303.4	47.0	23.4	7.39
One-Way (80)	694.5	303.9	50.2	26.9	6.52
Two-Way (160)	642.2	288.0	46.0	20.9	8.84
One-Way (160)	695.4	315.5	50.1	25.1	8.55
Two-Way (320)	631.3	289.8	45.1	19.3	8.41
One-Way (320)	647.5	296.1	46.8	22.8	12.32
Two-Way (480)	672.2	273.1	47.6	18.6	8.7
One-Way (480)	670.9	284.7	47.9	22.2	12.75

7.4.6 Number of Re-Entrant Paths Complexity Level 2 (SV4-L2):

Since SV4-L1 is same as SV2-L1 and SV3-L1 only SV4 L2 and L3 are presented.

Table 14. Routing Details (SV4-L2)

Operation No.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
Work Center No.	1	2	3	4	5	6	5	6	5	6	5	6	5	6	7	8	9
Operation Time Prod1	30	30	30	30	4	4	5	5	6	6	5	5	10	10	30	30	30
Operation Time Prod2	20	20	20	20	2	2	3	3	2	2	3	3	10	10	20	20	20

Table 15. Results (SV4-L2):

TPOFPA & Strategy of Communication	Flow Time Mean	Flow Time SSD	Number in System Mean	Number in System SSD	% Lost Demand
Two-Way (20)	581.2	269.8	42.2	22.9	4.20
One-Way (20)	586.4	273.1	42.8	23.2	4.20
Two-Way (80)	598.6	293.2	43.2	22.5	5.94
One-Way (80)	642.9	335.5	47.3	27.1	5.51
Two-Way (160)	591.2	279.9	42.5	20.5	7.54
One-Way (160)	635.9	299.1	45.9	23.9	7.25
Two-Way (320)	604.5	284.6	43.3	20.3	6.81
One-Way (320)	608.5	249.6	43.7	22.4	10.43
Two-Way (480)	614.6	319.8	45.5	19.9	8.11
One-Way (480)	621.4	271.1	44.9	23.0	10.14

7.4.7 Number of Re-Entrant paths complexity Level 3 (SV4-L3):

Table 16. Routing Details (SV4-L3)

Operation No.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
Work Center No.	1	2	3	4	5	6	5	6	5	6	5	6	5	6	5	6	5	6	7	8	9
Operation Time Prod1	30	30	30	30	4	4	5	5	6	6	5	5	2	2	3	3	5	5	30	30	30
Operation Time Prod2	20	20	20	20	2	2	3	3	2	2	3	3	2	2	3	3	5	5	20	20	20

Table 17. Results (SV4-L3)

TPOFPA & Strategy of Communication	Flow Time Mean	Flow Time SSD	Number in System Mean	Number in System SSD	% Lost Demand
Two-Way (20)	588.1	265.9	42.7	22.7	4.93
One-Way (20)	592.7	271.1	43.2	23.1	4.78
Two-Way (80)	599.9	291.8	43.3	22.5	6.23
One-Way (80)	657.9	338.9	47.8	26.9	5.8
Two-Way (160)	601.3	288.1	43.2	20.7	7.10
One-Way (160)	643.1	302.3	46.4	23.9	7.39
Two-Way (320)	607.6	287.6	43.6	19.9	7.25
One-Way (320)	599.5	248.7	44.3	21.3	8.99
Two-Way (480)	654.5	276.4	46.4	19.4	8.41
One-Way (480)	634.4	272.5	45.8	22.9	10.72

Based on the experimentation, the following observations are made:

(1) Across all the different RFS structures that were tested, the control architecture with two-way communication strategy was at least as good as the one-way communication strategy and in most cases was better.

(2) The performance of both strategies is comparable when the Time Period Of Periodic Actions (TPOFPA) is small.

(3) The performance of both control architectures deteriorates as the TPOFPA is increased. However the deterioration is less in case of two-way communication and much quicker in case of the control architecture employing one-way communication.

(4) The gap in the performance between the two alternative architecture widens as the TPOFPA increases.

(5) The one-way communication architecture results in higher mean flow time for the parts as compared to the one with two-way communication. This is true for the cases where the percentage of production not made (PPNM) is comparable in both cases.

(6) If the mean flow time for the one-way communication architecture is lower for a given TPOFPA as compared to that for two-way communication, then it is only at the expense of a significant deterioration in the percentage of production not made.

(7) The standard deviation of flow time is higher in case of the one-way communication strategy.

7.5 Experimentation in the Context of Process Time Variability (Factor SV6)

Throughout this stage of experimentation, the RFS structure was fixed. The structure is shown in Figure 11. Number of work stations in each work center = 2. Production targets for each product are shown in the Table 18.

Table 18. Weekly Production Targets

day --->	5	10	15	20
Prod 1	50	75	100	75
Prod 2	50	50	50	50

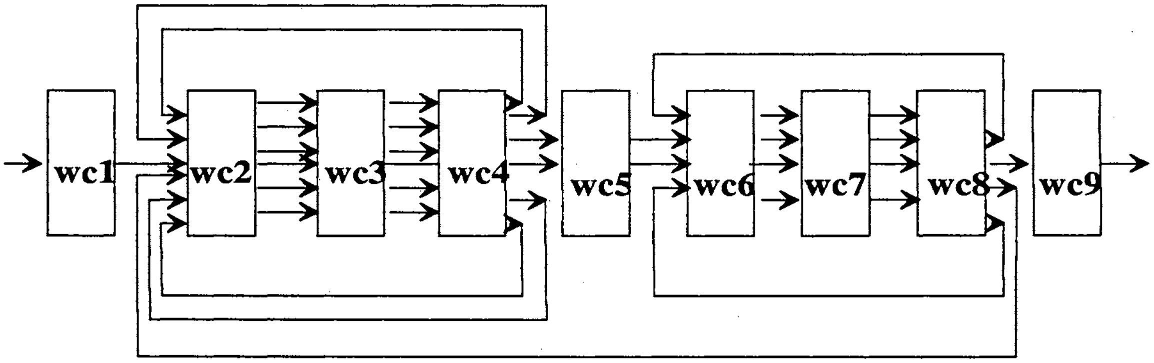


Figure 11. RFS Structure For Studying Effects of Randomness

Table 19. Routing Details for Studying Effects of Randomness

Operation No..	1	2	3	4	5	6	7	8	9	10	11	12
Work Center No.	1	2	3	4	2	3	4	2	3	4	2	3
Product 1 Time	30	5	5	5	5	5	5	5	5	5	5	5
Product 2 Time	20	3	3	3	3	3	3	3	3	3	3	3

Operation No.	13	14	15	16	17	18	19	20	21	22
Work Center No.	4	2	3	4	5	6	7	8	6	7
Product 1 Time	5	5	5	5	15	10	10	10	10	10
Product 2 Time	3	3	3	3	10	5	5	5	5	5

Operation No.	23	24	25	26	27	28	29	30	31	32	33	34
Work Center No.	8	6	7	8	2	3	4	5	6	7	8	9
Product 1 Time	10	5	5	5	5	5	5	15	5	5	5	30
Product 2 Time	5	5	5	5	5	5	5	10	5	5	5	20

All process times are mean values of uniform distribution with the range of the distribution determined by the level of the factor SV6. Different levels of this factor are;

Level 1 (Low): Half Range = 10 % of mean. Thus if mean process time is say 30.0, then the uniform distribution will range from 27.0 to 33.0.

Level 2 (Medium): = 40 % of mean, and

Level 3 (High): = 70 % of mean.

Table 20 presents the different treatment combinations in the experimental setup. As shown therein, three levels of the factor SV6, five levels of the factor TPOFPA (Time Period Of Periodic Actions) and two levels of the factor communication strategy are involved. Each cell represents one combination of these factors. In each cell five replications are carried out. One replication is one simulation run. Each simulation is of terminating type ending after big time period T. There is no warmup period in each simulation. Further, each simulation starts with no parts in the RFS. Each simulation is carried out on the experimental unit. The experimental unit is the RFS with its fixed structure, and fixed weekly production targets. Further, every time the simulation run is made the seed for the run is generated randomly. It is not necessary to carry out all 5 replications in one cell and then progress to the other cell. In other words, one replication in any cell randomly picked can be carried out, then another replication in another randomly picked cell can be carried out and so on. The order in which the runs are made has no impact on the results. Thus the experimental design is Completely Randomized Design with factorial arrangements of the treatments.

Table 20. Treatment Combinations for Study in the Context of Process Time Variability

Time Period of Periodic Action	20.0		80.0		160.0		320.0		480.0	
Communication Strategy	TW	OW	TW	OW	TW	OW	TW	OW	TW	OW
SV6 - Low										
SV6 - Medium										
SV6 - High										

Statistics were collected on Flow Time mean and SSD, Percentage Production Not Made, Number In System Mean (NISM) and SSD. ANOVA was carried out. The SAS program is given in Gharpure [1994]. The model is;

Dependent Variable = f(COMSTGY, OPTVAR, TPOFPA, COMSTGY * OPTVAR, COMSTGY * TPOFPA, OPTVAR * TPOFPA, COMSTGY * OPTVAR * TPOFPA)

where "dependent variable" is one of the measures of merit, viz. Flow Time Mean (FLOWM), Flow Time Sample Standard Deviation (FLOWSD), Percentage Production Not Made (PPNM), and Number In System Mean (NISM).

