

**APPLICATION OF LEAN SCHEDULING AND PRODUCTION  
CONTROL IN NON-REPETITIVE MANUFACTURING SYSTEMS  
USING INTELLIGENT AGENT DECISION SUPPORT**

A thesis submitted for the degree of Doctor of Philosophy

by

Theopisti C. Papadopoulou

School of Engineering & Design

Brunel University

January, 2013

## Abstract

Lean Manufacturing (LM) is widely accepted as a world-class manufacturing paradigm, its currency and superiority are manifested in numerous recent success stories. Most lean tools including Just-in-Time (JIT) were designed for repetitive serial production systems. This resulted in a substantial stream of research which dismissed a priori the suitability of LM for non-repetitive non-serial job-shops.

The extension of LM into non-repetitive production systems is opposed on the basis of the sheer complexity of applying JIT pull production control in non-repetitive systems fabricating a high variety of products. However, the application of LM in job-shops is not unexplored. Studies proposing the extension of leanness into non-repetitive production systems have promoted the modification of pull control mechanisms or reconfiguration of job-shops into cellular manufacturing systems.

This thesis sought to address the shortcomings of the aforementioned approaches. The contribution of this thesis to knowledge in the field of production and operations management is threefold:

Firstly, a Multi-Agent System (MAS) is designed to directly apply pull production control to a good approximation of a real-life job-shop. The scale and complexity of the developed MAS prove that the application of pull production control in non-repetitive manufacturing systems is challenging, perplex and laborious.

Secondly, the thesis examines three pull production control mechanisms namely, Kanban, Base Stock and Constant Work-in-Process (CONWIP) which it enhances so as to prevent system deadlocks, an issue largely unaddressed in the relevant literature.

Having successfully tested the transferability of pull production control to non-repetitive manufacturing, the third contribution of this thesis is that it uses experimental and empirical data to examine the impact of pull production control on job-shop performance.

The thesis identifies issues resulting from the application of pull control in job-shops which have implications for industry practice and concludes by outlining further research that can be undertaken in this direction.

# Contents

List of Abbreviations.....	viii
List of Figures .....	xiii
List of Tables.....	xiv
Acknowledgements.....	xvi
Dedication.....	xvii
1 Introduction .....	1
1.1 Research context.....	1
1.2 Background to the main research themes.....	3
1.2.1 Production Planning and Control .....	3
1.2.2 Push versus pull production control.....	4
1.2.3 Repetitive versus non-repetitive manufacturing .....	6
1.3 Research rationale .....	7
1.4 Research aim and objectives .....	13
1.5 Research boundaries and limitations .....	14
1.6 Thesis structure.....	15
2 Background research into lean thinking .....	18
2.1 Origins in Japanese manufacturing .....	19
2.1.1 The Toyota Production System (TPS).....	19
2.1.2 Just-in-Time (JIT) production .....	20
2.2 Dissemination of the TPS .....	22
2.2.1 Extension to Japanese suppliers and manufacturers.....	23
2.2.2 Diffusion in western manufacturing .....	25
2.3 Emergence of lean thinking .....	27
2.4 Lean at the enterprise level .....	31
2.5 Lean implementation .....	35
2.5.1 TPS practices and underpinning philosophy.....	36
2.5.2 JIT production elements and techniques.....	39
2.5.3 Lean Manufacturing facets .....	44
2.6 Why become lean? .....	47
2.6.1 Lean benefits.....	47
2.6.2 Lean metrics and performance measurement systems.....	50
2.6.3 Assessment of degree of leanness .....	52
2.7 Common lean misconceptions.....	54
2.7.1 Is lean simply a set of tools? .....	54

2.7.2	Lean versus rival world-class manufacturing paradigms.....	55
2.7.3	Lean applicability issues and limitations.....	58
2.8	Recent industrial applications of the lean paradigm .....	61
2.9	Chapter summary .....	64
3	Production Planning and Control (PPC) .....	68
3.1	Contextual factors influencing production scheduling systems.....	69
3.1.1	Nature of demand.....	70
3.1.2	Production orientation and order fulfillment policy.....	70
3.1.3	Manufacturing process and shop-floor configuration.....	72
3.2	PPC framework.....	76
3.2.1	Aggregate planning and the Master Production Schedule (MPS) .....	78
3.2.2	Material Requirements Planning (MRP) .....	80
3.2.3	Capacity planning .....	83
3.3	Scheduling in manufacturing systems .....	85
3.3.1	Scheduling in the process industries .....	87
3.3.2	Scheduling in project manufacturing.....	88
3.3.3	Scheduling in flow systems.....	89
3.3.4	Scheduling in job-shops.....	97
3.3.4.1	Loading .....	102
3.3.4.2	Sequencing.....	103
3.3.4.3	Forward/backward Scheduling.....	107
3.4	Control .....	109
3.4.1	Push production control .....	110
3.4.2	Pull production control .....	111
3.4.2.1	Kanban system operating principles .....	112
3.4.2.2	Kanban supporting infrastructure.....	115
3.4.3	Theory of Constraints (TOC).....	117
3.4.4	Kanban variants and hybrids .....	118
3.4.5	Kanban design and modelling.....	122
3.4.6	Performance comparisons of pull control mechanisms .....	126
3.5	Chapter summary .....	128
4	Application of simulation techniques in production planning and control .....	132
4.1	Modeling in decision-making.....	133
4.1.1	Prescriptive versus descriptive modelling.....	134
4.1.2	Simulation suitability and modeling process .....	136
4.2	Conventional simulation.....	139

4.3 Agent-based simulation (ABS) .....	141
4.3.1 Intelligent software agents .....	141
4.3.2 Multi-agent systems (MAS) and applicability .....	143
4.4 Applications of simulation modeling in production planning and control.....	147
4.4.1 Production planning and control using MAS.....	147
4.4.2 Agent-based job-shop scheduling .....	151
4.4.3 Simulation modeling in applications of pull control in job-shops.....	154
4.4.3.1 Use of conventional simulation.....	154
4.4.3.2 Use of agent-based simulation.....	158
4.5. Chapter summary .....	161
5 Lean scheduling and control of non-repetitive production systems using intelligent agent decision support .....	165
5.1 Scheduling system overview .....	165
5.2 Job-shop infrastructure and operation.....	166
5.2.1 Job-shop operation under push production control .....	166
5.2.2 Job-shop operation using pull production control mechanisms.....	166
5.2.3 Job-shop's reaction to unexpected changes .....	172
5.3 Job-shop scheduling system configuration.....	174
5.4 Agent-based simulation model .....	179
5.4.1 Overview of agents in BASS .....	181
5.4.1.1 System Manager Agent (SMA).....	181
5.4.1.2 Customer Agent (CA) .....	183
5.4.1.3 Failure Manager Agent (FMA) .....	183
5.4.1.4 Dispatcher Agent (DA) .....	183
5.4.1.5 Machine Agent (MA) .....	185
5.4.1.6 Job Manager Agent (JMA).....	187
5.4.1.7 Workstation Supervisor Agent (WSA) .....	192
5.4.1.8 Workstation Input Buffer Agent (WIBA) .....	192
5.4.1.9 Workstation Output Buffer Agent (WOBA) .....	193
5.4.1.10 FGI Manager Agent (FGIMA) .....	194
5.4.1.11 Performance Monitor Agent (PMA) .....	195
5.5 BASS development .....	195
5.5.1 JACK™ overview .....	196
5.5.2 Implementation of BASS using JACK™ .....	200
5.6 BASS verification.....	203
5.7 Chapter summary .....	204

6 Experimental and empirical validation of the application of pull production control to non-repetitive production systems.....	207
6.1 Experimental testing using BASS.....	207
6.1.1 Design of experiments.....	207
6.1.2 Computational results.....	212
6.1.3 Sensitivity analysis .....	224
6.2 Application of pull production control to an industrial case-study.....	227
6.2.1 Company profile .....	227
6.2.2 Data, limitations and implementation issues .....	228
6.2.3 Simulation results and analysis .....	232
6.3 Chapter summary .....	234
7 Conclusions.....	238
7.1 Summary of research rationale, aims and key contributions.....	238
7.1.1 Historical evolution, implementation and current state of the lean paradigm .....	239
7.1.2 Pull production mechanisms within Production Planning and Control (PPC) hierarchical frameworks.....	240
7.1.3 The transferability of pull production control using simulation .....	242
7.1.4 Gaps and limitations of applications of pull production control to non-repetitive manufacturing systems .....	242
7.1.5 Design and development of an intelligent agent decision support system for the extension of pull production control to lean job-shops.....	243
7.1.6 Evaluation of job-shop performance following the introduction of pull production control .....	244
7.2 Implications for theory .....	246
7.3 Implications for current industry practice .....	247
7.4 Directions for further research.....	247
References .....	249
Appendix A: Most common dispatching rules in production scheduling.....	291
Appendix B: Complete listing of external events in BASS.....	294
Appendix C: JACK™ Design of Machine Agent (MA) .....	297
Appendix D: BASS verification using experiment BVE1.....	309
Appendix E: JACK™ Performance tracing log for experiment BVE1 .....	313
Appendix F: BASS verification using experiment BVE2 .....	322
Appendix G: BASS verification using experiment BVE3 .....	327
Appendix H: BASS verification using experiment BVE4.....	331
Appendix I: Stochastic data used in experiment LA40 .....	337

Appendix J: Stochastic data used in experiment ABZ7 .....	339
Appendix K: Stochastic data used in experiment ABZ9 .....	341
Appendix L: Stochastic data used in experiment YN1.....	343
Appendix M: Stochastic data used in experiment YN4.....	345

## List of Abbreviations

ABS	Agent-Based Simulation
ACO	Ant Colony Optimisation
ADI	Advanced Demand Information
AFT	Average flow time
AGV	Automated Guided Vehicles
AHP	Analytic Hierarchy Process
AI	Artificial Intelligence
AIAG	Automotive Industry Action Group
ALBP	Assembly Line Balancing Problem
ANP	Analytic Network Process
AOS	Agent Oriented Software
AOSE	Agent-Oriented Software Engineering
API	Application Programming Interface
APICS	American Production and Inventory Control Society (APICS)
APS	Advanced Planning System
AS/RS	Automated Storage and Retrieval System
ATO	Assemble-to-Order
BASS	Brunel Agent Scheduling System
BDI	Beliefs-Desires-Intentions
BOM	Bill-of-Materials
BPR	Business Process Reengineering
CA	Customer Agent
CABS	Coordination for Avoiding Machine Starvation
CCPM	Critical Chain Project Management
CIM	Computer-Integrated Manufacturing
CONWIP	Constant Work-in-Progress
COVERT	Cost Over Time
CPM	Critical Path Method
CPR	Capacity Requirements Planning
CR	Critical Ratio
CTBS	Customised Token-Based System
DA	Dispatcher Agent
DAI	Distributed Artificial Intelligence
DBR	Drum-Buffer-Rope
DDM	Distributed Decision-Making
DES	Discrete-Event Simulation
DRC	Dual Resource Constrained



DTO	Design-to-Order
ECR	Enhanced Critical Ratio
EDD	Earliest Due Date
EDI	Electronic Data Interchange
EKCS	Extended Kanban Control System
EOQ	Economic Order Quantity
ERP	Enterprise Resource Planning
ETO	Engineer-to-Order
FASFS	First Arrival At Shop First Served
FCFS	First Come First Served
FGI	Finished Goods Inventory
FGIMA	FGI Manager Agent
FIPA	Foundation of Intelligent Physical Agents
FJSSP	Flexible Job-Shop Scheduling problem
FKS	Flexible Kanban System
FMA	Failure Manager Agent
FOQ	Fixed Order Quantity
FOR	Fewest Operations Remaining
FPS	Ford Production System
GA	Genetic Algorithm
GKCS	Generalised Kanban Control System
GKS	Generic Kanban System
GM	General Motors
GT	Group Technology
GUI	Graphical User Interface
HIHS	Horizontally Integrated Hybrid Systems
HPF	Highest Pull Frequency
HPP	Hybrid Push/Pull
HRM	Human Resources Management
HVLV	High Variety low Volume
I/O	Input/Output
IMVP	International Motor Vehicle Program
JADE	Java Agent Development
JAL	JACK™ Agent Language
JDE	JACK™ Development Environment
JIT	Just-in-Time
JMA	Job Manager Agent
JSSP	Job-Shop Scheduling Problem
KBACO	Knowledge-Based Ant Colony Optimisation
KPS	Kawasaki Production System

LAI	Lean Advancement Initiative
LAI	Lean Aircraft Initiative
LCC	Least Changeover Cost
LCFS	Last Come First Served
LESAT	Lean Enterprise Self-Assessment Tool
LFJ	Least Flexible Job
LM	Lean manufacturing
LOPNR	Least Number of Operations Remaining
LP	Lean Production
LPT	Longest Processing Time
LTWK	Least Total Work
LWKR	Least Work Remaining
MA	Machine Agent
MABS	Multi Agent-Based Simulation
MAS	Multi-Agent System
MCE	Manufacturing Cycle Efficiency
MDD	Modified Due-Date
MILP	Mixed Integer Linear Programming
MINLP	Mixed Integer Non-Linear programming
MIT	Massachusetts Institute of Technology
MKS	Modified Kanban System
MNC	Maximum Number of Cards
MOPNR	Most Operations Remaining
MPS	Master Production Schedule
MRP	Material Requirement Planning
MRP II	Material Resources Planning
MS	Minimum Slack
MTBF	Mean Time Between Failures
MTO	Make-to-Order
MTS	Make-to-Stock
MTTR	Mean Time To Repair
MWKR	Most Work Remaining
NP	Non-Polynomial
NUMMI	New United Motor Manufacturing Inc.
ODD	Operation Due Date
OEE	Overall Equipment Effectiveness
OPT	Optimised Production Technology
OR	Operations Research
ORR	Order Review and Release
PERT	Programme Review and Evaluation Technique

PMA	Performance Monitor Agent
POLCA	Paired-Cell Overlapping Loops of Cards with Authorisation
POQ	Period Order Quantity
PPC	Production Planning and Control
PSP	Production Smoothing Problem
QC	Quality Control
QRM	Quick Response Manufacturing
R&D	Research and Development
RCCP	Rough-Cut Capacity Planning
RFID	Radio Frequency Identification Sensors
RKS	Reconfigurable Kanban System
RL	Reinforcement Learning
RL	Repetitive Lots
RMG	Repetitive Manufacturing Group
ROA	Return on Assets
ROE	Return on Equity
ROS	Return on Sales
S/RO	Slack Per Remaining Operation
SALBP	Simple Assembly Line Balancing Problem
SD	System Dynamics
SIRO	Service In Random Order
SMA	System Manager Agent
SME	Small-to-Medium Enterprise
SMED	Single-Minute Exchange of Die
SPC	Statistical Process Control
SPT	Shortest Processing Time
SSD	Scheduled Start Date
SST	Shortest Setup Time
STN	State-Task Networks
STPT	Shortest Total Processing Time
TADE/T	Total Absolute Deviation of Earliness/Tardiness
TBM	Time-Based Manufacturing
TMC	Toyota Motor Corporation
TMC	Time-based Competition
TOC	Theory of Constraints
TP	Throughput
TPS	Toyota Production System
TQM	Total Quality Management
TWC	Total Work Content
UK	United Kingdom