Table 21. ANOVA for Flow Time Mean (FLOWM)

Source	**DF**	**SS**	**MS**	**F**	**Pr > F**
Model	29	1612044.03	55587.73	41.45	0.0001
COMSTGY	1	663239.88	663239.88	494.51	0.0001
OPTVAR	2	387777.86	193888.93	144.56	0.0001
TPOFPA	4	313906.43	78476.61	58.51	0.0001
COMSTGY*OPTVAR	2	29220.55	14610.28	10.89	0.0001
COMSTGY*TPOFPA	4	192892.47	48223.12	35.95	0.0001
OPTVAR*TPOFPA	8	15573.17	1946.65	1.45	0.1823
COMSTGY*OPTVAR* TPOFPA	8	9433.66	1179.21	0.88	0.5362
Error	120	160945.20	1341.21		
Corrected Total	149	1772989.22			

The following observations can be made:

- The model accounts for a large proportion of the variability.
- The interactions between COMSTGY*OPTVAR are significant. So also between COMSTGY*TPOFPA.
- The interaction between OPTVAR*TPOFPA does not seem to be dominant. So also among all three.
- The interaction among COMSTGY*TPOFPA is stronger than between COMSTGY*OPTVAR.
- DUNCAN's multiple range test for the above showed that flow time resulting in one-way strategy is significantly higher than in two-way (1015.6 vs. 882.6).

Since COMSTGY occurs in both interactions, the data was sorted by COMSTGY and ANOVA was carried out to study the interactions.

Table 22. ANOVA for Flow Time Mean with COMSTGY = One-Way

Source	**DF**	**SS**	**MS**	**F**	**Pr > F**
Model	14	783413.35	55958.1	41.72	0.0001
OPTVAR	2	314810.95	157405.47	117.36	0.0001
TPOFPA	4	459072.72	114768.18	85.57	0.0001
OPTVAR*TPOFPA	8	9529.67	1191.21	0.89	0.5289
Error	120	160945.19	1341.21		

Table 23. ANOVA for Flow Time Mean with COMSTGY = Two-Way.

Source	**DF**	**SS**	**MS**	**F**	**Pr > F**
Model	14	165390.8	11813.63	8.81	0.0001
OPTVAR	2	102187.47	51093.73	38.09	0.0001
TPOFPA	4	47726.17	11931.54	8.9	0.0001
OPTVAR*TPOFPA	8	15477.16	1934.64	1.44	0.1868
Error	120	160945.20	1341.21		

Duncan's Multiple Range Test for mean flow time was carried out with operation time variability (OPTVAR) as the independent variable. The results are in the following table. In this table DG means Duncan Grouping.

Table 24. Duncan's Test for Flow Time Mean with Independent Variable = OPTVAR

ONE-WAY			**TWO-WAY**			**DIFFERENCE**
DG	**FLOWM**	**OPTVAR**	**DG**	**FLOWM**	**OPTVAR**	
A	1101.14	high	A	930.76	high	170.38*
B	1001.25	Medium	B	875.98	Medium	125.27*
C	944.405	Low	C	841.08	Low	103.325*

The following comments can be made;

- The same Duncan grouping letter in a given column for a specific communication strategy indicates no significant difference among means.
- Differences in mean flow time when averaged over all levels of TPOFPA (Time Period of Periodic Actions), are not the same for the three levels of Operation Time Variability within each of the communication strategy.
- A * indicates that the means in that row are significantly different.

- The mean flow time for one-way is higher than that achieved for two-way at a given level of OPTVAR.
- For both one-way and two-way, the mean flow time increases as OPTVAR increases, but the increase is faster in one-way than in two-way (@ 16 .5% vs. @10.2% from low to high).

Duncan's multiple range test for mean flow time was carried out with Time Period of Periodic Actions (TPOFPA) as the independent variable. The results are in the following table. In this table DG stands for Duncan Grouping.

Table 25. Duncan's Test for Flow Time Mean with Independent Variable = TPOFPA

ONE-WAY			TWO-WAY			DIFFERENCE
DG	**FLOWM**	**TPOFPA**	**DG**	**FLOWM**	**TPOFPA**	
A	1165.95	480.0	A	903.37	480.0	262.58*
B	1002.4	320	A	897.38	160.0	105.02*
B	986.76	80	A	897.33	320.0	89.43*
B	985.203	160	A	880.44	20	104.76*
C	937.69	20	B	834.53	80	103.16*

The following comments can be made:

- Differences in mean flow time when averaged over all levels of OPTVAR (Operation Time Variability), between one-way and two-way strategies are not the same for the five levels of Time Period of Periodic Actions.
- The mean flow time for one-way is higher than that achieved for two-way at a given level of TPOFPA.
- For both one-way and two-way, the mean flow time increases as TPOFPA increases, but the increase is faster in one-way than in two-way.

Similar analysis was done for sample standard deviation of flow time, percentage production not made (PPNM) and mean Number In System (NISM) (Tables 26 to 40). In every case it was found that;

- The model accounts for the variability to a large extent.

- The interactions between COMSTGY*OPTVAR are significant. So also between COMSTGY*TPOFPA.
- The interaction between OPTVAR*TPOFPA does not seem to be dominant. So also among all three.
- The interaction among COMSTGY*TPOFPA is stronger than between COMSTGY*OPTVAR.
- DUNCAN's multiple range test showed that the dependent variable of concern in one-way strategy is significantly higher than in two-way.
- Differences in the dependent variable of concern when averaged over all levels of OPTVAR (Operation Time Variability), between one-way and two-way strategies are not the same for the five levels of Time Period of Periodic Actions.
- The dependent variable of concern for one-way is higher than that achieved for two-way at a given level of TPOFPA.
- For both one-way and two-way, the dependent variable of concern increases as TPOFPA increases, but the increase is faster in one-way than in two-way.

In above the "dependent variable of concern" refers to one of the measures of merit.

The following sub-sections present the results for the remaining measures of merit.

7.5.1 Flow Time-Sample Standard Deviation (FLOWSD)

Table 26. ANOVA for Flow Time Sample Standard Deviation (FLOWSD)

Source	**DF**	**SS**	**MS**	**F**	**Pr > F**
COMSTGY	1	300587.04	300587.04	493.90	0.0001
OPTVAR	2	7432.31	3716.15	6.11	0.003
TPOFPA	4	16699.04	4174.76	6.86	0.0001
COMSTGY*OPTVAR	2	17414.58	8707.29	14.31	0.0001
COMSTGY*TPOFPA	4	81413.88	20353.47	33.44	0.0001
OPTVAR*TPOFPA	8	17420.10	2177.51	3.58	0.0009
COMSTGY*OPTVAR* TPOFPA	8	3415.26	426.91	0.7	0.6897
Error	120	73031.79	608.70		

Since COMSTGY occurs in both interactions, the data was sorted by COMSTGY and ANOVA was carried out to study interactions.

Table 27. ANOVA for FLOWSD with COMSTGY = One-Way

Source	DF	SS	MS	F	Pr > F
Model	14	105248.53	7517.75	12.35	0.0001
OPTVAR	2	20223.91	10111.96	16.61	0.0001
TPOFPA	4	73939.65	18484.91	30.37	0.0001
OPTVAR*TPOFPA	8	11084.96	1385.62	2.27	0.0269
Error	120	73031.79	608.70		

Table 28. ANOVA for FLOWSD with COMSTGY = Two-Way

Source	DF	SS	MS	F	Pr > F
Model	14	38546.65	2753.33	4.52	0.0003
OPTVAR	2	4622.97	2311.49	3.79	0.025
TPOFPA	4	24173.27	6043.32	9.93	0.0001
OPTVAR*TPOFPA	8	9750.40	1218.80	2.00	0.052
Error	120	73031.79	608.70		

Duncan's multiple range test for FLOWSD was carried out with operation time variability (OPTVAR) as the independent variable. The results are in Table 29.

Table 29. Duncan's Test for FLOWSD with Independent Variable = OPTVAR

ONE-WAY			TWO-WAY			DIFFERENCE
DG	FLOWSD	OPTVAR	DG	FLOWSD	OPTVAR	
A	364.936	High	A	263.47	Low	101.47*
B	334.634	Low	A\B	249.22	Medium	85.41*
B	326.878	Medium	B	245.16	High	81.72*

Table 30. Duncan's Test for FLOWSD with Independent Variable = TPOFPA

ONE-WAY			TWO-WAY			DIFFERENCE
DG	FLOWSD	TPOFPA	DG	FLOWSD	TPOFPA	
A	391.953	480.0	A	277.39	20	114.56*
B	358.535	80	A	264.18	160	94.36*
C	332.6	320	A	257.19	320	75.41*
C	329.4	20	B	234.06	480	95.34*
D	298.255	160	B	230.28	80	67.98*

7.5.2 Percentage Production Not Made (PPNM)

Table 31. ANOVA for Percentage Production Not Made (PPNM)

Source	DF	SS	MS	F	Pr > F
Model	29	2046.49	70.57	29.88	0.0001
COMSTGY	1	466.049	466.05	197.34	0.0001
OPTVAR	2	391.96	195.98	82.99	0.0001
TPOFPA	4	858.39	214.60	90.87	0.0001
COMSTGY*OPTVAR	2	16.32	8.16	3.46	0.0347
COMSTGY*TPOFPA	4	275.53	68.88	29.17	0.0001
OPTVAR*TPOFPA	8	14.87	1.86	0.79	0.615
COMSTGY*OPTVAR* TPOFPA	8	23.37	2.92	1.24	0.2836
Error	120	283.39	2.36		
Corrected Total	149	2329.88			

Since COMSTGY occurs in both interactions, the data was sorted by COMSTGY and ANOVA was carried out to study interactions. ANOVA for PPNM as the dependent variable and for COMSTGY = One-Way is given below.

Table 32. ANOVA for PPNM with COMSTGY = One-Way

Source	DF	SS	MS	F	Pr > F
Model	14	1329.00	94.93	40.19	0.0001
OPTVAR	2	278.80	139.40	59.17	0.0001
TPOFPA	4	1039.81	259.95	110.06	0.0001
OPTVAR*TPOFPA	8	10.40	1.30	0.55	0.0.8166
Error	120	283.39	2.36		

Table 33. ANOVA for PPNM for COMSTGY = Two-Way

Source	DF	SS	MS	F	Pr > F
Model	14	251.44	17.96	7.60	0.0001
OPTVAR	2	129.48	64.74	27.41	0.0001
TPOFPA	4	94.12	23.53	9.96	0.0001
OPTVAR*TPOFPA	8	27.83	3.48	1.47	0.1752
Error	120	283.39	2.36		

Duncan's multiple range test for PPNM was carried out with operation time variability (OPTVAR) as the independent variable. The results are in Table 34.

Table 34. Duncan's Test for PPNM with Independent Variable = OPTVAR

ONE-WAY			TWO-WAY			DIFFERENCE
DG	PPNM	OPTVAR	DG	PPNM	OPTVAR	
A	13.67	high	A	9.22	high	4.45*
B	10.00	Medium	B	7.06	Medium	2.94*
B	9.26	Low	C	6.08	Low	3.18*

Duncan's multiple range test for PPNM was carried out with Time Period of Periodic Actions (TPOFPA) as the independent variable. The results are in Table 35. In this table DG means Duncan Grouping.

Table 35. Duncan's Test for PPNM with Independent Variable = TPOFPA

ONE-WAY			TWO-WAY			DIFFERENCE
DG	PPNM	TPOFPA	DG	PPNM	TPOFPA	
A	16.29	480.0	A	8.76	320	7.53*
B	13.40	320	A	8.39	480	5.01*
C	11.55	160	A	7.83	160	3.72*
D	7.28	80	B	6.41	80	0.87
D	6.37	20	B	5.88	20	0.49

7.5.3 Number In System Mean (NISM)

Table 36. ANOVA for Number In System Mean (NISM)

Source	**DF**	**SS**	**MS**	**F**	**Pr > F**
Model	29	3519.2	121.35	40.26	0.0001
COMSTGY	1	1687.87	1687.87	559.99	0.0001
OPTVAR	2	642.90	321.45	106.65	0.0001
TPOFPA	4	579.26	144.82	48.05	0.0001
COMSTGY*OPTVAR	2	73.98	36.99	12.27	0.0001
COMSTGY*TPOFPA	4	444.88	111.22	36.9	0.0001
OPTVAR*TPOFPA	8	64.34	8.04	2.67	0.0099
COMSTGY*OPTVAR* TPOFPA	8	25.97	3.25	1.08	0.3839
Error	120	361.69	3.01		
Corrected Total	149	3880.89			

Since COMSTGY occurs in both interactions, the data was sorted by COMSTGY and ANOVA was carried out to study interactions. ANOVA for NISM as the dependent variable and for COMSTGY = One-Way is given in Table 37.

Table 37. ANOVA for NISM with COMSTGY = One-Way

Source	DF	SS	MS	F	Pr > F
Model	14	1540.54	110.04	36.51	0.0001
OPTVAR	2	575.72	287.86	95.5	0.0001
TPOFPA	4	918.18	229.54	76.16	0.0001
OPTVAR*TPOFPA	8	46.64	5.83	1.93	0.0615
Error	120	361.69	3.01		

Table 38. ANOVA for NISM with COMSTGY = Two-Way

Source	DF	SS	MS	F	Pr > F
Model	14	290.79	20.77	6.89	0.0001
OPTVAR	2	141.16	70.58	23.42	0.0001
TPOFPA	4	105.96	26.49	8.79	0.0001
OPTVAR*TPOFPA	8	43.67	5.46	1.81	0.816
Error	120	361.69	3.01		

Duncan's multiple range test for NISM was carried out with operation time variability (OPTVAR) as the independent variable. The results are in Table 39.

Table 39. Duncan's Test for NISM with Independent Variable = OPTVAR

ONE-WAY			TWO-WAY			DIFFERENCE
DG	NISM	OPTVAR	DG	NISM	OPTVAR	
A	55.72	High	A	47.11	High	8.61*
B	51.44	Medium	B	45.17	Medium	6.27*
B	49.02	Low	C	43.77	Low	5.25*

Duncan's Multiple Range Test for NISM was carried out with Time Period of Periodic Actions (TPOFPA) as the independent variable.

Table 40. Duncan's Test for NISM with Independent Variable = TPOFPA

ONE-WAY			TWO-WAY			DIFFERENCE
DG	NISM	TPOFPA	DG	NISM	TPOFPA	
A	58.80	480.0	A	46.28	480	12.53*
B	51.62	80	A	45.97	160	5.65*
B	50.95	320	A	45.77	320	5.18*
C	50.07	160	A	45.71	20	4.36*
D	48.84	20	B	43.01	80	5.84*

7.6 Experimentation in the Context of Availability Variability (Factor SV5)

One source of randomness is due to machine failures. It has three components; MTBF, MTTR and 'variability' in each of these two. Thus though the mean availability might be the same at different levels, the randomness is not the same. Consider the 80% availability case. The following three levels for this factor are defined:

Level 1: MTBF = 60.0 (Uniform with half range = 5% of mean), 57.0 to 63.0
AND MTTR = 15.0 (Uniform with half range = 5% of mean), 14.25 to 15.75
Level 2: MTBF = 240.0, AND MTTR 60.0 each uniform with half range = 5% of mean
Level 3: MTBF = 480, AND MTTR =120 each uniform with half range = 5% of mean.

The MTBF and MTTR for all work stations is the same at a given level of the factor SV5. Thus at level 1, all the work stations have MTBF of 60.0 and MTTR of 15.0. The structure of RFS, the product routings, and the demands are as before. No processing time variability is involved in this stage of experimentation. Table 41 presents different factors and their levels. The experimental design is CRD with factorial arrangement of treatments. The experimental unit is the RFS with the fixed structure and fixed weekly production targets. This design is same as the one described in section 7.6 with the exception that the factor SV6 is now replaced by factor SV5. Five replications are made in each cell.

Table 41. Treatment Combinations - Variability in Machine Availability

Time Period of Periodic Action	**20.0**		**80.0**		**160.0**		**320.0**		**480.0**	
Communication Strategy	TW	OW	TW	OW	TW	OW	TW	OW	TW	OW
SV5 - Low										
SV5 - Medium										
SV5 - High										

- Over all it was found that the two-way strategy performed better across the whole range of Time Period Of Periodic Actions as compared to the one-way strategy.

- In particular, the flow time means and SSD were less for two-way as compared to one-way.
- The increase in these performance measures w.r.t. the increasing TPOFPA were less in two-way than in one-way.
- Also the increase in flow time mean and SSD with increasing availability variability, was less for two-way as compared to one-way.
- All the above remarks also apply to the performance measures NISM and PPNM.

The details of all the ANOVA tests and Duncan's Multiple Range Tests that were carried out in SAS are given in Gharpure [1994] with the SAS program.

7.7 Effect of the Optimization Objective Used in the LP

One of the research questions posed in section 4.3.1 is "What is the relationship between the types of objective functions used in the MCDM block of the control architecture and the performance of the shop?". Thus if different objectives are used in arriving at the loading rates and part-operation rates, will they cause a difference in performance? The intuitive answer is "yes". To see that this really happens the RFS shown in Figure 12 was considered.

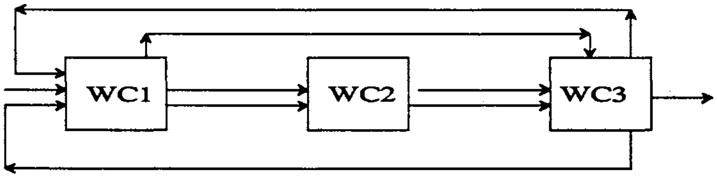


Figure 12. RFS Structure for Studying Effect of Type of Objective Function Used in LP

The routing for two products is as shown in Table 42.

Table 42. Routing Details - Effect of Type of Objective Function

Operation No.	1	2	3	4	5	6	7	8
Work Center No.	1	2	3	1	2	3	1	3
Operation Time Product 1	5	10	10	10	5	5	10	5
Operation Time Product 2	10	5	5	10	10	5	5	10

The production targets for the two products were 40 each at the end of each week for the next four weeks. Every work center had only one work station.

In the MCDM block a single-objective LP was formed with the objective to minimize the WIP waiting at work center 1, for all the days (20 days). We denote this objective as MOBJ1 in the subsequent discussions. Mathematically MOBJ1 can be expressed as;

minimize Σ (Y0101j + Y0104j + Y0107j + Y0201j + Y0204j + Y0207j)

where j = 1 to 20, for each day. Y0107 represents number of parts of product 1, waiting for operation 7.

The resulting loading rates and part-operation rates arrived at were then used in the control architecture with two-way communication strategy and with TPOFPA = 40.0 min. Then a similar LP was formed but with the objective of minimizing the WIP waiting at work center 3. We denote this objective as MOBJ2 in subsequent discussion. Mathematically MOBJ2 can be expressed as;

minimize Σ (Y0103j + Y0106j + Y0108j + Y0203j + Y0206j + Y0208j)

where j = 1 to 20, for each day. Y0208j means the number of parts of product type 2 waiting at the beginning of the jth day for operation 8. Note that both products 1 and 2 wait at the work center 3 for operations 3, 6 and 8. The resulting loading rates and operation rates were used to control the RFS. The comparative performance of the RFS under each objective is summarized in the following tables 43 to 46.