UML	Unified Modelling Language
US	United States
VE	Value Engineering
VIHS	Vertically Integrated Hybrid Systems
WIBA	Workstation Input Buffer Agent
WINQ	Work In The Next Queue
WIP	Work-in-Progress
WOBA	Workstation Output Buffer Agent
WSA	Workstation Supervisor Agent
WSPT	Weighted Shortest Processing Time

## List of Figures

<b>Figure 1.1:</b> Outline of thesis .....	17
<b>Figure 2.1:</b> Landmark reports and key milestones in lean thinking (Source: Holweg, 2007).....	29
<b>Figure 2.2:</b> LAI Lean enterprise transformation roadmap (Source: Nightingale et al., 2010) .....	34
<b>Figure 2.3:</b> TPS four level prism delivery model (Source: Towill, 2007) .....	38
<b>Figure 2.4:</b> Principle component analysis of the four lean bundles (Source: Shah and Ward, 2003) .....	45
<b>Figure 3.1:</b> P:D ratios in different demand response policies (Source: Slack et al., 2010) .....	72
<b>Figure 3.2:</b> Workflow diversity in job-shops .....	74
<b>Figure 3.3:</b> Types of production layouts suitable for different levels of production volume and variety (Source: Groover, 2008) .....	75
<b>Figure 3.4:</b> Planning and control hierarchy.....	86
<b>Figure 3.5:</b> Conceptual job-shop scheduling framework .....	109
<b>Figure 4.1:</b> Modelling steps .....	137
<b>Figure 4.2:</b> Schematic of papers reporting applications of simulation modelling in the wider context of production planning and control .....	162
<b>Figure 5.1:</b> Operation of the Kanban-controlled job-shop.....	167
<b>Figure 5.2:</b> Batch-splitting in the Kanban-controlled job-shop .....	170
<b>Figure 5.3:</b> Batch-splitting in the job-shop controlled by the Base stock mechanism.	171
<b>Figure 5.4:</b> Overview of the push/pull-controlled job-shop's configuration .....	180
<b>Figure 5.5:</b> Overview of the JACOB™ interface.....	199
<b>Figure 5.6:</b> The MA's main constructs in JDE and design view diagram of its external interface.....	201
<b>Figure 5.7:</b> Design view diagram of the MA's capabilities .....	202
<b>Figure 5.8:</b> Design view diagram of the MA's TaskProcessing structure .....	202
<b>Figure 5.9:</b> Design view diagram of the MA's StatusMonitoring structure .....	202
<b>Figure 6.1:</b> Aircraft fasteners produced by LCA .....	227

## List of Tables

<b>Table 3.1:</b> Comparison of process and product layouts.....	76
<b>Table 3.2:</b> Performance objectives in operations scheduling (T'Kindt and Billaut, 2006) .....	100
<b>Table 3.3:</b> Commonly cited dispatching rules and reporting sources (Multiple bibliographic sources) .....	105
<b>Table 6.1:</b> Experimentation parameters for static problems .....	208
<b>Table 6.2:</b> Experimentation parameters for dynamic problems .....	211
<b>Table 6.3:</b> Performance output for LA16 (10x10) 50 job instances.....	213
<b>Table 6.4:</b> Performance output for LA20 (10x10) 50 job instances.....	213
<b>Table 6.5:</b> Performance output for LA21 (15x10) 75 job instances.....	214
<b>Table 6.6:</b> Performance output for LA25 (15x10) 75 job instances.....	214
<b>Table 6.7:</b> Performance output for LA26 (20x10) 100 job instances.....	215
<b>Table 6.8:</b> Performance output for LA30 (20x10) 100 job instances.....	215
<b>Table 6.9:</b> Performance output for LA31 (30x10) 150 job instances.....	216
<b>Table 6.10:</b> Performance output for LA35 (30x10) 150 job instances.....	216
<b>Table 6.11:</b> Performance output for SWV15 (50x10) 250 job instances.....	217
<b>Table 6.12:</b> Performance output for SWV20 (50x10) 250 job instances.....	217
<b>Table 6.13:</b> Performance output for LA40 (15x15) 31 job instances including rush orders.....	220
<b>Table 6.14:</b> Performance output for ABZ7 (20x15) 38 Instances including rush orders .....	221
<b>Table 6.15:</b> Performance output for ABZ9 (20x15) 38 Instances including rush orders .....	221
<b>Table 6.16:</b> Performance output for YN1 (20x20) 38 Instances including rush orders .....	222
<b>Table 6.17:</b> Performance output for YN4 (20x20) 38 Instances including rush orders .....	222
<b>Table 6.18:</b> Performance output for LA40 (15x15) 31 job instances.....	225
<b>Table 6.19:</b> Performance output for ABZ7 (20x15) 38 job instances.....	225
<b>Table 6.20:</b> Performance output for ABZ9 (20x15) 38 job instances.....	225
<b>Table 6.21:</b> Performance output for YN1 (20x20) 38 job instances.....	225
<b>Table 6.22:</b> Performance output for YN4 (20x20) 38 job instances.....	225
<b>Table 6.23:</b> List of operations and workstation data LCA case study .....	228
<b>Table 6.24:</b> Operations layout for sample of 10 orders LCA case study.....	229
<b>Table 6.25:</b> Part data for sample order 1: LCA case study .....	231

<b>Table 6.26:</b> Part data for sample order 2: LCA case study .....	231
<b>Table 6.27:</b> Performance of push and pull production control LCA case study .....	233

## Acknowledgements

First and foremost, I would like to wholeheartedly thank my PhD supervisors, Dr Alireza (Ali) Mousavi and Mr Peter Broomhead. Ali kindly agreed to take over the supervision of this PhD at a very difficult phase in my life. Completing the PhD whilst working full-time has been challenging at times but Ali remained confident in my ability to complete this research. I am grateful for the time both Peter and Ali devoted and all the ideas they contributed in order to improve the quality of this thesis.

I am grateful to Professor Sarah Sayce for supporting me and helping me secure sponsorship for the final years of my research. I thank my colleagues at work Mr Timothy Bennett for his continuous encouragement and support and Dr Karen Clarke for always being helpful, supportive and a valuable source of motivation.

I would like to acknowledge the technical advice offered by Mr David Shepherdson during the implementation of the MAS developed in this thesis and thank him for the excellent collaboration.

I am grateful to LCA, the manufacturing company which generously provided the data for the industrial case-study presented in this thesis and for offering a number of factory visits and interviews with their production manager.

My time at Brunel University was made enjoyable with the help of other research students I met and became friends with. I also felt accepted as a valuable member of an exciting and stimulating scientific and research community and this is thanks to the support and efforts of the University's academic and administrative staff.

I would like to thank my parents who raised me with love and helped me in all my pursuits. My mother Kalliopi and sister Nelli for their love and outstanding support and encouragement even through the hardest of times.

Above all, I thank God for his grace, guidance and protection and for helping me achieve all that I have accomplished in my life.



## **Dedication**

To the memory of my late father Christos Papadopoulos with all my love and warm thanks for his loving care and support in all the years we had together.

# 1 Introduction

This chapter sets the scene for the research carried out in this thesis. Here the origins, core and evolution of lean thinking are presented. Drawing evidence from the relevant literature, the following sections support the rationale for this research and provide a brief overview of the intended contribution to the existing body of knowledge in this area. Subsequently, research questions are formulated and linked to a set of research objectives. The boundaries and limitations of this research are also discussed. This chapter concludes by providing an outline of the thesis layout and synopsis of the following chapters.

## 1.1 Research context

Following its introduction on the shop-floors of Japanese factories more than half a century ago, lean thinking continues to receive the undiminishing attention of academic research and industry practice. Studies considering leanness in the context of best manufacturing and operations management paradigms are at the forefront of research (Narasimhan et al., 2005; Boyle and Scherrer-Rathje, 2009; Hallgren and Olhager, 2009). Whilst several scholarly works investigate lean implementation enablers (Worley and Doolen, 2006; Scherrer-Rathje et al., 2009) another significant volume of the relevant literature considers the issue of measuring lean adoption and success (Bayou and De Korvin, 2008; Wan and Chen, 2008; Eroglu and Hofer, 2010; Vinodh and Chintha, 2010; Bhasin, 2012; Panizollo, et al., 2012; Stump and Badurdeen, 2012).

The first form of leanness, the Toyota Production System (TPS) was an innovation developed by revolutionary Taiichi Ohno of Toyota Motor Corporation (TMC) (Hines et al., 2004). The early post-World War II years found the Japanese automobile industry fighting for its survival faced with capital constraints, scarce resources and fierce competition by domestic car manufacturers. The pioneers of TMC studied the mass production models of their biggest American competitors mainly Ford and General Motors only to recognise that under the circumstances prevailing at the time in Japan they were unworkable. Holweg (2007) maintains that Ohno's views which since revolutionised manufacturing practices around the globe were at the time merely the result of common-sense thinking in response to TMC's need for economic survival.

The lean ethos of the TPS was echoed through the call for elimination of the seven forms of waste as identified at the time by TMC. According to Ohno (1988) the two supporting pillars of the TPS were Just-in-Time (JIT) production and automation. JIT called for parts to flow to the final assembly only when and in the quantities needed.

Ohno recognised that the prerequisites for JIT were small lots and a means of communication to control production called kanban. The term automation was used to describe the combination of empowered problem-solving employees and automated machines capable of mistake-proofing.

Small lot production and JIT provided Toyota the ability to offer product diversity whilst minimising capital tied up in inventories, warehouse space and unnecessary defects. Nonetheless, it soon became evident that the success of the TPS relied on close collaboration with suppliers a fact which triggered the dissemination of the TPS practices to companies outside Toyota. Whilst supplier manuals were produced in the early 1970s (Hines et al., 2004) the first academic article reporting on the TPS was published in English in 1977 (Shah and Ward, 2007).

However, it was not until the oil crisis in the fall of 1973 that interest in the Japanese manufacturing practices started to grow immensely. Whilst economies worldwide were collapsing, TMC was able to sustain its growth and earnings (Ohno, 1988). What followed was a period of growing awareness of the secrets of TMC which coincided with the publication of landmark publications (Sugimori et al., 1977; Shingo, 1981; Monden, 1983). The secrets of TMC were unveiled to the world. The term lean manufacturing was coined by Krafcik (1986) to describe the TPS in a seminal study aiming to report the findings of the NUMMI benchmarking project.

In order to shed more light on the full range of elements encompassed in the TPS, Womack et al. (1990) published the "Machine that changed the world" whereby the TPS was referred to as lean production. Responding to a notable piecemeal approach by practitioners and academics alike to understand the discernible lean potential, Womack and Jones (1994) argued that the latter could not be realised unless lean was seen as a philosophy adopted at the enterprise level. Papadopoulou and Özbayrak (2005) subscribed to this view by suggesting that leanness is an inherently dynamic and evolving philosophy calling for a holistic approach in its implementation.

Latest research confirms that other sectors including construction, healthcare and other service and process industries have adopted the principles of lean manufacturing in order to modernise and improve their performance (Abdulmalek and Rajgopal, 2007; Dickson et al., 2009; Sacks et al., 2010).

Leanness was born out of the need to survive the post World War II era. It expanded and evolved in response to subsequent financial crises and the appreciation of the universality of financial and performance challenges facing all sectors. In the current economic climate it is more crucial than ever that all industrial sectors including

manufacturing revisit the lean paradigm in order to sustain their long-term survival (Scherrer-Rathj et al., 2009; Mollenkopf et al., 2010).

## **1.2 Background to the main research themes**

### **1.2.1 Production Planning and Control**

Production Planning and Control (PPC) is concerned with satisfying customer demand for products with supply provided by limited manufacturing resources (Barnes, 2008). Kempf et al. (2000) argue that the efficacy of PPC is fundamental to the success of the manufacturing system as a whole. The importance of PPC is also advocated in (Shobrys and White, 2002) whereby decisions made in the framework of PPC are considered in terms of their large economic impact on production systems.

Mula et al. (2006) present a taxonomy of various PPC models specifically designed to respond to uncertainty. Their classification scheme includes hierarchical production planning which broadly supports planning decisions at different levels. A detailed overview of the hierarchical PPC model is presented by Gathier and Frazier (2002). Hierarchical PPC involves a disaggregating top-down approach spanning across three discrete levels.

Long-term capacity planning is performed at corporate level resulting in facilities, equipment and supplier plans which in turn determine the organisation's overall capacity. These plans require strategic level decisions with a timescale of several years. Aggregate planning involves rough medium-term plans allocating the available production capacity to projected demand.

Firm and forecast orders are used to compile detailed short-term production plans determining quantities of finished products at the master production scheduling level. Master Production Schedules (MPS) are used to generate detailed material requirement plans for dependent demand items. Combining information from each product's itemised list of components, namely the Bill-of-Materials (BOM) and up-to-date information on inventory status, material requirement plans determine the quantities and timing of materials and components required to ensure the MPS can be met. In a study conducted to ascertain the adoption of different material planning methods, Jonsson and Mattson (2002) verify the view that computerised Material Requirement Planning (MRP) systems and their latest generations Material Resources Planning (MRPII) and Enterprise Resource Planning (ERP) continue to be widely preferred and used in real-life production settings.

According to Stoop and Wiers (1996) material requirements are the necessary input for day-to-day scheduling carried out at shop-floor level. Scheduling primarily entails loading and sequencing decisions. Whilst the former aim to determine the exact timings (start/finish times) that jobs (operations) can be assigned for processing to available resources, sequencing seeks to establish the exact prioritisation order in which jobs assigned to the same machine (within a workstation) will be processed (Barnes, 2008). Groover (2008) considers scheduling in the wider context of shop-floor control which further encompasses order release driven by MRP and order progress concerned with tracking the status of orders and flow of Work-in-Progress (WIP) through the system.

Scheduling functions are subject to constraints imposed by the finite capacity of available resources, job routings and due dates. Scheduling problems seek to simultaneously optimise a number of objectives e.g. job-flow times, resource utilisation, machine set-up time, inventory costs etc. Pinedo and Chao (1999) consider the intrinsically complex nature of scheduling and describe the majority of scheduling problems as NP-hard which can only be solved to sub-optimality within acceptable computational time. In real-life applications, schedules are affected by unpredictable events which can cause deviations from planned performance. Disturbances affecting capacity levels are often the result of machine break-downs, operator sickness etc. whereas order variability is associated with order cancellation or rush orders. Inaccurate scheduling input including processing and set-up times is viewed as another form of disturbance (Stoop and Wiers, 1996). Variability is reported as the second main contributing factor to scheduling complexity (Stevenson, 2006).

### **1.2.2 Push versus pull production control**

The widespread adoption of MRP-based systems has not been problem-free. Early studies pointed out the limited capability of these systems to handle capacity limitations and coordinate the flow of goods between workstations and different sections of the shop-floor (Stoop and Wiers, 1996). MRP nervousness refers to its limited responsiveness to dynamic changes affecting scheduling at the shop-floor level and is one of its most commonly reported weaknesses (Benton and Shin, 1998). Nagendra and Das (1999) are emphatic about the failure of MRP and MRPII to bridge the gap between planning and control arguing these systems were built primarily for planning.

Despite the evolution of MRP, the disadvantages of software packages based on the MRP logic remained at the focal point of subsequent studies. Significant deviations between planned order release and receipt driven by MRP and actual timings as well

as the failure of MRP to schedule operations on exact workstations were attributed to the ineffective capacity planning performed by MRP (Ho and Chang, 2001). Moreover, the findings of Kumar and Meader (2002) support the premise that feeding MRP systems with projected in addition to firm demand inevitably leads to inflated orders thus compromising the system's inventory control performance.