Table 43. Comparison of Queue Lengths and Time In Queue

	Queue Length		Time in Queue	
Part-Operation (WC #)	MOBJ1	MOBJ2	MOBJ1	MOBJ2
11 (1)	0.62	0.31	37.36	18.55
12 (2)	0.08	0.02	4.72	1.42
13 (3)	0.08	0.02	4.81	1.36
14 (1)	0.33	0.49	20.10	30.06
15 (2)	0.03	0.007	1.91	0.48
16 (3)	0.12	0.06	7.36	3.85
17 (1)	0.36	0.39	21.86	24.36
18 (3)	0.09	0.08	5.51	4.58
21 (1)	0.96	0.53	55.96	28.66
22 (2)	0.01	0	0.71	0
23 (3)	0.04	0.03	2.20	1.4
24 (1)	0.88	0.77	51.37	41.65
25 (2)	0.01	0.0005	0.68	0.03
26 (3)	0.05	0.02	2.98	1.22
27 (1)	0.49	0.38	28.82	21.03
28 (3)	0.14	0.11	8.11	6.22

In the above 13 (3) means part type 1, waiting for operation 3, at work center number 3. All the queue lengths and queue waiting time values are mean values.

Table 44 gives mean queue lengths in front of each work center.

Table 44. Work Centerwise Queue Lengths

Work Center 1		Work Center 2		Work Center 3	
MOBJ1	MOBJ2	MOBJ1	MOBJ2	MOBJ1	MOBJ2
3.65	2.88	0.13	0.03	0.52	0.32

Table 45 gives productwise time in system and queuing times for the two alternative objectives.

Table 45. Productwise Time In System and Queuing Time

	Time In System		Queuing Time	
	MOBJ1	MOBJ2	MOBJ1	MOBJ2
Product 1	164.17	145.00	104.17	85.00
Product 2	210.84	160.22	150.84	100.22

Table 46 compares the time in system, number in system and percentage production not made over both product types.

Table 46. Time In System and Number In System - Averaged Over Products

	MOBJ1	MOBJ2
Time In System	187.87	152.71
Number In System	6.22	5.047
PPNM (%)	8.44	5.94

The previous comparative tables show that the performance of the RFS differs in case of control under guidance of loading rates and part-operations rates as arrived at by the use of MOBJ1 (Objective 1, viz. minimization of the WIP in front of Work Center 1), vis. a vis. the performance under the guidance of MOBJ2 (minimization of WIP in front of Work Center 3). This was expected. The results are interesting. Note that the MOBJ1 is trying to minimize the WIP in front of WC1, and MOBJ2 is trying to minimize WIP in front of WC3. However it so happens that MOBJ2 not only serves its objective (that of reducing WIP in front of WC3) but also reduces WIP in front of WC1. This is even lower than that achieved by MOBJ1 !! This may be explained by observing that the WC3 has a greater number of later operations and hence it is a down-stream work center in this sense. The LP that tries to minimize the WIP in front of WC3 therefore also causes the parts operations rates and the loading rates to be such as to result in low WIP at all three work centers. There is a kind of a pull effect from WC3.

The purpose of this example was not to make a detailed study of the different minimization objectives, but rather to show that the LP objective function significantly affects the performance of the RFS. If this objective is properly chosen then the desired performance of the RFS can be achieved. The various previous sections in this chapter have shown that the type of communication strategy and the Time Period of Periodic Action matter a lot in effective control of RFS. This section brings out the point that even the type of objective function used in the LP matters.

CHAPTER 8

COMPARISONS, POSSIBLE EXTENSIONS AND FUTURE RESEARCH

8.1 Introduction

This chapter presents comparisons, possible extensions and future research for enhancing the control architecture. The second section presents an example to show the use of multiplicity of objectives in arriving at the loading rates and the part-operation rates which are then used by the hierarchical controller. The third section briefly describes an experiment that was carried out to compare the performance of the control architecture with the experiments described by Glassey and Petrakian [1989]. The performance of the architecture can be enhanced by improving the quality of each of its features. The fourth section therefore presents the possible ways in which enhancements can be made. The fifth section briefly describes the place of the architecture in the overall CIM thrust of a company. It brings out the possible relationships of the architecture with the existing information systems, existing manufacturing control philosophy, role of humans in the control loop, etc. In this context possible extensions of the architecture and system integration issues are discussed. The sixth section presents some of the future research questions that need to be answered.

8.2 Use of Multiple Objectives for Control

This section presents an example in which the use of multiple objectives for RFS control is demonstrated. Consider two products and the routings as shown in Table 47 and Table 48. Suppose two non-commensurable objectives are being pursued. The first objective is to minimize the number of parts throughout the RFS. The second objective is to maximize the sum of all the part-operation rates over all the days in a big time period.

Table 47. Routing for Product 1

Operation #	1	2	3	4
Work Center #	1	2	3	1
Time	30	20	25	20

Table 48. Routing for Product 2

Operation #	1	2	3	4
Work Center #	3	2	1	3
Time	20	30	20	25

The objectives are expressed mathematically as follows:

OBJECTIVE 1 (Z_1)

Minimize Σ(1 Y0101j + 2 Y0102j + 3 Y0103j + 4 Y0104j + 5 Y0105j + 1 Y0201j + 2Y0202j + 3 Y0203j + 4 Y0204j + 5 Y0205j)

Where j is from 1 to 20 days, Y0103j denotes number of parts of product 1 waiting for the third operation at the beginning of j^{th} day. Note that the parts waiting for later operations are viewed to be more critical as their weights in the objective function are larger than those waiting for the earlier operations.

OBJECTIVE 2 (Z_2)

Maximize Σ (U0101j + U0102j + U0103j + U0104j + U0201j + U0202j + U0203j + U0204j)

Where j is from 1 to 20, and U0203j denotes the rate at which operation 3 is being performed on product 2 during j^{th} small time interval (day).

The SIMOLP procedure [Reeves and Franz, 1985] is followed as per the following steps.

1. Solve the two single objective LP problems, one each for each of the two objective functions. Calculate two non-dominated points.

UNDER Z_1

Value of Z_1 = 1354.29, Value of Z_2 = 100.0

UNDER Z_2

Value of $Z_1 = 7845.51$, Value of $Z_2 = 150.94$

2. Have the decision maker review the above two points. If he/she settles for one of the two points then the algorithm stops, the preffered point being selected.

3. If the decision maker prefers to explore further, then form a hyper-plane passing through the two points (-1354.29, 100) and (-7845.51, 150.94) and use it as the objective function in the next iteration. The hyperplane in this case is a straight line through the two points. The weights on the two objectives are, 1 for Z_2 and 0.0078 for Z_1 as found from the equation of the line.

4. Formulate a new LP with same constraints as before but with the objective function as MAX ($Z_2 + 0.0078 (-Z_1)$).

With this objective (call it Z_3)

Value of $Z_1 = 3141.6$, Value of $Z_2 = 127.5625$

Over all value, i.e. value of $Z_3 = 103.058$

If the decision maker now prefers the solution corresponding to Z_3 then the algorithm stops or the algorithm restarts with again two points (one of them being Z_3) that are preferred out of the three points.

The effect of using the loading rates and part-operation rates corresponding to the three objectives is studied by conducting three experiments corresponding to each set of loading and part-operation rates for each non-dominated point. Various results are summarized in Table 49.

Table 49. Effect of Multiple Objectives on RFS Performance

	Z_1	Z_2	Z_3
TIME IN SYSTEM	159.82	438.23	273.77
PPNM	0.6	2.5	0.6
WC1 Utilization	0.717	0.721	0.789
WC2 Utilization	0.514	0.622	0.619
WC3 Utilization	0.716	0.878	0.869
No. IN SYSTEM	3.28	10.40	6.68

Since the objective 1 (Z1) strives for minimization of WIP, the number in system is the least (3.28) for RFS controlled under Z1 (see last row). On the other hand it is the maximum (10.40) under Z2. Under Z3 (which is a compromise between Z1 and Z2) the number in system falls in between the two previous numbers. On the other hand the utilization is more under Z2 as compared to under Z1. The PPNM (Percentage Production Not Made) is less under Z1 than under Z2. Note that under Z3, PPNM is the same as under Z1. A manager who is prepared to accept higher flow time (273.77 minutes) under Z3 as compared to Z1 (159.82 minutes) as he wishes to have higher utilization will settle for accepting Z3 over Z1. In that case, the loading rates and part-operation rates arrived under Z3 will be passed on to the controllers.

8.3 How the Architecture Performed

Glassey and Petrakian [1989] describe the performance of the use of bottleneck starvation avoidance with queue prediction in shop floor control. The experiments were performed on a hypothetical wafer fab that consisted of four work centers which are used to produce two product types. Work center 1 is the bottleneck. The description of the equipment is given in the following table.

Table 50. Details of the RFS

Work Center Number	Number of Work Stations	MTTF	MTTR
1	3	900	100
2	2	700	100
3	2	1500	100
4	2	1350	150

Table 51 clarifies the terminology in this dissertation w.r.t. that used in the cited paper.

Table 51. Comparison of Terminology

Glassey and Petrakian Terminology	Terminology in this Dissertation
Work Station	Work Center
Machine	Work Station

The fab is used to produce two types of products and their routings are given in Table 52.