The emergence of JIT gave further thrust to the MRP debate. Barnes (2008) presents JIT as a philosophy, planning and control system and an inventory control system. Viewed as a philosophy JIT encapsulates the lean ideals of waste elimination, continuous improvement and employee involvement. Promoted as an alternative to MRP, the JIT production and control system is centred on pull control.

Driven by the MPS, MRP-based systems push demand information and materials downstream the supply chain. By operating in this fashion, MRP can hardly adapt to any changes in demand or unexpected conditions on the shop-floor. In contrast to the inflexible push nature of MRP, adaptive JIT production systems rely on actual demand to pull finished products through the supply chain (Waller, 2003). In pull systems, customer orders trigger production at the assembly stage, which in turn pulls the necessary subassemblies and components from preceding workstations and so forth. In this manner, demand information propagates sequentially upstream the supply chain whereas materials flow in the opposite direction.

JIT pull production as conceived by TMC requires a visual control system to be in place to facilitate communication and coordination between workstations. In their early forms these systems were linked to kanbans i.e. cards affixed to containers used to authorise production and transfer of parts. However, most recent applications of the system use alternative signals such as flags or designated floor space (Heizer and Render, 2004). In its simplest form a kanban system uses a single card type to authorise production. Dual-card systems use different types of cards to authorise withdrawal and production of parts (Japan Management Association, 1989). Visual control signals ensure the right quantity of parts is produced only when needed. Nevertheless, the system's success in eliminating work-in-progress and controlling inventory levels is further attributed to the fundamental JIT principle of small lots (Shingo, 1981).

Whilst the visual control system plays a key role in synchronising production between workstations, the effective operation of pull systems also requires levelled schedules. In JIT pull systems, level capacity loading is achieved by converting the projected customer demand of the MPS into a level rate-based mixed-model schedule (Stevenson, 2006). Rate-based schedules can be adjusted to buffer against demand variations (Feld, 2001).

Compared to MRP systems, JIT production planning and control appears to have additional merits in terms of capacity planning, coordination and control of material flow and inventory management. Whilst the driving force behind this is primarily pull production, the latter also enables JIT systems to be flexible and adaptive to variability. These comparisons have polarised proponents of the JIT and MRP systems on issues concerning their superiority and need for widespread adoption (Standard and Davis, 1999; Gupta, 2002).

### **1.2.3 Repetitive versus non-repetitive manufacturing**

From the early years of its existence, JIT was advocated as a production system suitable for repetitive manufacturing (Benton and Shin, 1998; Akturk and Erhun, 1999). According to the Association for Operations Management, repetitive manufacturing *“is the repeated production of the same discrete products or families of products. Repetitive methodology minimizes setups, inventory, and manufacturing lead times by using production lines, assembly lines, or cells. Work orders are no longer necessary; production scheduling and control is based on production rates. Products may be standard or assembled from modules”* (Cox and Blackstone, 1998 cited in Johnson and Malucci, 1999, p.12).

A typical example of a repetitive manufacturing setting is an automobile production facility similar to that in which the JIT system was pioneered. Repetitive processing is employed when production volumes are high compared to relatively low product variety (Stevenson, 2006). Flow-line layouts comprising workstations arranged in sequence are the shop-floor configurations found in these systems. In flow-shops, workstations must be in close proximity and tightly interlinked to facilitate a smooth and uninterrupted flow of WIP parts. This type of shop-floor configuration allows little variation between products and although mixed-model production lines can be used, production output is fairly standardised (Wild, 1989).

The production settings most commonly associated with non-repetitive manufacturing are job-shops. Job-shops employ functional layouts in which identical or similar general purpose machines are grouped together and laid out on designated sections of the shop-floor. These non-serial systems can accommodate the production of high product mixes in small volumes and therefore must be designed to allow flexibility and diversification (Pinedo and Chao, 1999). Products fabricated in job-shop settings are complex involving jumbled process routings. As a result, processing at workstations is

intermittent and requires equipment to be operated by multi-skilled workforce (Groover, 2008).

### **1.3 Research Rationale**

The reason for solely associating lean scheduling with repetitive manufacturing systems is twofold. Firstly, lean thinking was introduced in repetitive flow-shop environments and initial implementations were mainly attempted in similar manufacturing settings (White and Prybutok, 2001; Hallgren and Olhager, 2009). Secondly, key lean scheduling enablers including JIT and the kanban system were developed in line with specific design and functional characteristics of repetitive production environments (Huang and Kusiak, 1996; Plenert, 1999).

Motivated by the success of Japanese manufacturing practices and specifically that of the kanban system, Krajewski et al. (1987) investigated the impact of the type of production environment on the effectiveness of the adopted production scheduling system. Their study considered a number of diverse plant types in the United States (US) and concluded that key to the success of the scheduling system are settings inherent to the production environment rather than the selected scheduling system itself.

Several works challenged the applicability and performance of lean scheduling in non-repetitive production systems. Singh and Brar (1992) describe JIT as a closed loop production control system which relies on feedback from closely interacting stages. Frequent variations in product variety and volume coupled with the diverse process routings of manufactured products in non-repetitive manufacturing are identified as the main factors hindering the application of JIT in these systems. Andijani (1997) suggests that synchronisation and coordination in pull production systems can only be achieved if the kanban discipline is exercised on closely connected and interdependent serial workstations.

Connected and balanced flow processes are recognised as another important precondition for the successful implementation of JIT by Porter et al. (1999). This view is further supported by Marek et al. (2001) who suggest that pull production controlled by kanbans requires steady part flow in fixed routings. They further argue that production in high volumes contradicts the fundamental principle and JIT performance objective of WIP minimisation. The importance of flow in JIT systems is also stressed by Slack et al. (2010) who identify tightly linked production stages as an effective way for reducing inventory build up.



Nevertheless, the successful adoption of lean scheduling tools and the discernible benefits arising from their implementation in repetitive production environments (Zhu and Meredith, 1995) provided the inspiration for the first studies to extend lean scheduling in non-repetitive manufacturing systems. According to Geraghty and Heavey (2005) a review of the relevant literature published in the last 20 years shows that these investigations were undertaken in two different directions. The first concerns research carried out with the aim to create new or combine existing pull control strategies and enhance the benefits of pull control in non-repetitive systems. On the other hand, researchers experimented with the integration of MRP and JIT so as to combine the synergistic benefits that hybrid systems can bring to non-repetitive production environments.

Chang and Yih (1994) investigate the use of a modified Kanban system in non-repetitive production environments. Under the proposed Generic Kanban System (GKS) a fixed number of kanbans is placed at each workstation and can be used for any product type processed by the system. Generic kanbans allow a job to be released to the shop-floor only if there is a kanban available for it at each workstation. Interestingly, the proposed kanban system is tested on a production line comprising three workstations and clearly the non-repetitive nature of this setting is solely linked to variable demand and processing times. The serious limitation of this work is pointed out in their recommendations for further research by suggesting that more research is required on the implementation and testing of the system in a job-shop with diverse routings.

The implementation of JIT in non-sequential production environments is considered by Levasseur and Storch (1996). In the context of their work, the term non-sequential refers to manufacturing settings in which parts can be routed to any operation within the same cell. A dual kanban card system is proposed and tested in the case of six operations performed on 10 machines. In addition to the obvious small scale of their simulation model, kanbans are used to balance uneven workloads caused by variable processing times rather than trigger production based on real customer demand. Therefore, the proposed kanban system is effectively a modified version of a traditional push system.

Unlike push production, a pull system can operate in various alternative modes. Moreover, a number of associated strategies have been developed to exercise pull production control in the framework of JIT. Although three major pull production control policies dominate the literature, in particular, Kanban, Basestock and Constant Work-in-Progress (CONWIP) many variations of these are also reported (Liberopoulos and

Dallery, 2000). The difference between various pull production control strategies lies in the mechanics of flow coordination and control.

Similarly to other hybrid systems, CONWIP is differentiated from pure pull control (Spearman and Zazanis, 1992 cited in Beamon and Bermudo, 2000). Under CONWIP, pull control is implemented only at the two gates of the system, i.e. system entrance and exit and push control is exercised in all intermediate production stages. A fixed number of cards is used to maintain a constant level of WIP in the system at any time. A job can be released into the system only if there is a card available for it (Huang et al., 1998). Despite certain operational and performance limitations of the system, Framinan et al. (2003) describe CONWIP as a flexible yet robust pull mechanism that can be easily implemented in a variety of manufacturing environments including job-shops.

CONWIP control is proposed by Luh et al. (2000) as an effective way to control WIP levels in job-shops. The proposed synergistic methodology utilises Lagrangian multipliers to relax the capacity constraints and allow the problem to be decomposed into smaller-scale sub-problems. The latter are then solved using backward dynamic programming while heuristic methods ensure the generation of feasible solutions. Ryan et al. (2000) present an extension of CONWIP control to multiple-product job-shops with diverse routings. Their study mainly focuses on the determination of the overall WIP and further its allocation to the product mix to achieve a required service level. A throughput-driven heuristic is proposed and tested in the case of an open queuing network model. Numeric results point out a sensitive trade-off between service level and throughput. An alternative approach to determine individual card counts in CONWIP controlled job-shops is presented by Ryan and Choobineh (2003). A non-linear mathematical program is proposed that bounds throughput in the CONWIP controlled job-shop modelled as single chain multiple class closed queuing network.

An extensive review of pull control mechanisms and hybrid systems adopted in the case of non-repetitive production environments is presented by Geraghty and Heavey (2005). They use simulation in order to carry out a comparative study of Kanban, CONWIP, Hybrid Kanban/CONWIP, Basestock and Extended Kanban and analyse their performance in terms of service level and WIP. Nonetheless, their simulation does not consider a non-repetitive environment. On the contrary, they use a model previously developed by Hodgson and Wang (1991) which is based on a five-stage parallel/serial line used for the fabrication of two components of the same product.

Özbayrak et al. (2006) study the effects of pull control policies on the performance of a Small-to-Medium Enterprise (SME) operating within a Make-to-Order (MTO) supply chain. The job-shop system under investigation uses seven different machines to handle the fabrication of ten different components feeding the final assembly stages downstream the supply chain. Simulation is employed to model the system's operation under traditional push control and pull control introduced in two alternative modes, i.e. tight pull and relaxed CONWIP. Papadopoulou and Mousavi (2007a) employ agent-based simulation to apply CONWIP control in a non-repetitive production environment. The modelled system is a dynamic job-shop with stochastic order arrivals and processing times. Their study is mainly concerned with the impact of various dispatching rules on system performance in terms of a number of time, due date and work-in-progress related metrics.

One of the first attempts to integrate MRP and JIT in a job-shop environment is presented by Huq and Huq (1994). MPS quantities generated by MRP were used to set the container size of a single-card kanban system thus allowing MRP to be embedded into JIT used to control the dispatching of jobs in a hypothetical job-shop. Their study seeks to establish levels of processing times, set-up times, load variations and machine breakdowns which validate the selection of the embedded system and places little emphasis on the shop-floor control aspects of the latter.

Razmi et al. (1998) give further support to the view that as push and pull systems were designed to operate under specific conditions within certain manufacturing environments, they are not applicable to different settings unless certain modifications are made. Recognising the shortcomings of previous research, they develop a computer-based model to study the effects of interacting factors pertaining to different environments on the suitability of different push, pull and hybrid models. They suggest that hybrid systems in job-shops combine the manual control offered by pull systems with the long-term planning capabilities of MRP. A kanban-based shop-floor control system which can be used to extend the planning capabilities of MRP in non-serial manufacturing environments is proposed by Nagendra and Das (1999). The three modular components of the system are tested under a variety of scenarios and the study concludes by suggesting that further research is required in order to make the hybrid MRP/JIT system generally applicable to various production scenarios.

Counter to common perception, Ho and Chang (2001) argue that the product explosion logic of MRP renders it a pull system at materials planning level whilst it remains a push system at the shop-floor control level. On this basis, they propose the integration of MRP and JIT into a hybrid system which uses MPS demand to trigger production at

the last workstation whilst heuristics embedded in the system schedule operations at the shop-floor by pulling components from preceding operations. The hybrid system is used to minimise total production costs whilst meeting due dates in the case of multi-stage production-inventory systems. Another important limitation of their work is that it does not consider the impact of the planning horizon on the performance of the system. Finally, it is unclear how MPS demand can pull production at the final operation by using solely real customer demand data thus delivering one of the key benefits of pull production.

Depending on the levels in which integration is performed, hybrid control models are classified as either Vertically Integrated Hybrid Systems (VIHS) or Horizontally Integrated Hybrid Systems (HIHS) (Kochran and Kim, 1998 cited in Geraghty and Heavey, 2005, p. 440). In hybrid models embedding MRP into JIT, the synergy concerns systems operating at two different levels i.e. material requirements planning and shop-floor respectively and therefore the integration performed is vertical. Conversely, hybrids developed from combinations of push/pull control mechanisms at the shop-floor level result in horizontal integration.

Geraghty and Heavey (2005) argue that the limited adoption and practical use of VIHS is due to the complexity of effectively coordinating the use of MRP calculations at each production stage. Clearly, this complexity increases exponentially in the case of dynamic large-scale real-life job-shops. Moreover, the review of previous research investigating the extension of leanness to the scheduling functions of non-repetitive production environments by adopting HVIS manifests that despite the applicability of hybrid control strategies, reported implementations of pure pull control are limited.

In addition to research investigating the extension of lean scheduling in non-repetitive environments by studying the integration either of MRP and JIT or various pull control mechanisms with the aim to develop more suitable hybrids, the review of the relevant literature reveals a significant volume of research which followed an alternative approach. This includes studies promoting part grouping and layout reconfigurations jointly or in isolation with the aim to increase the system's manufacturing repetitiveness and facilitate the introduction of lean scheduling tools.

The integration of MRP and JIT, the determination of a rate per day schedule and the implementation of back flushing are a few of the JIT enablers introduced by Sandras (1985) in reconfigured job-shop to support its lean transformation. Job-shop reconfiguration into cellular layouts and flexible manufacturing systems was further proposed by Kelleher (1986). Gravel and Price (1988) consider the adaptation of the

JIT and kanban systems to a small firm operating as a job-shop. They conduct a pilot study which simulates the assembly of one product and suggest that the shop-floor layout was revised to enable the introduction of the new methods. Their conclusions identify a number of modelling and practical issues that need to be resolved before their findings can be generalised. Physical layout changes to improve problem visibility and work-flow were carried out by Martin-Vega et al. (1989) allowing the introduction of JIT in the case of a wafer fabrication facility. Their study reported significant reductions in cycle times.

The creation of a virtual flow-shop within the existing job-shop of a medium-sized make-to-order manufacturing company is considered by Lee et al. (1994). Jobs with high production volumes and standard routings are separated from low volume jobs with diverse routings. The virtual flow-shop controlled by a hybrid push/pull system is dedicated to the processing of high volume jobs. Stockton and Lindley (1994) develop an alternative method to Group Technology (GT) for indentifying product families and arranging equipment within cells in a High Variety low Volume (HVLV) environment. The proposed process sequence cells are controlled by a similar system which uses MRP to push materials through successive cells whilst intra-cell flow and inventory build up is controlled by kanbans.