Table 52. Product Routings

Route of Product 1		Route of Product 2	
Work Center Number	Processing Time	Work Center Number	Processing Time
2	36	3	31
1	20	1	25
4	120	2	29
1	25	1	20
2	29	3	34
1	25	1	25

Different start ratios were used in the experimentation. Thus a start ratio of N/M means that for every N starts of product 1 there are M starts of product 2. Three sets of simulations with three different start ratios (1/2, 5/8 and 2/5) were carried out. Five different dispatching rules were used, viz. FIFO (First in First Out), SIPT (Shortest Imminent Processing Time) also called SPT (Shortest Processing Time), SRPT (Shortest Remaining Processing Time), LDUPT (Longest Delay per Unit of Processing Time), and BQP (Bottleneck Queue Prediction). The BQP dispatching policy has been proposed by Glassey and Petrakian in their paper. "This policy makes use of queue size projections and lead time estimates. Since the bottleneck is the work center whose queue affects the most waiting time performances, the immediate objective of this policy is to minimize the size of the queue in front of the bottleneck."

To compare the performance of the architecture the author used a start ratio of 1/2, i.e. for every lot of product 1 released, 2 lots of product 2 were released. The bottleneck work center (Work Center 1) utilization was ensured to be 99.72 %. The experiment was run for a time horizon of 20 days and it was found that the average queuing time for product 1 was 792.1 min., while for product 2 it was 663.0 min. The figures reported in the previously cited reference for 99.72% utilization of the bottleneck are 772.9 min. for product 1, and 694.1 min. for product 2.

In the experimentation carried out in this dissertation, as well as in the research conducted by Glassey and Petrakian, both time to failure and time to repair distributions were Exponential with mean values as MTTF (Mean Time To Failure) and MTTR (Mean Time To Repair) respectively, as shown in Table 50.

The main purpose of this research was to provide an architecture in which all the available release rules and dispatching rules can be easily included. All said and done, the fact remains that the performance of the control architecture depends on the quality of the release rules and the dispatching rules provided, the quality of the knowledge bases and the possible enhancements in various features of the architecture. With this view in mind the next section explores the possible extensions/enhancements that can be made to the architecture.

8.4 Possible Extensions

This section discusses some of the possible extensions or enhancements. In an architecture one finds the descriptions of;

- types of building blocks
- their functionality and
- their relationships with each other.

In the case of the RFS control architecture,

- controller objects, knowledge bases, and the MCDM block are the building blocks
- the methods and duties of each of these blocks are the functionalities, and
- their relationships are captured in the messages they send to each other, the way they have been organized (is-a, has-a organization), and the way they exchange required information.

So the possible extensions to this architecture can be along the three dimensions, viz. building blocks, their functionalities and their relationships. The following paragraphs present some of these extensions. The discussion is not intended to be exhaustive. It is intended to trigger systematic thought processes.

MCDM Block:

Functionality extensions:

This block does not have any program for executing the SIMOLP. The SIMOLP procedure has to be done separately and for small problems it can be done manually. Also the block uses linear programming facility only. One can add integer programming, quadratic programming and even nonlinear programming so that more types of objective functions and constraints can be included. However, computational issues need to be addressed.

Relationship extensions:

Presently this block is used in an off-line manner. The passing of information to the controllers is not automated. Also the information from the RFS regarding state of the RFS (inventories in buffers, machine availability) is obtained in an off-line manner. This can be automated.

Block modification:

The block consists of a FORTRAN program. It may be desirable to have this block also as an object which can communicate with the standard LP package. This can lead to advantages such as modularity while implementing different extensions.

Controllers:

Functionality extensions:

Maximum enhancements can be made here. For example one can add new types of and more efficient release algorithms, more dispatching rules and other heuristics. The present architecture does not have very sophisticated release rules or dispatching rules such as described in Wein (1988) and in Glassey and Resende (1988). The better the quality of these heuristics and the more the diversity of these available rules, the better will be the architecture.

Relationship extensions:

Presently the controllers have only vertical coupling. One can explore the possibility of providing horizontal coupling between the different work center controllers.

Block modifications:

Presently the hierarchy of the controllers is fixed, viz. three levels; Plant Controller, Work Center Controller and Work Station Controller. This is inflexible. One can extend the architecture to provide flexible hierarchy of any depth.

Knowledge bases:

Functionality extensions:

As in the case of controllers, here too, the functional extensions can come about by adding more AI rules, say for example to capture the realities in the RFS control. The better these rules and the more realistic they are the better the control architecture.

Relationship extensions:

In the interrogator protocol of the controllers more methods can be added to respond to a wider variety of questions posed by the knowledge base to the controller objects.

Block modifications:

Presently there are no meta rules in the knowledge base. If one has to capture the reality of the RFS control then the number of rules in the knowledge base will increase and then the order in which these rules will be evoked in a consultation can become a very critical factor. A need will then arise to write rules about the rules, i.e. the meta rules to ensure efficient consultation.

8.5 Place of the Architecture in the Big Picture

The control architecture should be viewed as a subsystem that functions in the overall organizational context. This subsystem interacts with other subsystems of the organization by way of exchanging data, information and control commands. The role of this subsystem can be better understood by considering its place in the big picture. Further, understanding its relative role and place in the big picture can help in understanding the system integration issues that have to be addressed. Figure 13 portrays the relative position of the architecture and various interactions with other subsystems in the organization which are explained in the following paragraphs. In the following, CA stands for Control Architecture.

CA - MPS, MRP, Priority:

The CA obtains from the MPS (Master Production Schedule), MRP (Materials Requirement Planning), and Priority Setting modules; production targets, i.e. quantity and due date for each product type and priorities for the product types.

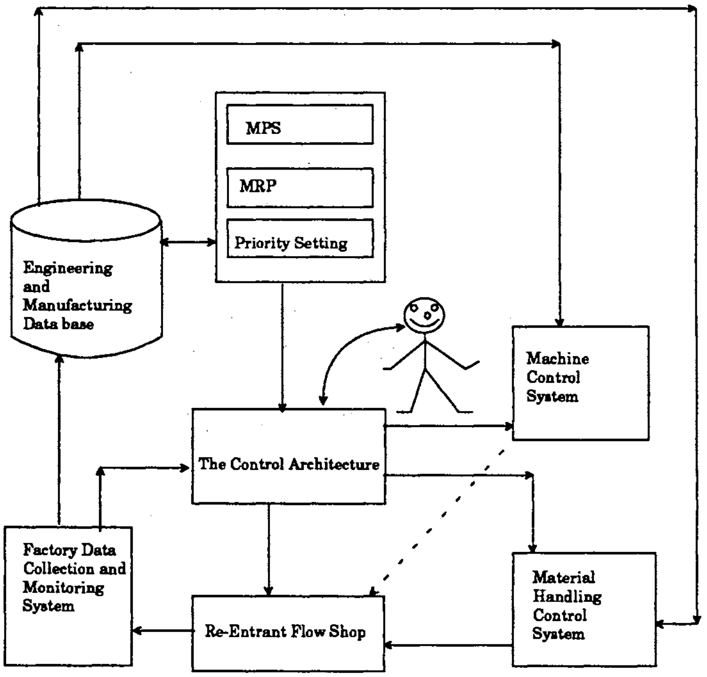


Figure 13. Place of the Architecture in the Big Picture

CA - Factory data collection and monitoring system:

The CA gets from this system status of machines, inventory positions, locations of lots waiting, and queue lengths. Note that the data gathering objects in the dissertation will be replaced by the available factory data gathering and monitoring system.

CA - Machine control system:

CA tells the machine (work station) control system the next lot to be processed. In response to this information the work station control system might requisition the necessary tools (in case of wafer fabrication a tool can be a mask of a particular pattern), necessary computer control programs (which will be then down loaded), etc. and then command the machine to start processing when the lot is loaded onto the machine. The machine control system will interact with the engineering and manufacturing data base to get the engineering details as regards required masks/tools, required computer control programs, required quality control standards etc.

CA - Material handling control system:

CA informs the material handling control system as and when a lot is completed at a particular work center. The material handling system will interact with the engineering and manufacturing data base to get details as regards the next work center for the lot and material handling characteristics of the lot, i.e. lot size, weight, dimensions, etc. (these might be factors used to decide the type of material handler to be used). Then the material handling control system will cause movement of the lot.

CA - Re-entrant flow shop:

The CA sends the control commands to the RFS as regards the lot release and dispatching decisions.

CA - Human:

The CA will have to interact with human beings at different levels and for different purposes. Thus the RFS manager will interact with CA, say, to modify the loading rates and part-operations rates, or to specify changed priorities or to specify modifications to the target production (which was specified by the MPS/MRP/Priority module in the first place), etc. The supervisors at different work centers will interact with the CA to get the next lot decisions and may modify these decisions if so desired. The CA will have to be modified to provide the capability of "human in loop".

8.6 Directions for Future Research

No research is complete. Research is an ongoing process, in which research findings are only a milestone. The research findings are not the end in themselves. During the process of research the researcher always comes up with a list of things he would have wanted to do, or wanted to answer but could not do in the limited time frame. The author of this dissertation presents below a list of questions for future research.

- What will be the impact of providing horizontal connectivity among the work center controllers on the performance of the architecture in the context of various complexity factors? It is felt that the horizontal connectivity will provide for better coordination among work centers and hence might lead to better control.
- What will be the implications of providing a control architecture with flexible depth? Presently the controller hierarchy is 3 levels deep; plant level, work center level and work station level.
- What system integration issues will have to be tackled to ensure the successful functioning of the control architecture in the organizational CIM thrust ?
- Can there be a standard methodology by which the present control architecture is quickly tuned and modified to be site-specific for tailoring to the needs of a real RFS to be controlled? Note that the present architecture does not contain site-specific algorithms, heuristics or policies for release control or dispatching. It also does not contain site-specific knowledge about the RFS in its knowledge bases.

CHAPTER 9

SUMMARY AND CONCLUSIONS

9.1 Introduction

This chapter summarizes the research efforts. First a summary of the research efforts is presented in the context of the objectives that were set to be achieved in the process of carrying out this research. The second section then presents the contributions of this research to the body of knowledge in the domain of control of RFS.