The hybrid push/pull system proposed in the aforementioned two studies is further employed by Li and Barnes (2000) to assess the impact of shop-floor layout on the performance of kanbans. Their study experiments with a job-shop environment processing 19 parts on 28 machines of 12 types which is configured in a functional or cellular layout. It is concluded that conversion to cells should be preferred if set-up time reduction is significant. Li (2003) employs simulation to study the effects of core JIT supporting concepts when push and pull control is implemented. The study suggests that the adoption of pull production should be integrated with cellular manufacturing, one piece production and conveyance and tailored material handling equipment supporting one piece material flow. Simulation results confirm the author's initial expectation that the full potential and benefits of JIT can be only be realized if the adoption of pull production in job-shops is coordinated with the application of key JIT supporting elements.

Sharing the view that pull control is not well-suited for job-shops, Framinan and Ruiz-Usano (2002) study the transformation of a job-shop into a flow-shop layout by adding duplicate machines and re-arranging the existing. They use linear programming to minimise machine investment costs. A different approach is adopted by Özbayrak and Papadopoulou (2004) who investigate the extension of lean scheduling and pull

production control in high variety low volume systems where no prior shop-floor reconfiguration is performed. Part flow analysis is proposed instead in order to group parts based on process route commonality and increase the system's degree of repetitiveness.

From the above, it can be inferred that research studies investigating the adoption of JIT and pull control in reconfigured job-shops have only been performed in small-scale/complexity theoretical models and therefore demonstrate limited robustness and scalability. They are also subject to various limitations which concern unresolved modelling issues and the cost and time implications of their adoption in realistic scenarios.

#### **1.4 Research aim and objectives**

The review of the relevant literature presented in previous sections, points to a significant gap in current research. This calls for further investigation of the transferability of lean scheduling into non-repetitive production environments. On this basis, this research seeks to explore the extension of lean scheduling to non-repetitive manufacturing environments where no prior shop-floor reconfiguration or planning and control systems integration (vertical or horizontal) is performed. A novel approach employing agent-based simulation is proposed to introduce lean scheduling and pull control to large-scale dynamic job-shops.

This thesis poses the following two research questions:

Research Question 1: Is the application of pull production control to job-shops possible without prior shop-floor reconfiguration and/or horizontal/vertical integration of planning and control systems?

Research Question 2: Does the direct application of pull production control to job-shops improve their performance?

In seeking to provide answers to these research questions, this thesis aims to achieve the following objectives:

1. Develop in-depth understanding and knowledge of the lean manufacturing paradigm, its evolution and associated components.
2. Investigate lean scheduling and pull production control mechanisms and develop strong insight into their operating principles and differences.

3. Design a conceptual model of a dynamic job-shop scheduling system capable to operate under push and pull production control.
4. Carry out comparative analysis of alternative simulation technologies calling attention to the superiority of agent-based simulation in handling the modelling complexities of pragmatic production scheduling systems.
5. Develop a fully specified job-shop scheduling model.
6. Critically review and evaluate previous applications of simulation modelling in the scheduling and control of lean job-shops.
7. Design an agent-based architecture to implement the model and simulate its operation. Build suitable infrastructure to allow the system to be robust, scalable, reconfigurable and adaptable to different scenarios.
8. Assess and compare the performance of the job-shop scheduling system under push and pull production control. Validate the results of the agent-based scheduling model by carrying out a wide range of static and dynamic simulation experiments.
9. Verify the practical value of the developed agent-based lean scheduling system by modelling the conversion of a real-life job-shop operating as a push system into a lean job-shop controlled by pull control mechanisms.

## **1.5 Research boundaries and limitations**

This research is concerned with the short-term scheduling and control of machine operations performed at the shop-floor level. Long-term production planning performed at a higher level provides the main input for scheduling in the form of job lists with specific release times and due dates. The main functions performed in the framework of scheduling include the allocation and subsequently sequencing of job operations on machines suitable to perform their processing. Production control mainly deals with the coordination of flow of operations between workstations. As discussed in previous sections, both push and pull production control will be considered.

Scheduling decisions aim to optimise system performance in terms of a number of due date, flow time, WIP and utilisation metrics. Scheduling functions are complex, as they seek to balance the trade-offs between conflicting objectives under a number of constraints. In the framework of this research, dynamic scheduling is considered as

various scheduling parameters including processing times, arrival of rush orders, cancellation of jobs and machine breakdowns are subject to uncertainty.

The context in which production scheduling and control are considered is confined to non-repetitive manufacturing systems, in particular non-serial job-shops capable of processing a variety of jobs with diverse process routings. Job-shops employ layout configurations where multi-purpose machines are grouped together in disconnected areas of the shop-floor.

Areas which are **not** within the scope of this research are listed below:

- Capacity and material requirements planning
- Material handling, transportation and storage systems
- Integration of material planning, scheduling and control systems
- Layout reconfigurations of non-repetitive manufacturing systems
- Development of new pull control policies
- Detailed design issues of Kanban and other pull control mechanisms
- Development of new agent-based simulation software

## 1.6 Thesis structure

Further to this introduction, the remainder of the thesis is organised in six further chapters. A brief overview of each chapter is presented below.

**Chapter 2** presents a thorough review and appraisal of the existing literature on lean manufacturing. The concept of lean manufacturing is analysed, followed by a review of its evolution and key implementation enablers. Particular emphasis is placed on its scheduling and control tools and techniques. Common misconceptions among academic and practitioner circles related to its transferability to non-repetitive manufacturing systems are also debated from a critical viewpoint.

**Chapter 3** reviews background information on production scheduling and control theory. Scheduling and control functions performed in the context of a broader PPC hierarchy are analysed and linked to the main inputs, outputs and constraints of the scheduling process. The highly complex and combinatorial nature of scheduling is also considered. The principles and associated tools of lean scheduling and control are analysed and emphasis is placed on the latest developments in the area of pull



production control. This chapter presents a conceptual model of a job-shop scheduling system and identifies the pull control mechanisms that will be applied to non-repetitive job-shops using simulation.

**Chapter 4** adopts a deductive approach to identify a suitable modelling approach for formulating answers to the research questions of this thesis. Agent-based simulation is compared to conventional simulation techniques. The analysis endorses the suitability of agent-based simulation for modelling complex heterogeneous and distributed systems. The chapter further provides an extensive review of previous implementations of lean manufacturing to the scheduling and control functions of non-repetitive production systems. Related research investigating the introduction of pull control to jobs-shops is critically appraised, highlighting key limitations in the work of other researchers.

**Chapter 5** develops the conceptual job-shop scheduling system presented in chapter 3 into a fully specified model. The infrastructural and behavioural components of the model are presented in detail and particular emphasis is placed on design features introduced to allow the modelled job-shop to operate under pull production control. This is followed by the representation of the model as a multi-agent system. The chapter presents the functions and interfaces of the various agent types and provides a brief overview of the development platform used to implement the designed agent-based scheduling system. The chapter concludes by discussing the verification of the agent-based scheduling system using small-scale problems.

**Chapter 6** provides details of a wide range of experimentation scenarios designed to test the performance of the modelled job-shop scheduling system under push and pull control. A case-based approach is also followed to test and assess the application of pull production control to a real-life job-shop. The results of the simulation experiments and industrial case-study are presented and extensively analysed in this chapter.

**Chapter 7** initially presents a re-statement of the rationale and aim of this thesis. Following an overview of the main arguments derived from the primary and secondary research presented in previous chapters, the research questions are revisited and conclusions are drawn reinforcing the contribution of this thesis. The chapter concludes by presenting recommendations for further extensions of this research.

The organisation and flow of the chapters of this thesis is diagrammatically illustrated in Figure 1.1. This associates each chapter with the research objectives it seeks to achieve and categorises the research carried out in each chapter into secondary and primary.

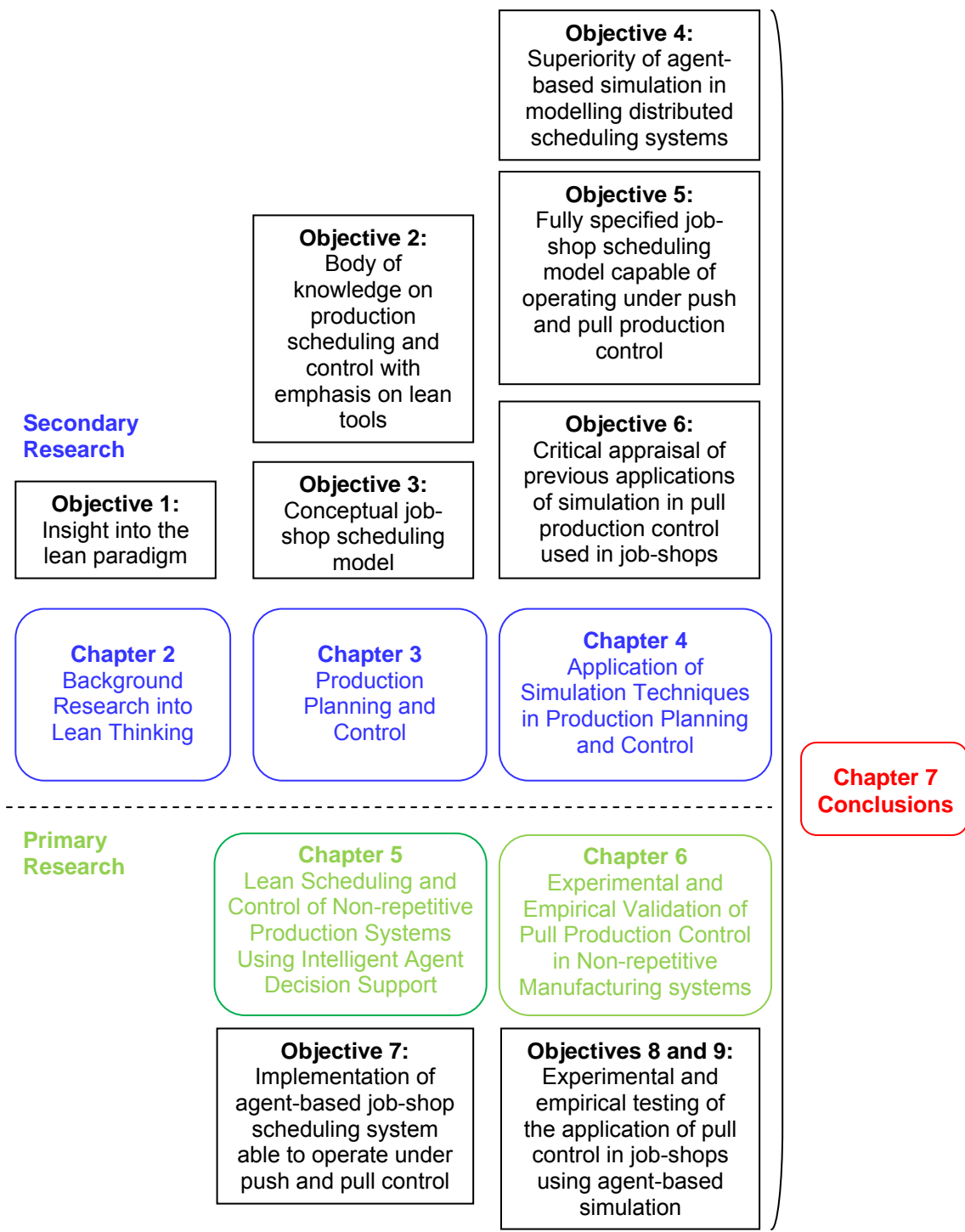


Figure 1.1 Outline of thesis

## **2 Background research into lean thinking**

Lean manufacturing (LM) is advocated as a world-class manufacturing paradigm. Although it dates back to the post World War 2 era LM continues to receive the undiminished attention of academic research and managerial practice. This chapter sets to explore the origins, scope as well as fundamental principles and practices encompassed within LM in an attempt to expose the reasons behind its overwhelming success.

LM is a term introduced in the late 1980's to describe a production system conceived by the Japanese car manufacturer Toyota. Leanness refers to the continuous pursuit of waste elimination. This aim is equivalent to the maximisation of value measured as the ability to manufacture a product that meets the needs of the customer by utilising manufacturing resources efficiently.

In order to achieve these strategic objectives, LM encapsulates several tools and techniques which need to be employed at the operational level. The engine behind LM is considered to be JIT production. This in turn is heavily dependent on other LM tools which provide the necessary infrastructure for its successful operation.

Section 2.1 investigates the origins of LM and seeks to analyse the contextual factors which led to the conception of its forerunner, i.e. the TPS. The TPS is analysed in relation to its main cornerstones with particular emphasis on JIT. The diffusion of TPS initially to Toyota's supply chain and other Japanese manufacturers and further to U.S. manufacturers is considered in section 2.2. This also reviews influential publications which facilitated the dissemination of the TPS. The historical overview of leanness continues in section 2.3 which discusses the emergence of LM. Section 2.4 reviews the lean enterprise model which constitutes the most contemporary form of leanness. Lean implementation issues are considered in section 2.5 which discusses LM practices and tools and their classification. The benefits resulting from the adoption of LM are presented in section 2.6. LM is compared to other rival manufacturing paradigms in section 2.7 which also evaluates its universality and suitability for all types of manufacturing systems. Section 2.8 reviews empirical research which showcases recent successful applications of LM. The conclusions of this chapter are presented in section 2.9.

## **2.1 Origins in Japanese manufacturing**

The Japanese origins of lean thinking are well-documented in the literature. However, contrary to common perception the essence of LM was not conceived in Toyota's automotive business. The first seminal publications disseminating the secrets of TPS to the world (Cusumano, 1985; Ohno, 1988) suggest that it was a single innovation in Toyota's weaving business that provided the original inspiration for the development of TPS. This invention concerned the automatic loom developed by entrepreneur and later founder of TMC Sakichi Toyoda.

### **2.1.1 The Toyota Production System (TPS)**

After the Toyoda Spinning and Weaving Company dissolved, one of its engineers Taiichi Ohno joined TMC in 1943, a time of serious economic hardship for the company. Ohno (1988) explains that adopting the mass production principles of the Ford Production System (FPS) was simply prohibitive. Mass production allowed little margin for product diversity demanded by the post-war domestic automotive market. More importantly, the FPS promoted production in large batches which in the case of the TMC required further investment in new production facilities. However, the company was facing capital shortages and mass production could only lead to further increase of its large inventory of unsold cars. Reichhart and Holweg (2007) describe the TPS as a hybrid system conceptualised by Toyota by embedding small-batch production techniques into Ford's mass production model.

Cusumano (1985) maintains that what was later described as one of the most important breakthroughs in the history of manufacturing was in fact, merely a common-sense approach to the challenges the company faced at the time. Inspired by the auto-activated weaving machine attributed to Sakichi Toyoda, Ohno (1988) developed the concept of automaton, which later became one of the two fundamental pillars of TPS. The automatic loom was equipped with a device allowing it to distinguish between normal and abnormal operating conditions and causing it to stop instantly if a warp or weft thread broke thus preventing machine time and material loss.

Cost minimisation through waste elimination in tandem with full appreciation/utilisation of employees (Sugimori et al., 1977) and continuous improvement (Shingo, 1981) were the three overriding objectives behind TPS. Ohno (1988) argued that producing zero waste can lead to optimum efficiency. Furthermore, waste is inversely related to value, another fundamental concept in the lean thinking. Removing waste from the manufacturing process can thus increase product value as perceived by the customer

(Womack and Jones, 1996). According to Ohno (1988) manufacturing waste exists in seven forms: overproduction; waiting; transportation; over-processing; inventory; movement; defective parts and products.

Inventory is a common form of waste in production systems. Both WIP and Finished Goods Inventory (FGI) result in tied up capital and storage space. However, this explains only partially the emphasis placed on its reduction within the TPS. Following conventional thinking, production systems often use inventory as buffer against problems in the production process and changes in demand. Nevertheless, through the prism of TPS this approach is wasteful and counter-productive as it conceals problems and hinders continuous improvement. A common analogy used to describe this relationship presents inventory as the water level in a lake and production problems as rocks hidden in the water (Hay, 1988). In line with this analogy, the objective of inventory minimisation equates to lowering the level of water and exposing problems. Problem identification and solving have been advocated as essential attributes of the TPS as they enhance process knowledge and allow cost-effective continuous improvement (Spear, 2002).

Shingo (1981) suggested that in the framework of the TPS there are two forms of overproduction. Quantitative overproduction creates *more* products than actually needed. On the other hand, early overproduction creates products *before* they are needed. Early overproduction inevitably creates FGI which must be stored and managed until it is finally sold to the customer. The main method to eliminate early overproduction is in fact the second main pillar of the TPS, JIT production.

### **2.1.2 Just-in-Time (JIT) production**

Timeliness of production is the essence of JIT. The initial inspiration for the development of JIT production came by studying the operation of American supermarkets. Supermarkets are retail units in which customers can purchase what they require at the time and quantity required. Ohno (1988) explains the analogy between a JIT production system and a supermarket. Each process in the production system is viewed as the customer whilst the role of the immediately preceding process is similar to that of a supermarket.

According to Sugimori et al. (1997, p.555) *"just-in-time production is a method whereby the production lead time is greatly shortened by maintaining the conformity to changes by having all processes produce the necessary parts at the necessary time and have on hand only the minimum stock necessary to hold the processes together"*. The above definition clearly presents JIT as a system coupling production planning and control

with inventory management. JIT production employs a set of intertwined practices to perform these two functions. Pull production, level scheduling and kanban control are primary JIT practices, which in turn are supported by several subsidiary techniques.

Pull production involves the withdrawal of parts by subsequent processes (Sugimori et al., 1977). Conventional production systems are push systems. They use information on projected demand and available inventory to determine production schedules which push WIP from the raw materials inventory to the final assembly regardless of the timing parts are required. In JIT systems, demand triggers production in the final assembly and is subsequently propagated to preceding processes thus determining the flow of subassemblies, components and raw materials within the system. In this manner, production at a given workstation is initiated only when its output is required by the succeeding workstation (Silver et al., 1998). In contrast to push production which is centrally controlled by MRP-based systems, pull production control is decentralised and implemented locally between closely linked workstations (Reichhart and Holweg, 2007).

Significant variations in the timings and quantities of parts withdrawn by subsequent processes would increase waste as peaks in demand could only be catered for by maintaining excessive inventory (Ohno, 1988). Such variations would hinder JIT production. Therefore, a levelled production schedule at the final assembly is the prerequisite for JIT production (Yavuz and Akçali, 2007). Levelled production schedules are determined by dividing the monthly product requirements of the master production schedule by the number of working days in a month to establish a stable daily production rate. In this fashion, the levelled final assembly schedule creates a relatively stable demand for subassemblies and components (Harrison, 1992).

Mixed modelling is a technique used to facilitate production levelling by balancing capacity and load at workstations (Shingo, 1981). The aim of mixed model production is to produce apart from a level output, the full mix of products repeatedly each day or other short interval (Vollmann et al., 1997). Establishing the takt time determined by the levelled daily production rate and ensuring that it does not exceed the cycle time i.e. the processing time in workstations is identified as one of the fundamental design rules to facilitate levelled mixed model production (Black, 2007).

Production levelling and mixed modelling generated the requirement to reduce production lot sizes as much as possible. Small lot sizes were described by Ohno (1988) as Toyota's main challenge to conventional mass production wisdom. Sugimori et al. (1977) present one piece production and conveyance as the second requirement

for JIT pull production. However, they explain that the aim in one piece production is in fact to approximate the condition in which each process can produce and stock as little as one piece at a time. The first attempts to implement small lot production proved strenuous as set up times for pressing dies were taking a number of hours to complete. Shingo (1981) explains how analysing internal and external set up elements paved the way for the development of the Single-Minute Exchange of Die (SMED) system allowing TMC to perform four-hour set ups in only three minutes.

In line with the definition provided by Sugimori et al. (1977) one objective of JIT is to shorten production lead times. Ohno (1988) suggested that establishing production flow by closely linking workstations can eliminate the waste of transporting and storing parts thus minimising lead time. However, according to Monden (1998; p. 64) waste between processes can be eliminated by synchronising the flow of parts. Process synchronisation or balancing can be achieved if every process finishes at the same pace with the average cycle time. Production flow also results in better coordination between workstations (Harrison, 1992).

Production flow and synchronization are prerequisites for the effective operation of the Kanban system. In its original form, the system used a fixed number of cards to authorise production and the movement of parts within pairs of consecutive processes. Each of these cards was attached to a container of parts, providing information on the type of parts, production/transfer quantity, destination etc. (Hay, 1988). Cards used to authorise the processing and transfer of parts in line with the principles of JIT therefore control production and inventory levels within the system. Kanban is the operating method for JIT production, also quoted as the “autonomic nerve” of JIT systems (Ohno, 1988; p. 29).

## **2.2 Dissemination of the TPS**

The TPS was hailed as one of the greatest success stories in the history of manufacturing and served as a production model embraced by many companies worldwide. However, according to Spear and Bowen (1999) despite TMC’s extraordinary openness about the system and numerous visits of the company’s plants both in Japan and the U.S by business executives and operations managers from other firms, very few of these adoptions proved successful.

Research carried out by Towill (2007) demonstrates that 55% of U.S. manufacturers are still striving to introduce elements of the model into their production system. It is further pointed out that successful implementations of the system are often subject to

limitations, with enhanced inventory performance mainly concerning reduction in WIP levels, not FGI. Similar findings are reported by Swamidass (2007) suggesting that implementations of TPS in low performing U.S. firms do not always result in the desirable inventory reduction. New (2007, p. 3546) adds further support to these findings by pointing out that *“the popularity of the TPS as a subject of discussion and research seems only to be matched by the widespread inability of organizations to adopt and apply the ideas with anything like the success of Toyota”*.

Failed attempts to successfully assimilate the TPS were initially attributed to Japanese diligence and other social and cultural issues confined to Japan and the TMC (Spear and Bowen, 1999). Cusumano (1988) however, dismisses this argument by contrasting the superior performance of Japanese-run factories in the U.S. against that of other plants both in the U.S. and Japan. Spear and Bowen (1999) suggest that observers found the TPS difficult to decode due to the trodden nature of its various components as well as Toyota's own innovative practice of constantly challenging the system and improving the performance of its tools. The TPS evolution driven by its continuous improvement constituent and progress made on the lean learning curve have been identified as the main challenges faced by those attempting to disentangle the true essence of the system (Hines et al., 2004).

Pil and Fujimoto (2007) suggest that the TPS evolved significantly from the system originally envisioned by Ohno. The authors argue that *“the strength of production models lies not in understanding how tightly interwoven systems of practices interact in synergistic ways. Rather, it rests on the ability to leverage that understanding in a directed manner to identify novel changes in practice to meet evolving environmental demands”* (p. 3758). Their study highlights the importance of analysing the evolutionary trajectories of production systems in order to deepen our understanding of their operation and challenges associated with their implementation.

### **2.2.1 Extension to Japanese manufacturers and suppliers**

Small lot production implemented in the framework of JIT offered the advantages of reduced WIP levels and increased flexibility and responsiveness. However, reducing production lot sizes had implications on procurement and generated the requirement for synchronised small lot deliveries from suppliers (Shah and Ward, 2007). According to Ohno (1988) extending the TPS to suppliers was not an automatic process. In fact, following the introduction of the TPS to car engine manufacturing in the 1950s and the final vehicle assembly in the 1960s it was not until the early 1970s that the system was adopted by TMC's network of suppliers.



Apart from the obvious impact on the effective operation of the system, this development serves as an important landmark in the dissemination of the TPS. Holweg (2007) explains that although it is not possible to accurately pinpoint the time that TPS documentation appeared in the public domain, it is likely that the first formal documents on TPS were supplier manuals. These manuals were produced by the Purchasing Administration Department of the TMC around 1965 with the aim to introduce suppliers to the requirements of JIT delivery.

Schonberger (1982a, b) emphasised the importance of JIT purchasing and supplier integration. Insight into how TMC managed its supply chain is provided by Womack et al. (1990, p. 60-62). Suppliers were organised in functional tiers. First tier suppliers were effectively part of TMC's integrated team for product development whereas second tier suppliers were process engineering specialists responsible for supplying the first tier with individual components. TMC held significant fraction of the equity of these supplier organisations with the latter also having substantial cross-holdings. Partnering agreements formalised long-term relationships with suppliers who were encouraged to share not only information and best practice but also the financial benefits resulting from improvements.

González-Benito and Suárez-González (2001) present a JIT purchasing model comprising the following three conceptual levels: JIT purchasing philosophy, JIT purchasing techniques and JIT purchasing and delivery control system. Whilst JIT purchasing philosophy emphasises on the collaborative approach and partnering ethos of the lean supply chain, JIT purchasing techniques comprise tools to promote supplier communication and involvement. Finally, the JIT purchasing and delivery control system includes small lot and synchronised supplies, frequent deliveries and the use of standardised containers which are the main operating requirements of JIT supply deliveries. The Kanban system provided the mechanism for organising and controlling JIT deliveries. The use of kanbans removed inventory buffers across the supply chain encouraging early identification of setbacks and proactive problem-solving so that failures at various stages of the process could be avoided (Womack et al., 1990).

Apart from its extension to the TMC's supplier network, the TPS was fairly unnoticed by other Japanese manufacturers until the first oil crisis in 1973. Ohno (1988) explains that whilst the Japanese economy was badly hit and the majority of businesses suffered great financial losses during that economic downturn, the TMC sustained its growth arousing domestic interest in the TPS. Holweg (2007) suggests that the limited attention the system attracted until then was not due to a veil of secrecy covering

Toyota's powerful weapon but merely a result of the lack of documentation shedding light on the rapidly evolving production model.

The pressures of the volatile economic environment and by contrast Toyota's resilience led other Japanese automakers including Hino and Daihatsu to imitate the production model implemented by Toyota (Cusumano, 1988). Other reported cases include Kawasaki Heavy Industries (Hall, 1982 cited in Hallihan, 1997) and Mazda (Womack et al., 1990) which sought assistance directly from Toyota in order incorporate elements of the TPS into their plants. The Synchro-MRP system implemented at Yamaha Motor Company (De Toni et al., 1988) and the Kawasaki Production System (KPS) (Schonberger, 1982a) were descendants of the TPS developed at that time. Whilst the influence of the TPS on these systems is indisputable, critics of the TPS argue that many of the ideas developed by Toyota were the result of cross-fertilisation from practices and innovations conceived by Honda, Nissan and others (New, 2007).

### **2.2.2 Diffusion in western manufacturing**

The first complete overview of the TPS including the principles underpinning the model and its fundamental tools and practices appeared in the first edition of Ohno's Toyota Production System (1988) published in 1978. However, as the first edition of the book was written in Japanese its contribution to the dissemination of the system was limited. The article by Sugimori et al. (1977) is quoted as the first prominent work on the TPS published in English (Shah and Ward, 2007). According to New (2007) the work by Sugimori et al. represents a beacon in the TPS-inspired research.

Holweg (2007) suggests that the influential contribution of this paper is attributed to three main factors. In contrast to other publications which focused on specific tools of the system, Sugimori et al. review the main components of the TPS and explain how work systems can adapt to it. The knowledge is sourced from TMC production control managers with hands on experience in the implementation of the model. Most significantly, the work of Sugimori et al. compares the performance of car assembly plants in the U.S. and Europe against the Toyota benchmark and is thus the first manifestation of the superior performance and competitive advantage of Toyota over their western counterparts.

Nonetheless, the geographical dispersion of the TPS was very slow and the system attracted little attention until the second oil crisis in 1979. Harrison (1992, p. 14) explains that the ability of Japanese manufacturers to sustain their financial standing despite the turbulent economic environment sparked great interest among western

manufacturers in Toyota's success story. Xerox benchmarked the performance of its U.S. plants against its Japanese subsidiary Fuji Xerox whereas other firms including Ford, Chrysler and Mitsubishi sought to unravel the Japanese paradox by establishing joint ventures with Japanese car manufacturers.

In 1979, an initiative led by the American Production and Inventory Control Society (APICS) resulted in the formation of the Repetitive Manufacturing Group (RMG) which was tasked with the dissemination and promotion of the TPS to repetitive manufacturers (Schonberger, 1982a). Among the meetings and factory tours organised by the group, Holweg (2007) cites the visit to Kawasaki's motorcycle plant in Nebraska aiming to introduce the study group to the KPS derivative of the TPS. The lessons learned from similar tours of Japanese manufacturing plants are discussed by Hayes (1981). His findings are illuminating as they suggest that whilst western manufacturers sought to reclaim their place in their home and international markets by building the factory of the future, Toyota increased the performance gap by revisiting manufacturing fundamentals in order to improve the factory of the present.

Following the second oil crisis in 1979, a five-year research programme was launched to investigate the role of the automobile industry in the future of manufacturing. Renamed in its second phase as the International Motor Vehicle Program (IMVP) the research initially entitled "The Future of the Automobile" was an international project involving a large network of universities and industrial collaborators led by the Massachusetts Institute of Technology (MIT) (Holweg, 2007). The agenda of the first phase of the programme was shaped by two parallel formidable developments in the US. The rapid increase of Japanese car imports and the growing numbers of Japanese transplant facilities opening in North America (Womack et al., 1990).

Holweg (2007) explains that funding released into the second phase of the IMVP was aimed at shifting the focus of the programme from legislation and trade union relationships and agreements to operational issues justifying the notable performance gap between Western and Japanese manufacturing. In 1984, a joint venture between General Motors (GM) and Toyota was established involving the re-opening of GM's car assembly plant in Fremont, California which due to significant shrinking of its business stopped operating in 1982 (Womack et al., 1990, p. 82-85).

The re-opened plant was named New United Motor Manufacturing Inc. (NUMMI) and was used to produce Toyota passenger cars for the U.S. market. NUMMI adopted the TPS fully and whilst 80% of the workforce was formerly employed by GM, the senior management were all from Toyota. Three years after NUMMI began its operation, its

performance was analysed and compared with that of a Toyota plant in Japan and a GM plant in the U.S. using data collected by the MIT research engineer John Krafcik during his training in Toyota plants in Japan (Krafcik, 1986).

According to Shah and Ward (2007) NUMMI constitutes the first formal introduction of the TPS in the U.S. The authors acknowledge that the informal diffusion of the system in western manufacturing which had began earlier was impaired by the inability of managers to fully grasp the multifaceted and evolving nature of the system and was therefore only carried out in a piece-meal fashion.

Further to NUMMI, initiatives led by the Automotive Industry Action Group (AIAG) contributed to the adoption of the system by car manufacturers across the U.S. and Canada (Hallihan et al., 1997). By the mid 1980s implementations of the TPS were common beyond the automotive industry. Hewlett Packard (Sandras, 1985) and Black and Decker (Hay, 1988) were amongst the first implementers of the system in other manufacturing divisions. Research carried out by Voss and Robinson (1987) suggests that 57% of a surveyed sample of UK manufacturers had adopted or considered implemented elements of JIT in their production environments. The benefits resulting from pilot adoptions of the system in West Germany are reported by Wildemann (1988).