9.2 Research Summary

The goal of this research was to develop a comprehensive architecture for controlling a RFS. The goal was pursued through the attainment of six objectives that were identified at the beginning of this research. The following paragraphs report the status of attainment of these research objectives.

Objective 1 - Develop Control Architecture:

The first objective was to develop the main outline of the architecture. This required identifying the manufacturing boundary over which the control would be exercised, identifying various components of the architecture and their interactions. The objective was achieved in Chapter 6, section 6.2 where the control architecture is presented at an abstract level and the building blocks and their relationships are depicted.

Objective 2 - Develop Main Components Identified In Objective 1

This objective required that each building block of the architecture be designed in detail and also the way each block relates to other blocks be determined. Subsections 6.2.1 to 6.2.7 dealt with this research objective. Specifically, these subsections describe the detailed design philosophy of each block of the architecture. These subsections describe the services or functionalities which are provided by each block and the interactions of each block with others. Critical design features for the controller architecture were identified. Specifically, two different types of vertical communication strategies were identified and also the Time Period of Periodic Actions (TPOFPA) was identified as an important controller feature.

Objective 3 - Develop Object Oriented Framework for Hybrid-Hierarchical Controller

This objective required that a framework of controller architecture be designed in the object oriented paradigm. Section 6.3 presents the outline of the different classes, their relationships and various important messages that are used by the controller objects. This section is provided only as an aid to understanding the detailed Smalltalk-80 code. The complete listing of the code is given in Gharpure [1994].

Objective 4 - Performance Measures

The measures to be used for comparing the performance of different controllers were identified in subsection 7.3.

Objective 5 - Evaluation of the Architecture

This required that the performance of the control architecture be evaluated for different design features, i.e. for both types of communication strategies and for different Time Period of Periodic Actions (TPOFPA) in the context of several types of complexities of the RFS. To this end, section 7.2 identified several types of RFS complexities in a systematic way. Then experimentation was carried out to investigate the performance of both communication strategies for different TPOFPA in the context of different types of structural complexities at different levels of complexities. This is described in section 7.4. The performance of the control architecture for two types of communication strategies and for different TPOFPA was then evaluated in the context of variability in processing time complexity. A $3 \times 5 \times 2$ factorial arrangement of treatments in a completely randomized design was used. The experimentation details and results are presented in section 7.5. Similarly, a 3 × 5 × 2 factorial arrangement of treatments in a completely randomized design was used to carry out the experimentation for studying the effect of variability in availability of machines (Work Stations). This is described in section 7.6. Section 8.3 describes an experiment to compare the performance of the architecture with the performance as reported in Glassey and Petrakian [1989]. The RFS was a hypothetical fab described in the above cited reference. Results obtained in this research were comparable to those of Glassey and Petrakian.

Objective 6 - Further Research

This objective required identification for areas of further research. Chapter 8, (section 8.4 onwards) deals with this objective. In this chapter the possible extensions that can be made to the architecture are described. Then the place of the architecture in the overall organizational context is identified. Lastly, the directions for further research are identified.

9.3 Research Contributions

Different researchers have investigated different approaches for control of RFS. These approaches fall in the category of mathematical programming, multiclass queuing networks for control, control theoretic approach, and expert systems. The research papers in this domain deal with different techniques, or some specific combinations of release rules and dispatching rules are studied. There has been no effort in the direction of thinking about a comprehensive control architecture that forms an integral part of the overall organizational context. This research had as its main goal the design of such an architecture. The architecture proposed here can be integrated into an existing organization, can be tailored to the needs of the organization and can be enhanced/extended to reflect existing realities in the organization. Further the different types of release policies or different types of dispatching rules that are investigated by previous researchers can be included in the proposed architecture to enhance its quality. Thus the research does not favor any one release rule or any one dispatching rule or a specific combination of these. Rather it provides a framework in which the latest rules, heuristics or algorithms can be integrated.

The completion of the research objectives presented in the previous section resulted in the following research contributions to the area of control of re-entrant flow shops:

- It was proposed and demonstrated that a hierarchical and hybrid control architecture consisting of collaborating objects can be effectively used for RFS control.
- Different complexities of the RFS were identified and the performance of the architecture in the context of these complexities was investigated.
- The key design features of the architecture (different controllers, MCDM block, etc.) were identified.
- The key design factors of the control approach (time period of periodic action and type of communication strategy) were identified.
- The role played by the key design factors of the control approach in the context of different complexity factors, and also the interactions among these control approaches and complexity factors were determined via statistical experimentation.

BIBLIOGRAPHY

Akella R., Choong Y. and Gershwin S. B. 1984. "Performance of Hierarchical Production Scheduling Policy", IEEE Transactions on Components, Hybrids and Manufacturing Technology, CHMT-7, 3, 215-217.

Ammons J. C., Govindraj T. and Mitchell C. M. 1988. "A Supervisory Control Paradigm for Real-Time Control of Flexible Manufacturing Systems", Annals of Operations Research, 15, 313-335.

Bakshi M. S. and Arora S. R. 1969. "The Sequencing Problem", Management. Science, 16, B247-B263.

Basnet C. B. 1990. On-Line Scheduling and Control of Random Flexible Manufacturing Systems within an Object Oriented Framework, Unpublished Doctoral Dissertation, School of Industrial Engineering and Management, Oklahoma State University.

Bensana E., Bel G. and Dubois D. 1988. "OPAL: A Multi-Knowledge-Based System for Industrial Job-Shop Scheduling", International Journal of Production Research, 26, 5, 795-819.

Bhuskute H. C., Duse M. N., Gharpure J. T., Pratt D. B., Kamath M., and Mize J. H. 1992. "Design and Implementation of a Highly Reusable Modeling and Simulation Framework for Discrete Part Manufacturing Systems", In Proceedings of the Winter Simulation Conference, Swain J. J., Goldsman D., Crain R. C., Wilson J. R. Eds., IEEE, Piscataway, NJ, 680-688.

Bona B., Brandimarte P., Greco C. and Menga G. 1990. "Hybrid Hierarchical Scheduling and Control Systems in Manufacturing", IEEE Transactions on Robotics and Automation, 6, 6, 673-685.

Bourne D. A. and Fox M. S. 1984. "Autonomous Manufacturing: Automating the Job-Shop", IEEE Computer, 17, 9, 76-86.

Bullers W. I., Nof S. Y. and Whinston A. B. 1980. "Artificial Intelligence in Manufacturing Planning and Control", AIIE Transactions, 12, 4, 351-363.

Cao J. and Bedworth D. D. 1992. "Flow Shop Scheduling in Serial Multi-Product Processes With Transfer and Set-Up Times", International Journal of Production Research, 30, 8, 1819-1830.

Caramanis M. and Sharifnia A. 1991. "Near Optimal Manufacturing Flow Controller Design", The International Journal of Flexible Manufacturing Systems, 3, 321-336.

Chen H. and Yao D. D. 1991. "Studies on Systems with Random Disruptions via Fluid Models", Proceedings of the 1991 American Control Conference, 449-454.

Chryssolouris G. 1987. "MADEMA: an Approach to Intelligent Manufacturing Systems", CIM Review, 3, 3, 11-17.

Chryssolouris G, Wright K., Pierce J. and Cobb W. 1988. "Manufacturing Systems Operations: Dispatch Rules Versus Intelligent Control", Robotics and Computer-Integrated Manufacturing, 4, 3, 531-544.

Conterno R., Menga G. and Villa A. 1987. "Hierarchical and Decentralized Control for Batch and Repetitive Manufacturing", Proceedings - 1987 IEEE International Conference on Robotics and Automation, IEEE Computer Society Press, 1855-1860.

Connors D., Feigin G. and Yao D. 1992. "Scheduling Semiconductor Lines Using a Fluid Network Model", Proceedings - Rensselaer's Third International Conference on Computer Integrated Manufacturing, IEEE Computer Society Press, 174-183.

Conway R. W., Maxwell W. L. and Miller L. W. 1967. Theory of Scheduling, Addison - Wesley, Reading, Mass.

Daniels R. L. and Chambers R. J. 1990. "Multiobjective Flow-Shop Scheduling", Naval Research Logistics - Quarterly, 37, 981-995.

Davis W. J. and Jones A. T. 1988. " A Real-Time Production Scheduler for a Stochastic Manufacturing Environment", International Journal of Computer Integrated Manufacturing, 1, 2, 101-112.

Economides S. and Cunningham F. 1987. "Microcomputer Simulation Improves Production Scheduling For Microelectronics Manufacturers", Journal of Manufacturing Systems, 6, 4, 267-275.

Elmagharby S. E. 1968. "The Machine Sequencing Problem - Review and Extensions", Naval Research Logistics - Quarterly, 15, 205-232.

Erickson C., Vandenberge A. and Miles T. 1987. "Simulation, Animation and Shop-Floor Control", Proceedings of the 1987 Winter Simulation Conference, Thesen A., Grant H. and Kelton W. D., Eds., IEEE, Piscataway, NJ, 649-653.

Fox M. S., Allen B. and Strohm G. 1982. "Job-Shop Scheduling: An Investigation in Constraint-Directed Reasoning", Proceedings of the National Conference on Artificial Intelligence, 155-158.

Gaffar L. K. and Cochran J. K. 1989. "Devloping a Real-Time Simulation Tool for Shop-Floor Decision Making", Proceedings of the 1989 Summer Computer Simulation Conference, 79-85.

Gershwin S. B. 1989. "Hierarchical Flow Control: A Framework for Scheduling and Planning Discrete Events in Manufacturing Systems", Proceedings of the IEEE, 77, 1, 195-209.