### **2.3 Emergence of lean thinking**

Lean is a term coined by Krafcik (1986) to describe a production model based on the principles of the TPS. Krafcik distinguishes between two types of production models which mark the pre and post-Ford eras in manufacturing. In this classification schema, the mass production system developed by Ford is the pure Fordism model typically employed by western manufacturers whereas the TPS is a production model with origins in Fordism and a Japanese flavour.

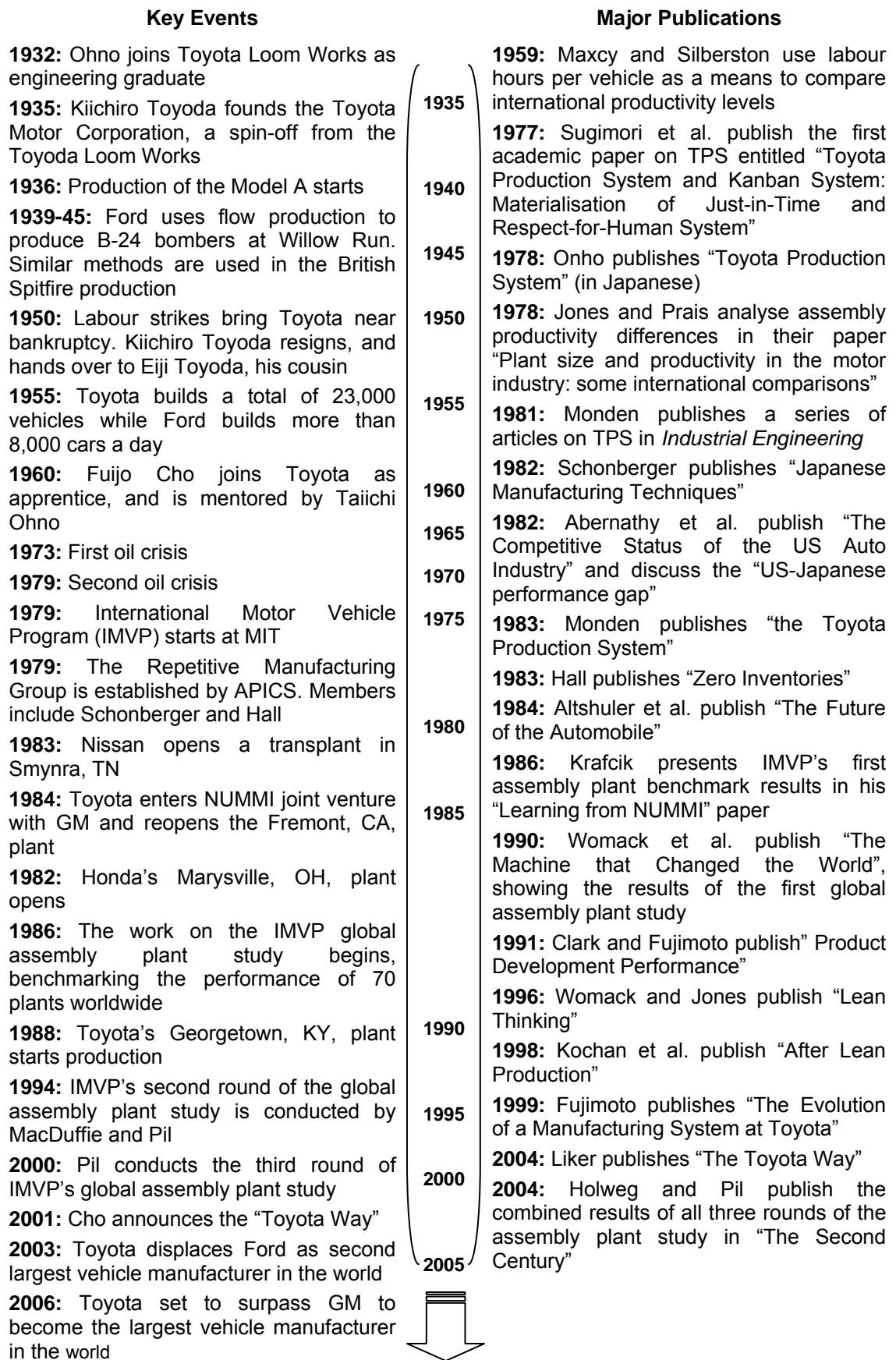
Comparing the two systems, Krafcik suggested that their main difference lies in the use of buffers. The pure Fordism model is a *buffered* model using high levels of inventory to cope with uncertainty and variability. In contrast, the TPS is a *bufferless Lean Production* (LP) system where *“inventory levels are kept at an absolute minimum so that costs could be shaved and quality problems quickly detected and solved; bufferless assembly lines assured continuous flow production; utility workers were conspicuous only in their absence from the payroll. If a worker was absent without notice, the team would fill in; repair areas were tiny as a result of the belief that quality should be achieved within the process, not within a rectification area”* (ibid, p. 45).

According to Hines et al. (2004) and Swamidass (2007), the term lean manufacturing was introduced by Womack et al. (1990, p. 13) to describe a production system which *“uses less of everything compared with mass production – half the human effort in the factory, half the manufacturing space, half the investment in tools, half the engineering hours to develop a new product in half the time. Also, it requires far less than half the needed inventory on site, results in many fewer defects, and produces a greater and ever growing variety of products”*. A definition of LM is also proposed by Liker (1998, p.481) referring to *“a philosophy that when implemented reduces the time from customer order to delivery by eliminating sources of waste in the production flow”*. Holweg (2007) presents landmark reports linked with some of the key milestones in the evolution of lean thinking already discussed above in the time line illustrated in Figure 2.1.

As the use of the lean terminology gained momentum it sparked controversy concerning the origins of the ideals encapsulated in LP. New (2007) argues that some audiences may view the emphasis placed by western followers on the lean features of the TPS as an attempt to undermine the Japanese-ness of the system. Reichhart and Holweg (2007, p.3701) are diametrically opposed to this view and suggest that the TPS is the ancestor of lean thinking. Reviewing the work of those credited with the introduction of lean terms, it is evident that there is sufficient acknowledgement of Toyota’s intellectual heritage. Womack et al. (1990, p. 68) suggest that through its extension from the final assembly to the supplier network and customer relations, Toyota’s LP emerged as a fully-fledged system.

According to Womack and Jones (2003), lean thinking can be distilled into the following five principles the synergy of which can eliminate waste:

1. The precise definition of value. Rethinking value from the perspective of the ultimate customer means defining value as the combination of product specifications and availability that meets the customer’s needs at a specific price. Failure to accurately specify value will result in the wrong product and create waste.
2. The identification of the value stream. The value stream is created by all the functions that need to be performed at three management levels, product development, information handling and product fabrication in order to create the end product. Removing activities that do not add value from the value stream, removes waste that the customer is unwilling to pay for.



**Source: Holweg (2007)**

**Figure 2.1 Landmark reports and key milestones in lean thinking**

3. The creation of flow. Once the value stream is fully mapped and non value-adding steps are eliminated, the departmentalised mentality of disconnected processes should give way to the re-organisation of value-adding activities so that information and materials can flow smoothly through the system without interruptions caused by batches, queues, breakdowns and defects.
4. The introduction of pull production. Contrary to conventional systems which push products from the raw materials to finished goods inventory, in pull production customers pull products as and when required from the factory. By applying the principles of value, value stream and flow, lead times can be reduced considerably allowing customers to pull products from the factory right away and thus creating a fairly stable demand.
5. The pursuit of perfection. No matter how lean processes are, in lean thinking there is always scope for further improvement. Waste elimination should be pursued through continuous, incremental and radical improvement of processes across the value stream involving everyone in the organisation.

Oliver (2007, p. 3726) suggests that over the years, various labels have been attached to lean principles including Japanese manufacturing, JIT or TPS. This view is shared by Lee and Jo (2007, p. 3666-3667) who argue that as TPS evolved from a set of waste-eliminating techniques to the post-Ford LP system, its multi-faceted nature and growing scope made the task of effectively and uniformly classifying it very difficult. The authors list several diverse descriptions of the system ranging from a simple goal, method, process or program to a belief, state of mind, strategy and philosophy.

Bhasin and Burcher (2006) found that adopters of LP often embrace it as a philosophy and use the tools and tactics encompassed in the system as the mechanisms to implement the lean way of thinking. Shah and Ward (2007) extend this view by suggesting that LP can be viewed from two different perspectives. From the philosophical viewpoint, it encapsulates the lean underlying principles and overriding goals whereas from the practical viewpoint it comprises a set of operational tools, techniques and practices that must be implemented to achieve the lean objectives.

Subscribing to this orientation, Scherrer-Rathje et al. (2009) recommend that in order to be a successful and sustainable programme in the long-run, the adoption of LP should incorporate two components. The operational/tactical component brings together all the lean tools and techniques whilst the strategic component comprises top management support, cultural change and employee involvement which ensure that the synergistic effect of the tools employed at the operational level is delivered.

A review of definitions/descriptions of LP, JIT and TPS sourced from several landmark publications is presented by Shah and Ward (2007). Their comparative analysis reveals significant similarities and overlapping in definitions used to describe these concepts. Most importantly, the authors are critical of the emphasis placed in many of these definitions on discrete tools rather than the system as a whole. In the light of this finding, they develop a conceptual definition which captures the integration of people and processes, internal and external stakeholders in which the strength of the system lies. They define LP as *“an integrated socio-technical system whose main objective is to eliminate waste by concurrently reducing or minimizing supplier, customer, and internal variability”* (ibid, p. 791).

Despite the obvious differences in their approaches, the findings of researchers exploring the true essence of LP converge at one point; the versatile but also inherently dynamic nature of the system. Papadopoulou and Özbayrak (2005) describe LP as a benchmark manufacturing paradigm undergoing a continuous evolution, the current state of which is expressed in the form of the lean enterprise. Bartezzagni (1999) sheds more light into the distinguishing features of production paradigms. He explains that paradigms unify general principles and criteria used to guide the design and management of production systems. In contrast to production models which are collections of manufacturing techniques considered optimal for a certain company context, manufacturing paradigms are universal standards applicable in different contexts.

## **2.4 Lean at the enterprise level**

Womack et al. (1990) presented the transition from mass to LP by dwelling on the systematic extension of the lean principles from the final assembly to product design and development, the supplier network and customer sales. Their approach made a two-fold contribution to the lean research. First of all, it hailed LP as a total operations management system. Secondly and perhaps most importantly, it laid the foundations for what constituted the next performance leap for lean adopters.

Global economic competition coupled with open market trade agreements and technological advancements in transportation and communication systems created a new array of challenges for contemporary manufacturers. Reliable and highly customised products are new customer priorities which require high levels of manufacturing flexibility, responsiveness and modernisation of product delivery methods. Unable to withstand these immense pressures alone, manufacturers



engaged in strategic alliances with partners across the supply chain in order to improve their performance and competitive advantage (Browne et al., 1995).

These partnerships led to the emergence of a novel business model, the extended enterprise. Extended enterprises transcend organisational barriers and seek to integrate all those functions performed at various stages of the whole product life cycle including procurement of materials, product development and engineering, fabrication and assembly, sales, distribution, after-sale customer services and finally, disposal and recycling (Jagdev and Browne, 1998).

Building an extended enterprise requires a thorough review of business-as-usual operations as well as a commitment to open communication, innovation and technological change by all allied partners. Advancements in Business Process Reengineering (BPR) facilitated most of these requirements and sparked further interest in the extended enterprise model (Childe, 1998). However, proponents of the model identified intra-enterprise integration of all the functions within an organisation as the main precondition for its further integration with external allies (Jagdev and Browne, 1998).

The LP model imposed a similar requirement for the holistic adoption of the lean mantra by all sections and departments within the organisation. Nevertheless, Womack and Jones (1994) argue that for lean adopters this is not the end of the road. A new improvement path can be created by integrating the breakthroughs of individual companies upstream and downstream the supply chain. According to the authors, the next leap in performance can be achieved by embracing the lean enterprise organisational model which they envision as *“a group of individuals, functions, and legally separate but operationally synchronised companies”* (ibid, p.93).

Jenner (1998) outlines the main principles governing the establishment and operation of extended enterprises. Building an enterprise involves substantial organisational restructuring so as to create closely linked flattened processes which facilitate coordination and flow of materials and information. The removal of middle management layers and the creation of multi-functional, self-managed teams enable the delegation of authority to lower echelons which have direct access to process-related problems and issues. Decision-making across the enterprise is subject to the process of amplification with different layers contributing to the assessment of choices with knowledge and information available at their level. Common focus and goals are identified as the key ingredient that holds enterprise partners together. Womack and

Jones (1994) propose that sharing the same business vision enables the group of allied companies to focus on maximising value for the customer.

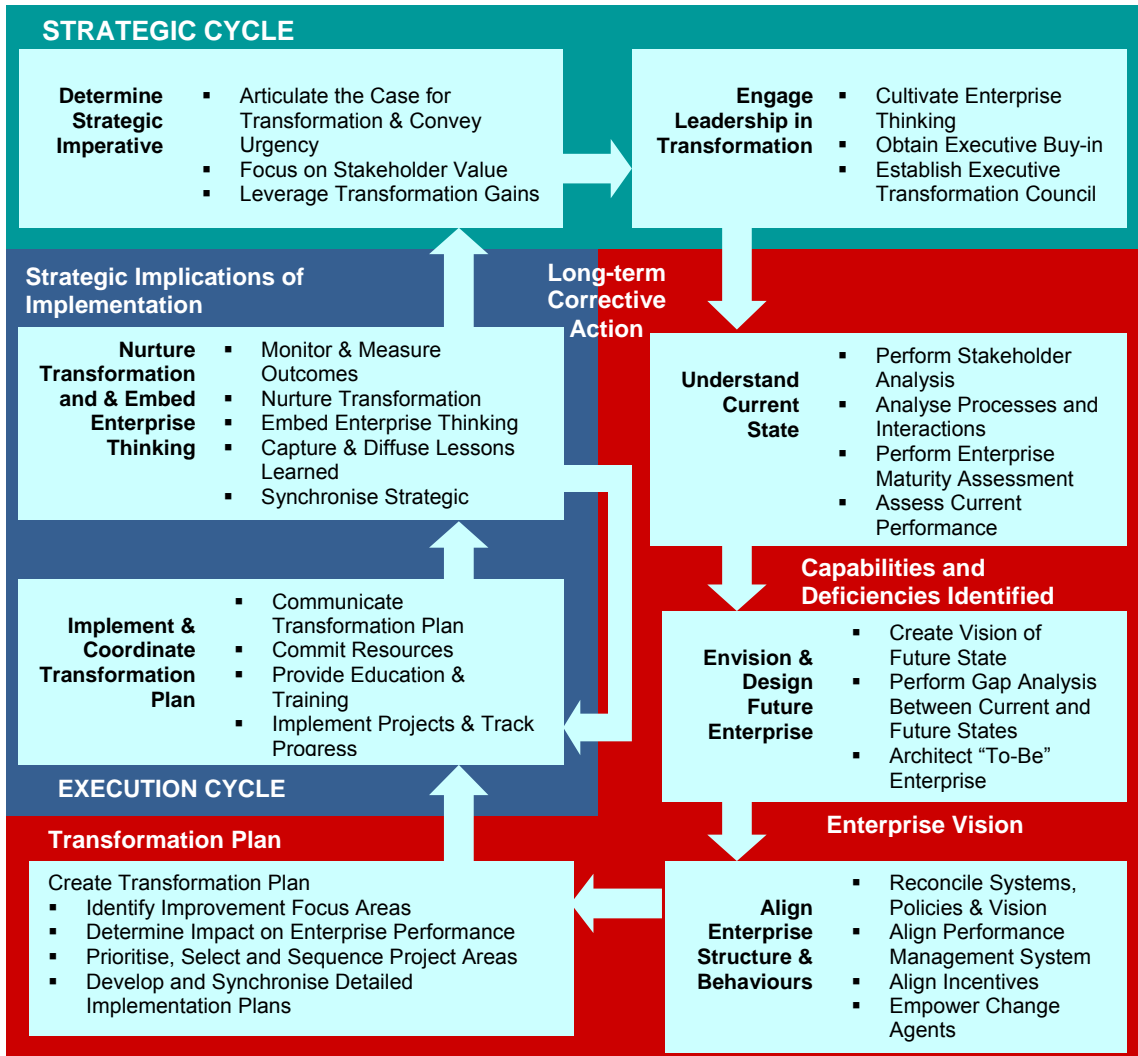
Womack and Jones (2003, p.277) recommend that the fundamental LP principles of value and value stream will need to be jointly revisited by allied partners so as to be elevated to the enterprise level. They proceed to suggest that value should be linked to the target cost that is, the price the customer is willing to pay for the product. The target cost will in turn determine the level of profit. Iterative re-examination of the value stream will remove waste, minimise cost and thus maximise each partner's share in the return-on-investment.

Having discussed what enterprises constitute, it is imperative to also clarify what they are not. Womack and Jones (1994) insist that despite being often compared to virtual corporations, lean enterprises differ dramatically from the former as they rely on long-term relationships between partners whereas the composition of virtual corporations is only temporary. Bititci et al. (2005) explain how extended enterprises differ from supply chains with the latter mainly focusing on maximising individual corporate goals whereas the former focus on maximising the performance and thus the competitive advantage of the overall extended enterprise.

Following a period of significant defence spending cuts, a similar programme to the IMVP was launched in the U.S. in 1993 focusing on the application of LP concepts in the field of aerospace. The Lean Aircraft Initiative (LAI) was organised as a consortium between the MIT and a range of collaborators including the US Air Force, the US Department of Defence and major US military aerospace contractors (LAI, 2012.). Its mission was to research, identify and disseminate best practice supporting the lean enterprise transformation which was deemed critical for the future prosperity of the sector.

From the early years of the consortium's existence, its main academic collaborator, the Lean Advancement Initiative (LAI) based at MIT, published a series of publications providing guidance on the implementation of lean techniques including manuals and self-assessment tools for lean adopters (Nightingale, 2009). A key reference model supporting the transformation to the Lean Enterprise is presented in Volume II of the "Transition-to-Lean Roadmap" guide (Bozdogan et al., 2000). The guide organises transformation activities into three basic cycles depending on whether associated decisions and actions are strategic or operational, they involve external or internal stakeholders and require short or long-term implementation.

Figure 2.2 illustrates a schematic representation of the most up-to-date version of the transformation roadmap (Nightingale et al., 2010). In this version, the Entry/Re-entry, Long Term and Short Term Cycles of the original model are depicted as the Strategic Cycle, the Planning Cycle and the Execution Cycle respectively.



Source: Nightingale et al. (2010)

Figure 2.2 LAI Lean enterprise transformation roadmap

The Entry/Re-entry cycle describes actions resulting from the strategic decision to adopt the lean paradigm and transform to a lean enterprise. These include the creation of a new vision, focusing on the lean principles and securing top management support and commitment. The Long Term Cycle is concerned with all the preliminary preparation and change of internal environment conditions the organisation should carry out before embarking on the more detailed planning and implementation of the transition. The main actions performed in this cycle concern mapping the value stream so as to eliminate non-value adding activities from all business systems of the

enterprise and internal restructuring in order to replace silos of authority with horizontally linked functions performed by empowered teams. The Short Term Cycle comprises the detailed actions that need to be implemented to actually achieve the transformation. These actions need to be monitored and re-evaluated so as to take corrective actions in line with the lessons learned or adjust to dynamic external conditions. The key feature of this cycle is the focus on continuous improvement, the introduction of lean systems and tools in addition to personnel training.

## **2.5 Lean implementation**

The upsurge of interest in Japanese management practices and their perceived benefits (Pegels, 1984; Voss, 1984; Cusumano, 1988) brought the issue of their implementation at the focal of point of the lean research. The link between implementation and the ability of firms to harness the benefits of manufacturing innovations invented in Japan is straightforward. Knowledge is key to correct implementation so first of all, companies need to be certain of “what exactly” they endeavour to apply (Mehra and Inman, 1992). Wafa and Yasin (1998) acknowledge that unless the implementation of the adopted system is successful the expected benefits will fail to materialise.

Investigating the implementation of JIT in manufacturing operations in the U.S., Mehra and Inman (1992) highlight the limitations of research in this area by recognising that findings are not adequately supported by empirical evidence and as such they are hardly generalisable. Their remarks however, appear to be at odds with recommendations made by Sakakibara et al. (1997) who maintain that despite being limited, the empirical literature can be used to verify the importance of JIT practices.

Depending on the adopted research methodology, relevant studies can be broadly classified as conceptual and empirical with the former based purely on secondary research and the latter employing at least one primary research method e.g. questionnaire surveys, interviews, field observations. This distinction is important as different approaches succeed in capturing the perceptions of different circles academic and/or practitioner (Zhu and Meredith, 1995).

Another point of divergence in the available literature concerns the primary research objective of the investigations undertaken. Whilst certain studies propose a complete implementation methodology, framework or model for the adoption of leanness, others place emphasis on the identification of the most significant or widely adopted elements, components, building blocks, tools, techniques, practices and so forth. As a result, the

findings reported in different studies are difficult to compare and generalise (Papadopoulou and Özbayrak, 2005).

Hallihan et al. (1997) observe that lack of consensus in providing universal definitions of systems and their components as well as the absence of standard terminology in the descriptions of features they encompass not only prevent generalisation of the research outcomes but cause great confusion. They proceed to explain that different terms are used to designate the very same feature as frequently as different features are grouped under the same term. Shah and Ward (2007) concur and argue that the semantic confusion surrounding LP results in the same components of LP being masked under different terminology.

The review of the relevant literature suggests that classification is another point in which previous studies fail to converge. Harrison (1992) states that due to their diverse nature, organising techniques encompassed in manufacturing systems into clusters is a complex task. Classification in this context concerns differentiating techniques into core or supporting main or infrastructure (ibid; Flynn et al., 1995). However, there exist studies which simply focus on the identification of critical implementation elements and discuss their prioritisation without separating them into core and peripheral or proposing any form of categorisation (Keller and Kazazi, 1993).

Early literature delving into the implementation of leanness was primarily focused on the TPS. Nevertheless, as interest in the success of the TPS continued to grow, there was a subsequent widening of focus away from the TPS on systems resembling it including JIT manufacturing and LP (McLachlin, 1997). The literature concerned with the implementation of leanness is effectively split into three main strands. The first considers the implementation of the TPS whereas the second and third the implementation of JIT and LP respectively. The most significant volume of research is interestingly covering the second strand.

### **2.5.1 TPS practices and underpinning philosophy**

Mehra and Inman (1992) found that although several studies consider the implementation of leanness within the context of a wider discussion, there is no significant volume of literature specifically addressing this issue. This observation holds true for the early TPS literature. Despite focusing primarily on JIT and the full utilisation of worker capability as the two most distinctive features of the TPS, the seminal work of Sugimori et al. (1977) provides useful guidance on how enabling practices interrelate to support the implementation of the TPS.

Monden (cited in Sakakibara et al., 1997) is credited with the dissemination of key TPS practices in the US through a series of articles focusing on JIT (1981a), the adaptation of the Kanban system (1981b), production smoothing (1981c) and reduction of set-up times (1981d). An integrated approach towards the adoption of the TPS providing an overview of the overall JIT methodology and supporting practices is presented in subsequent publications by Monden (1983, 1998).

The TPS and its overarching layers are discussed in Pegels (1984). Further to describing the operation of JIT and the role of the Kanban information management system, this work provides guidance on a range of shop-floor alterations required to facilitate the smooth operation of the Kanban system and achieve the objectives of JIT production. The overview of Kanban requirements focuses on simultaneous operations (andon yo-i-don), re-designed processes, modified tooling to enable quick machine set-up, multi-task workforce, autonomous inspections and production line warning systems (andon jidoka).

The illuminating review of the full features of the TPS presented by Ohno (1988) provides broad insight into how the system can be adopted. However, what distinguishes this work from the literature concerned with the issue of the TPS implementation is that it places emphasis on the importance of understanding the philosophy behind the innovations developed at Toyota and proposes their holistic adoption as a total production management system.

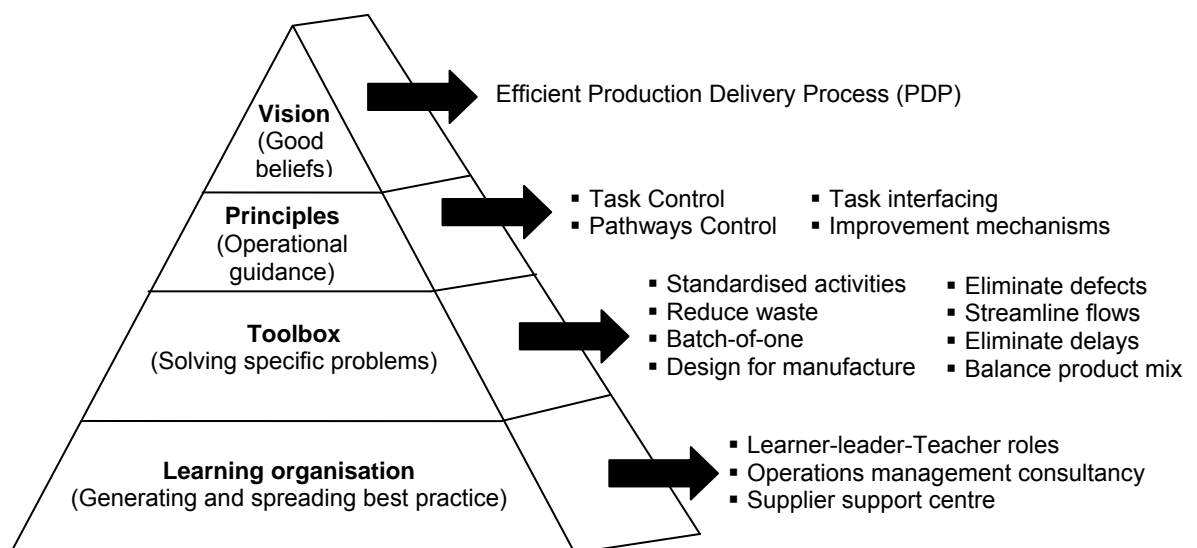
Spear and Bowen (1999) decode the DNA of the TPS and argue that Toyota's success lies not in specific practices but in four ground rules implemented in its factories. According to the authors, it is these rules which are observed in Toyota plant visits and hold the key to the implementation of the TPS. The premise of their research centres on Toyota's commitment to be a learning organisation which employs an almost scientific method to introduce changes. The scientific method is based on hypotheses formulation and experimentations as part of a rigorous and ongoing problem-solving process involving workers and managers at all levels of the organisation. The first three of the proposed rules are design rules determining how the content, sequence, timing of operations can be specified and how the latter can be linked to form a simple and direct product path. The fourth rule describes how workers can be engaged in continuous improvement following the scientific method.

Highlighting the importance of the proposed DNA code in fully understanding and exploiting the potential of the TPS, Towill (2007) develops a triangular prism model to describe the TPS production delivery process. The key elements of the TPS are

organised in four interlinked levels corresponding to vision, principles, toolbox and learning organisation. In line with the prism model, the main TPS principles relate to task interfacing and control which are prerequisites for task coordination and information flow; pathways control is linked to value stream mapping to remove unnecessary pathways and non-value adding steps and improvement programmes.

Interestingly, waste elimination features as one of the main tools of the TPS whilst in the early TPS literature (Sugimori et al., 1997, Shingo, 1981; Ohno, 1988) it is regarded as the ultimate objective and goal of the TPS. Practices supporting JIT production including batches of one item and balanced product mixes are also listed as TPS tools. Towill acknowledges that all the tools incorporated in the TPS prism model may be viewed as standard industrial engineering and production management practices. This observation is consistent with Hayes' (1981) findings according to which Toyota's success was the result of the emphasis placed by its managers on manufacturing basics and constant improvements of the entire manufacturing process.

The recognition of this fact also lends support to the view that the main point of differentiation of the TPS from conventional manufacturing systems lies in the concept of the learning organisation. The coaching of workers, the function of an internal consultancy and training of suppliers are the key elements listed in the learning organisation level of the model. Apart from being an effective visual tool, the four level prism presented in Figure 2.3 constitutes one of the most contemporary illustrations of the TPS.



Source: Towill (2007)

Figure 2.3 TPS four level prism delivery model

Inspired by the work of Spear and Bowen, Jayaram et al. (2010) develop a conceptual framework comprising rules and practices which typify the TPS. Their aim is to study the individual as well as synergistic effects of rules and practices on manufacturing performance measured in terms of cost, quality and time criteria. The proposed set of rules sets the context for the implementation of the practices. The rules govern structural work design and seek to promote learning and problem-solving not only within the organisation but externally by involving suppliers. The TPS practices comprise the Kanban system, preventive maintenance, GT, set-up time reduction techniques, in-plant Electronic Data Interchange (EDI), shared production schedule information with suppliers and JIT supplier delivery.

Their regression analysis reveals overall positive relationships between TPS rules and practices and manufacturing performance. However, the combined effects of interacting rules and practices on performance create a jumbled picture consistent with the complex nature of the TPS. These findings confirm that some of these synergistic effects may be lost if a piece-meal adoption of TPS practices is attempted instead of an integrated implementation.

### **2.5.2 JIT production elements and techniques**

In contrast to the TPS, the implementation of JIT production is considered as the central or simply a side issue in a plethora of papers. There are two possible reasons for this and it is likely that they are interrelated. Since the introduction of the TPS, its proponents have consistently promoted JIT as the main driving force behind the system. One explanation for the attention drawn to JIT may therefore lie in its widespread recognition as the most dominant concept of the entire system. The second possible explanation may be found in the daunting challenge presented by the sheer magnitude, scope and complexity of the TPS. It is logical to assume that due to the latter researchers took a partial approach when attempting to disentangle its true essence by decomposing the system and focusing on JIT as one of its integral parts.

Voss (1984) presents the following list of manufacturing practices which contribute to the success of Japanese firms: clean and orderly workplace; minimised inventory; problem prevention; continuous incremental improvement; incorporating quality in product design and workforce training; equipment policies relating to standardisation and maintenance. These recommendations are based on findings previously reported by Hayes (1981). The use of robots and quality circles are identified as practices of secondary importance. The list further extends to practices reported by other authors



including Kanban, MRP, worker ability to stop the line, attention to detail and strategic operations policy.