Gharpure J. T. 1994, "Design of a control architecture for Re-Entrant Flow Shops: Listing of Smalltalk-80 code and SAS Programs", Technical Report CIM-TRS-94-JG1, Center for Computer Integrated Manufacturing, Oklahoma State University.

Glassey R. C. and Resende M. G. C. 1988. "Closed-Loop Job Release Control for VLSI Circuit Manufacturing", IEEE Transactions on Semiconductor Manufacturing, 1, 1, 36-46.

Glassey R. C. and Petrakian R. G. 1989. "The Use of Bottleneck Starvation Avoidance with Queue Predictions in Shop Floor Control", Proceedings of the 1989 Winter Simulation Conference, MacNair E. A., Musselman J. and Heidelberger P., Eds., IEEE, Piscataway, NJ, 908-917.

Grant F. H. 1989. "Scheduling Manufacturing Systems with FACTOR", Proceedings of the 1989 Winter Simulation Conference, MacNair E. A., Musselman J. and Heidelberger P., Eds., IEEE, Piscataway, NJ, 277-280.

Grant F. H., Nof S. Y. and MacFarland D. G. 1989. "Adaptive/Predictive Scheduling in Real Time", Proceedings, Advances in Manufacturing Systems Integration and Processes, Society of Manufacturing Engineers, Dearborn, Michigan.

Graves S. C. 1981. "A Review of Production Scheduling", Operations Research, 29, 4, 646-675.

Graves S. C., Meal H. C., Stefek D. and Zeghmi A. H. 1983. "Scheduling Of Re-Entrant Flow Shops", Journal of Operations Management, 3, 4, 197-207.

Harmonosky C. M. and Robohn S. F. 1991. "Real-Time Scheduling In Computer Integrated Manufacturing: A Review Of Recent Research", International Journal of Computer Integrated Manufacturing, 4, 6, 331-340.

Hintz G. W. and Zimmermann H. J. 1989. "A Method to Control Flexible Manufacturing Systems", European Journal of Operations Research, 41, 321-334.

Johri P. K., 1991. "Engineering a Circuit Board Assembly Line for a Desired Capacity and Flowtime", Journal of Manufacturing Systems, 10, 6, 492-500.

Kimemia J. and Gershwin S. B. 1983. "An Algorithm for the Computer Control of a Flexible Manufacturing System", IIE Transactions, 15, 4, 353-362.

Kumar P. R., 1993 "Re-Entrant Lines", Queueing Systems: Theory & Applications, 13, 87-110.

Kusiak A. and Chen M. 1988. "Expert Systems for Planning and Scheduling Manufacturing Systems", European Journal of Operations Research, 34, 113-130.

Lasserre J. B. 1992. "An Integrated Model for Job-Shop Planning and Scheduling", Management Science, 38, 8, 1201-1211.

LINDO, 1985, LINDO SYSTEMS CORPORATION

Lou S.X. C. and Kager P. W. 1989. "A Robust Production Control Policy for VLSI Wafer Fabrication", IEEE Transactions on Semiconductor Manufacturing, 2, 4, 159-164.

Lozinski C. and Glassey C. R. 1988. "Bottleneck Starvation Indicators for Shop Floor Control", IEEE Transactions on Semiconductor Manufacturing, 1, 4, 147-153.

Lu S. H. and Kumar P. R. 1991. "Distributed Scheduling Based on Due Dates and Buffer Priorities", IEEE Transactions on Automatic Control, 36, 12, 1406-1416.

Maimon O. Z. and Gershwin S. B. 1988. "Dynamic Scheduling and Routing for Flexible Manufacturing Systems that have Unreliable Machines", Operations Research, 36, 2, 279-292.

Maley J. G., Ruiz-Meir S. and Solberg J. J. 1988. "Dynamic Control in Automated Manufacturing: a Knoweledge-Integrated Approach", International Journal of Production Research, 26, 1739-1748.

Miller D. J. 1989. "Implementing the Results of a Manufacturing Simulation in a Semiconductor Line", Proceedings of the Winter Simulation Conference, MacNair E. A., Musselman K. J., Heidelberger P., Eds., IEEE, Piscataway, NJ, 922-929.

Narhari Y. 1993. "Analysis of Distributed Scheduling Policies in Re-Entrant Lines", Preprint.

Panwalker S. S. and Iskander W. 1977. "A survey of Scheduling Rules", Operations Research, 25, 45-61.

ParcPlace Systems 1992., Objectworks\Smalltalk Release 4.1; User's Guide, ParcPlace Systems, Inc, CA.

Pinto P. A. and Rao B. M. 1992. "Joint Lot-Sizing and Scheduling for Multi-Stage Multi-Product Flow Shops", International Journal of Production Research, 30, 5, 1137-1152.

Prartt D. B., Farrington P. A., Basnet C. B., Bhuskute H. C., Kamath M., Mize J. H. 1991. "A Framework for Highly Reusable Simulation Modeling: Separating Physical, Information, and Control Elements", In Proceedings of the 24th Annual Simulation Symposium, Nelson B. L., Kelton W. D. and Clark G. M., Eds., 254-261.

Rajendran C. and Chaudhuri D. 1990. "Heuristic Algorithms for Continuous Flow-Shop Problem", Naval Research Logistics - Quarterly, 37, 695-705.

Reeves G. R. and Franz L. S. 1985. "A Simplified Interactive Multiple Objective Linear Programming Procedure", Computers & Operations Research, 12, 6, 589-601.

Rinnooy Kan A. H. G. 1976. Machine Scheduling Problems: Classification, Complexity and Computations, Nijhoff, The Hague, Netherlands.

Roundy R., Connors D. and Feigin G. E. 1992. "Scheduling Semiconductor Manufacturing Lines", Proceedings of the 1993 NSF Design and Manufacturing Systems Conference, SME, 1163-1169.

Sharifnia A. 1988. "Production Control of a Manufacturing System with Multiple Machine States", IEEE Transactions on Automatic Control, 33, 7, 620-625.

Sharifnia A. 1992. "Performance of Production Control Methods based on Flow Control Approach", Proceedings-Rensselaer's Third International Conference on Computer Integrated Manufacturing, IEEE Computer Society Press, 184-191.

Shin K. G. and Zheng Q. 1990. "Scheduling Job Operations in an Automatic Assembly Line", The Proceedings of the 1990 IEEE International Conference on Robotics & Automation, IEEE Computer Society Press, 176-181.

Sousa J. B. and Pereira F. L. 1992. "A Hierarchical Framework for the Optimal Flow Control in Manufacturing Systems", Proceedings-Rensselaer's Third International

Conference on Computer Integrated Manufacturing, IEEE Computer Society Press, 278-286.

Stecke K. E. and and Kim I. 1991. "A Flexible Approach to Part Selection in Flexible Flow System Using Part Mix Ratios", International Journal of Production Research, 29, 1, 53-75.

Steffen M. S. 1986. "A Survey of Artificial Intelligence-Based Scheduling Systems" Proceedings, Fall Industrial Engineering Conference, Institute of Industrial Engineers, Norcross, Georgia.

Tabak D. and Kuo B. C. 1971. Optimal Control By Mathematical Programming, Prentice Hall, Inc, Englewood Cliffs, NJ, 59-69.

Taillard E. 1990. "Some Efficient Heuristic Methods for the Flow Shop Sequencing Problem", European Journal of Operations Research, 47, 65-74.

Violette J. and Gershwin S. B. 1992. "Decomposition of Control in a Hierarchical Framework for Manufacturing Systems", Proceedings of the 1991 American Control Conference, IEEE Computer Society Press, 465-474.

Vollmann T. E., Berry W. L. and Whybark D. C. 1992. Manufacturing Planning and Control Systems, Second Edition, 178-181.

Wein L. M. 1988. "Scheduling Semiconductor Wafer Fabrication", IEEE Transactions on Semiconductor Manufacturing, 1, 3, 115-130.

XEROX 1991. Xerox Humble Reference Manual, Version 4, Xerox Corporation, CA.

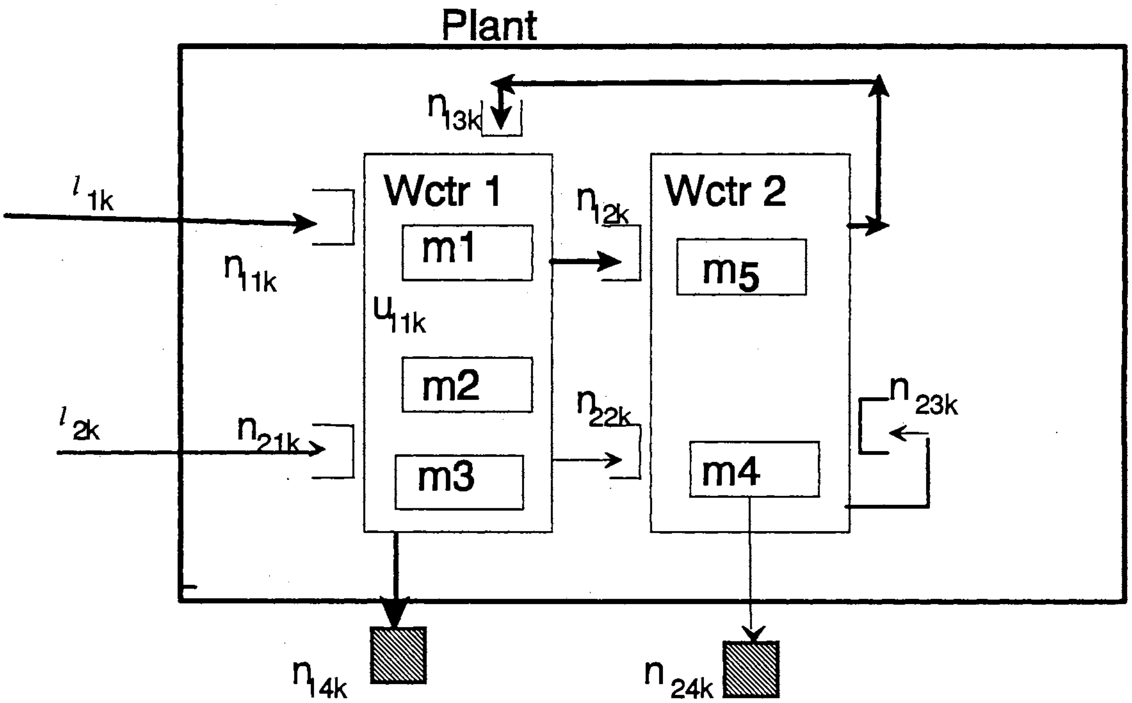
Yancey D. P. and Peterson S. 1989. "Implementation of Rule-Based Technology in a Shop Scheduling System", Proceedings of the 1989 Winter Simulation Conference, MacNair E. A., Musselman J. and Heidelberger P., Eds., 865-873.

Zhijun H. and Kai, 1990. "A New Control Method and System for FMS Job Scheduling", Transactions of the North American Manufacturing Research Institution of SME 1990, Journal of Production Research, 27, 1603-1623.

APPENDIXES

APPENDIX 1

RFS AS A DYNAMICAL SYSTEM



The above figure shows an example re-entrant flow shop. Various notations are explained below. The figure does not show the input buffer to the plant which is assumed to be of infinite capacity, having a large supply of raw material. The shaded rectangles are the output storage spaces of infinite capacity where finished products are stored and demand is satisfied from here. On the next page we will express the number of parts in various buffers at the beginning of a small time interval as related to the number of parts at the beginning of the previous small time interval and the values of the control variables (loading rates and production rates for each part-operation combination) in the previous small time interval.

$l_1(k)$ = Loading rate for part type 1 in kth small time interval (decision variable)

$l_2(k)$ = Loading rate for part type 2 in kth small time interval (decision variable)

$n_{ij}(k)$ = Number of parts of type i waiting for operation j

$u_{ij}(k)$ = Production rate of operation i on part type j (decision variable)

d_j = Demand for part type j (NOT rate)

t_{ij} = Operation time for operation i on part type j

The loading rate is the rate at which parts (raw material for parts) are released into the shop. Thus its units can be number of pieces per hour. The production rate for a part-operation combination is the number of times an operation on a part type is performed at a work center in say one hour. Let $y_{ij}(k)$ denote the level of parts of type i at operation j (continuous variable) then the equations relating buffer contents at beginning of (K+1) with buffer contents at the beginning of (K), and the input rate and the output rate from the buffer can be written as shown below.

$$y_{11}(k+1) = y_{11}(k) + [l_1(k) - u_{11}(k)]t_s$$

$$y_{12}(k+1) = y_{12}(k) + [u_{11}(k) - u_{12}(k)]t_s$$

All equations are not written here. Instead they are shown in the matrix form below.

$$\begin{bmatrix} y_{11}(k+1) \\ y_{12}(k+1) \\ y_{13}(k+1) \\ y_{14}(k+1) \\ y_{21}(k+1) \\ y_{22}(k+1) \\ y_{23}(k+1) \\ y_{24}(k+1) \end{bmatrix} = \begin{bmatrix} 1&0&0&0&0&0&0&0 \\ 0&1&0&0&0&0&0&0 \\ 0&0&1&0&0&0&0&0 \\ 0&0&0&1&0&0&0&0 \\ 0&0&0&0&1&0&0&0 \\ 0&0&0&0&0&1&0&0 \\ 0&0&0&0&0&0&1&0 \\ 0&0&0&0&0&0&0&1 \end{bmatrix} \begin{bmatrix} y_{11}(k) \\ y_{12}(k) \\ y_{13}(k) \\ y_{14}(k) \\ y_{21}(k) \\ y_{22}(k) \\ y_{23}(k) \\ y_{24}(k) \end{bmatrix} + \begin{bmatrix} t_s & 0 & -t_s & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & t_s & -t_s & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & t_s & -t_s & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & t_s & 0 & 0 & 0 \\ 0 & t_s & 0 & 0 & 0 & -t_s & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & t_s & -t_s & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & t_s & -t_s \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & t_s \end{bmatrix} \begin{bmatrix} l_1(k) \\ l_2(k) \\ u_{11}(k) \\ u_{12}(k) \\ u_{13}(k) \\ u_{21}(k) \\ u_{22}(k) \\ u_{23}(k) \end{bmatrix}$$

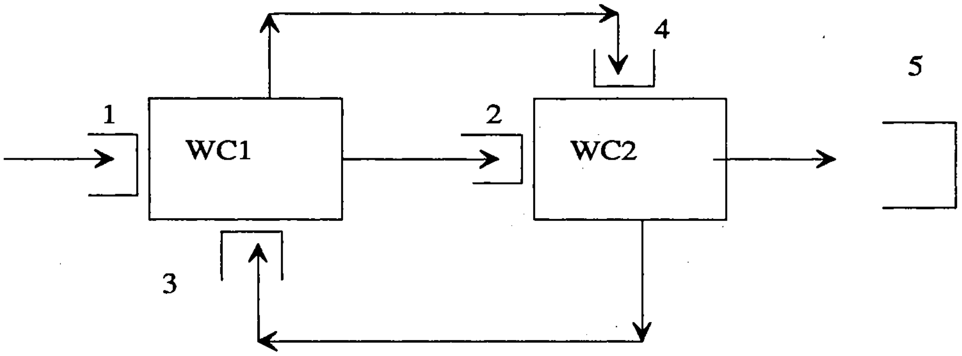
Thus we can write the equations as

$$\vec{y}(k+1) = \vec{A}\,\vec{y}(k) + \vec{B}\,\vec{u}(k)$$

APPENDIX 2

LINEAR PROGRAMMING FORMULATION

Consider the following RFS.



Suppose there is only one product. Consider the following table which gives process times and yield at each work center.

Operation No.	1	2	3	4
Operation Time	p1	p2	p3	p4
Yield (Fraction)	x1	x2	x3	x4

The material flow balance equation (Constraints) will be as follows:

$Y11(k+1) = Y11k + [L1k - U11k]\ ts$

$Y12(k+1) = Y12k + [U11k * x1 - U12k]\ ts$

$Y13(k+1) = Y13k + [U12k * x2 - U13k]\ ts$

$Y14(k+1) = Y14k + [U13k * x3 - U14k]\ ts$

$Y15(k+1) = Y15k + [U14k * x4]\ ts$

All the above equations are for all k from 1 to 20 if there are 20 days in big time period. In above equations Yijk is number of parts of product type i waiting for operation j at the beginning of kth day. Lik is the rate (parts/hour) at which the parts for type i are loaded in the RFS during kth day. Uijk is the rate (parts/hour) at which the parts of type i are being processed at operation j during the kth day. ts is the length of small time interval (say 8 hours). The last equation will be modified for those days at the end of which demand occurs as follows.

$Y15(k+1) = Y15k + [U14k * x4]\ ts - dk$

Where dk is the demand that occurs at the end of kth day.

The limited buffer capacities constraints will be as follows:

$Y11k + Y13k \leq b1$

$Y12k + Y14k \leq b2$

Where b1 and b2 are the buffer capacities of work center 1 and 2 respectively.

The capacity constraints will be as follows:

$U11k * p1 + U13k * p3 \leq n1 * (MTBF1 / (MTBF1 + MTTR1))$

$U12k * p2 + U14k * p4 \leq n2 * (MTBF2 / (MTBF2 + MTTR2))$

Where n1 and n2 are the number of work stations in work centers 1 and 2. MTBF1 and MTTR1 are the Mean Time Between Failure and Mean Time To Repair for work stations in work center 1.

If the RFS starts with no parts in any of the buffers in front of the work centers the constraints reflecting the initial conditions will be written as follows:

$Y111 = 0$

$Y121 = 0$

$Y131 = 0$

$Y141 = 0$

2

VITA

Jagannath T. Gharpure

Candidate for the Degree of

Doctor of Philosophy

Thesis: DESIGN OF A CONTROL ARCHITECTURE FOR RE-ENTRANT FLOW SHOPS

Major Field: Industrial Engineering and Management

Biographical:

Personal Data: Born in Bharuch, Gujarat, India, May 29, 1956, the son of Trimbak G. and Leela T. Gharpure.

Education: Graduated from The D.H. Parikh Model High School, Baroda, India, in May 1972; received Bachelor of Engineering Degree in Electrical Engineering from The M.S. University, Baroda, India in May 1978; received Master of Science Degree in Industrial Engineering from NITIE, Bombay, India in June 1981; completed requirements for the Doctor of Philosophy degree at Oklahoma State University in July 1994.

Professional Experience: Manufacturing Systems Engineer, Rallies India Ltd., from July 1981 to April 1982; Sales Officer, Genelec Ltd., from May 1982 to April 1983; Assistant Sales Manager, The Premier Automobiles Ltd., from May 1983 to March 1986; Lecturer (IE), NITIE, from April 1986 to April 1988; Assistant Professor (IE), NITIE, from April 1988 to December 1990; Research Associate, School of Industrial Engineering and Management, Oklahoma State University, from July 1991 to June 1994.

Professional Membership: Alpha Pi Mu, IIE, Tau Beta Pi, ORSA