Voss's main objective is to examine the level of adoption of these practices in the United Kingdom (UK) using the cases of one British manufacturing firm and one Japanese owned firm based in the UK as a test-bed. The survey results demonstrate a selective adoption of practices by the British firm. The same empirical results however confirm the widespread adoption of the majority of practices by the Japanese firm with some exceptions concerning the use of robots, quality circles and Kanban. The exclusion of quality circles is not surprising as workforce involvement in quality and continuous improvement is emphasised in other practices incorporated in the list. It is further not surprising to see automation excluded as when the concept was originally introduced in the context of JIT it was not associated with the use of robots (Hayes, 1981; Ohno, 1988). Nonetheless, the finding that the Japanese firm had not adopted the Kanban system is intriguing considering its emphasised importance for the operation of JIT production (Sugimori et al., 1997) and the fact that its antidote MRP was not utilised either.

With regards to the adoption of the Kanban system, similar findings are reported by Voss and Robinson (1987) in research employing questionnaires and interviews to establish the level of application of JIT techniques in the UK manufacturing industry. A list of 17 JIT purchasing, manufacturing and supply techniques identified through secondary research was used to collect and analyse the responses of manufacturing companies already implementing or planning to adopt JIT. Flexible workforce, WIP reduction, product simplification, preventive maintenance and statistical process control were ranked as the most frequently implemented or considered for adoption techniques.

Practices including mixed modelling, smoothed line build rate, parallel lines, U-shaped lines recognised by the authors as core JIT techniques were identified in the empirical survey as the least commonly used or considered for adoption with Kanban featuring at the bottom of the ranking table. In line with the authors' interpretations of the findings, the results revealed a preference towards easier to implement techniques whilst it was noted that elements requiring significant commitment to JIT principles or high investment costs were less favoured.

Harrison (1992) proposes a classification scheme whereby JIT techniques are organised as in-company, inter-company and supportive mechanisms. In-company JIT techniques support the conversion of the manufacturing system into a JIT production

facility and comprise amongst others integrated JIT/MRP, pull scheduling, lot size reduction, layout conversion, total quality and total productive maintenance. JIT deliveries, EDI and long-term contracts are some of the inter-company techniques used to extend JIT to suppliers whereas supporting mechanisms are peripheral systems and procedures e.g. Value Engineering (VE), Statistical Process Control (SPC), undercapacity scheduling used to facilitate the implementation of core techniques.

Elements of JIT which are critical to the successful implementation of the system are examined by Mehra and Inman (1992). By reviewing previous literature in which JIT implementation was either the main focus or one of the issues addressed, the authors identify 20 JIT elements which they further group under the following four broad implementation factors: management commitment, JIT production strategy, JIT vendor strategy, JIT education strategy. A questionnaire survey is employed to determine the criticality of each of these factors in the successful implementation of JIT. The statistical analysis of the survey data demonstrates a positive relationship between successful JIT implementation and two factors, namely production strategy and vendor strategy. Specific elements grouped under production strategy include set-up time reduction; in-house lot sizes; GT; cross-training; preventive maintenance whilst vendor lot sizes; sole sourcing; vendor lead time, quality certification of suppliers are JIT elements clustered under vendor strategy. Despite being of certain value, management commitment and education strategy were not verified as critical factors for the successful implementation of JIT.

Keller and Kazazi (1993) regard the creation of a JIT culture and the need for management commitment, workforce involvement and robust relationships with suppliers as the main prerequisites for the effective implementation of JIT. Their broad conceptual research identifies a set of critical JIT implementation techniques which are prioritised in the following order: Total Quality Management (TQM), inventory minimisation, commitment to the principle of getting things right-first-time, maximum flexibility, education and training. Warning lights (andons), autonomation (jidoka), continuous improvement (kaizen) and fool proof devices (poka yoka) are further recognised as important JIT implementation tools.

An auditing procedure designed to assess the speed and effectiveness of JIT implementation is developed by Kazazi (1994). The proposed auditing method is based on a checklist comprising 77 items which are grouped under five areas. 59 items relating to design, implementation and operation of JIT are identified using theoretical and empirical data and grouped under the following four headings: manufacturing technical system requirements, supplier relationships, human resources, quality and

reliability. The remainder 18 items concern performance criteria used to measure the benefits resulting from the successful implementation of JIT. As the main focus of this study is the assessment of JIT effectiveness, the identification of the 59 common JIT implementation practices is a useful by-product of this research.

An investigation specifically focusing on critical elements of JIT implementation is carried out by Zhu and Meredith (1995). The authors acknowledge that previous research attempting to address this issue produced mixed findings and use secondary research data to compile a list of 24 JIT implementation elements which are further ranked in terms of how frequently they are reported in the surveyed literature. Frequency distributions of the data based on research method and author (academic or practitioner) are presented and analysed. Apart from slight variations in the ranking, the same elements dominate consistently the highest ranking positions. More specifically, the ten most frequently reported implementation elements of JIT are: quality circles; set-up time reduction; cross-training; quality certificate of suppliers; GT; in-house lot sizes; vendor lead time; JIT education; relationship with supplier; vendor lot sizes.

It is noteworthy that Kanban and other fundamental JIT techniques relating to pull scheduling are not listed among the 24 identified elements. By relying merely on theoretical research, a significant limitation of this study is that it fails to associate the adoption of the identified JIT techniques with specific manufacturing contexts. As a result, direct comparisons of these results with the findings reported in studies considering manufacturing in certain geographical regions are neither straightforward nor safe.

Flynn et al. (1995) posit that TQM practices improve JIT performance by eliminating process variability and rework time whilst JIT practices improve quality performance by exposing problems and providing timely process feedback. Although the main aim of their study is to examine the interaction and trade-offs that exist between TQM and JIT, part of their research is concerned with the synthesis of conceptual data in order to identify practices unique to JIT. The authors claim that the overlapping that exists between TQM and JIT creates great difficulties in accurately associating practices with each specific system.

In the context of their work, practices are the approaches (inputs) used to achieve desirable performance (output). The following elements are proposed as unique JIT practices: Kanban, lot size reduction, set up time reduction and JIT scheduling. Further to a set of unique TQM practices a list of common infrastructure practices which create

the appropriate context for TQM and JIT is also presented comprising among others management support, workforce management and supplier relationship approaches.

In a similar study investigating the individual and combined effects of core and infrastructure JIT practices on manufacturing performance, Sakakibara et al. (1997) argue that the crucial role of supportive practices in the successful implementation of JIT is manifested by the high awareness and appreciation of these practices in the early JIT literature. Combining observations from plant visits and secondary research, they identify the following six key practices which they view as unique to JIT: set-up time reduction, scheduling flexibility, maintenance, equipment layout, Kanban, JIT supplier relationships. They further explain that practices which provide supportive infrastructure for the application of JIT are related to quality management, workforce management, manufacturing strategy, organisational characteristics and product design. Whilst their findings are significant in several respects, the proposed JIT core practices are rather broad with specific techniques or tools used to achieved certain desirable states e.g. scheduling flexibility, maintenance etc. deemphasised.

Ahmad et al. (2003) examine the impact of infrastructure practices on the effectiveness of JIT practices. The core JIT practices considered in their analysis are similar to those proposed by Sakakibara et al. with the exception of maintenance which is replaced by JIT links with suppliers. Similarities are also observed between the JIT infrastructure practices they propose and those already reported by Sakakibara et al. The authors organise the initiatives, procedures and skills which support the application of JIT into four sets: quality management, manufacturing strategy, work integration systems, Human Resources Management (HRM) systems.

A review of previous empirical research which sought to identify the most frequently utilised or considered for implementation JIT waste elimination techniques in several industrial regions is undertaken by Hallihan et al. (1997). The authors identify that the following nine JIT elimination techniques are commonly reported in the findings of previous surveys: flexible/cross trained workforce and job enrichment; WIP reduction and small lot sizing; JIT purchasing; TPM; set-up reduction; product simplification, standardisation and modularisation; operator centred quality; levelled and mixed production; GT and U-shaped lines.

The absence of Kanban and pull scheduling from the set of nine most frequently practiced JIT elimination techniques is surprising, considering the advocated close link between Kanban and JIT production and the crucial role of the Kanban system in controlling WIP and waste. Nevertheless, this finding is consistent with similar

observations reported in previous research (Voss, 1984; Voss and Robinson, 1987). Hallihan et al. fill the gap observed in the practiced core of JIT by supplementing the nine most frequently utilised JIT techniques with an additional set of four waste elimination techniques. They observe that the four selected techniques are reported in scholarly JIT publications. These include visual control systems; housekeeping; pull control and kanban, automation and defect control. The result of this merging is a combined core of 13 JIT waste elimination techniques.

A conceptual framework comprising the building blocks of JIT is developed by Swanson and Lankford (1998). The proposed five building blocks are company-wide commitment; right material at the right time; supplier relationships; communication linkage; quality; personnel.

In a recent study aiming to determine the requirements for effective adoption and implementation of JIT production, Matsui (2007) uses empirical evidence from the Japanese manufacturing industry to measure the impact of JIT building blocks on competitive performance. The recommended three JIT building blocks are: organisation and HRM; quality management, production information and JIT production systems; technology development and manufacturing strategy. A set of metrics is also proposed to assess these blocks.

It is interesting to note that some of these metrics e.g. MRP adaptation to JIT, kanban/pull system, repetitive nature of master schedule etc. are commonly recognised JIT tools which in this case are used to provide a performance measuring scale. This observation is in agreement with the views of previous researchers who identified the lack of universal terminology as one of the main limitations of the lean implementation research.

### **2.5.3 Lean Manufacturing facets**

JIT is one of the four LM bundles proposed by Shah and Ward (2003) in a study examining the relationship between lean implementation and operational performance. Three more bundles related to TQM, TPM and HRM are formed, comprising 22 interrelated LM practices identified through a survey of the literature. The aim of their study is two-fold. Initially they seek to examine the impact of the organisational context on the pattern of lean implementation determined by the selection of specific LM practices. Their research is further concerned with the synergistic effect of practices encapsulated in the lean bundles on performance.

The 22 LM practices and their categorisation into the four bundles is illustrated in Figure 2.4. This further shows the factor loadings determined during the empirical validation of the bundles. The JIT bundle comprises production flow techniques and a range of waste elimination concepts including that of the focused factory introduced by Skinner (1974) to describe a factory where focus is shifted from productivity to competitiveness. The inclusion of agile manufacturing strategies in the JIT bundle is intriguing. Agile manufacturing is advocated as a contemporary manufacturing paradigm attracting similar attention to that drawn to LM rather than an element of the latter (Narasimhan et al., 2006).

Papadopoulou and Özbayrak (2005) carry out conceptual research in order to identify elements which are critical for the successful implementation of LM. Their research reviews previous literature whereby the transition to a lean state, JIT or LP, has been the main focus or one of the issues addressed. The authors point out that diverse perceptions of leanness have resulted in mixed and conflicting recommendations with regards to its implementation.

Lean component	Factor loadings			
	JIT	TPM	TQM	HRM
Lot size reductions	<b>0.659</b>	0.062	0.007	0.031
JIT/continuous flow production	<b>0.649</b>	0.081	0.213	0.116
Pull system	<b>0.647</b>	-0.147	0.256	0.118
Cellular manufacturing	<b>0.631</b>	-0.234	0.180	0.105
Cycle time reductions	<b>0.586</b>	0.248	0.014	0.054
Focused factory production systems	<b>0.562</b>	0.051	0.170	0.164
Agile manufacturing strategies	<b>0.552</b>	0.327	0.075	0.146
Quick changeover techniques	<b>0.537</b>	0.336	0.030	-0.064
Bottleneck/constraint removal	<b>0.501</b>	0.349	0.126	0.151
Reengineered production processes	<b>0.440</b>	0.288	0.138	0.023
Predictive or preventive maintenance	-0.001	<b>0.715</b>	0.198	0.116
Maintenance optimisation	0.038	<b>0.681</b>	0.168	0.176
Safety improvement programs	0.012	<b>0.552</b>	0.240	0.089
Planning and scheduling strategies	0.314	<b>0.458</b>	0.050	0.141
New process equipment or technologies	0.248	<b>0.418</b>	0.147	-0.197
Competitive benchmarking	0.256	<b>0.364</b>	<b>0.361</b>	0.073
Quality management programs	0.024	0.178	<b>0.741</b>	0.079
Total quality management	0.177	0.219	<b>0.705</b>	0.160
Process capability measurement	0.211	0.101	<b>0.660</b>	-0.079
Formal continuous improvement program	0.179	0.271	<b>0.605</b>	0.206
Self-directed work teams	0.138	0.128	0.208	<b>0.758</b>
Flexible, cross-functional workforce	0.259	0.177	0.042	<b>0.710</b>

Source: Shah and Ward (2003)

Figure 2.4 Principle component analysis of the four lean bundles

They propose a classification scheme which organises LM elements into the following clusters: production flow management; product/process-oriented; production planning, scheduling and control; lean implementation; workforce management; supply chain management.

The lean implementation category comprises project management practices used to drive the introduction and application of leanness. Such practices include the creation of an appropriate lean vision, the commitment of top management, the use of lean champions etc. The same category further includes practices aiming to sustain and extend the lean project e.g. plant-wide adoption and continuous improvement. These findings support the main premise of their research in line with which the inherent dynamic nature and evolutionary characteristics of leanness create the secret formula for its remarkable success.

Viewing the transition to a LM system as a long-term journey and adopting a continuous improvement approach are also recognised by Bhasin and Burcher (2006) as important preconditions for the successful implementation of leanness. According to the authors, LM poses a combination of technical and cultural requirements. The technical requirements concern the adoption of 10 lean tools comprising continuous improvement/kaizen; cellular manufacturing; kanban; single piece flow; process mapping; SMED; step change (kaikaku); supplier development; supplier base reduction; five S and visual management; TPM; value and seven wastes.

The authors stress the importance of embracing the whole set of tools rather than adopting a piece-meal approach. The aforementioned technical requirements are generally described as lean tools and although this may be the case for some of them e.g. cellular manufacturing, kanban, process mapping, it can be arguably claimed that other requirements rather point to fundamental lean concepts such as those of value and the seven wastes or broader lean methodologies with continuous improvement as one characteristic example.

Doolen and Hacker (2005) use secondary research data to develop a LM model which they further use as a survey instrument to assess the level of lean adoption by electronics manufacturers in the US. The model categorises lean practices into the following impact areas: manufacturing equipment and processes; shop-floor management; new product development; supplier relationships; customer relationships; workforce management. Shah and Ward (2007) use empirical data to develop a conceptual model of LP comprising 48 practices/tools which are further grouped under

internal, supplier and customer-related constructs. Pull, flow, setup time reduction and productive maintenance techniques are some of the internal operating constructs.

## **2.6 Why become lean?**

The proliferation of articles addressing the issue of lean implementation prompted a subsequent shift of focus on performance-related issues. Proponents of lean strategies demonstrated their pre-eminence by underscoring their ability to improve strategic competitiveness in world-class manufacturing environments (Sánchez and Pérez, 2001). On the other hand, researchers called for a more judicious review of the preconditions necessary to create and sustain a lean competitive advantage. Lewis (2000) associates lean competitiveness with the firm's ability to utilise its strategic resources and reap the financial benefits generated from lean savings.

Mixed findings in relation to the effectiveness of JIT have been viewed as one of the main reasons for their relatively conservative adoption by US manufacturers (Fullerton and McWatters, 2001). Anecdotal evidence is emphasised by Soriano-Meier and Forrester's (2002) assertion that not only implementation patterns but further the resulting outcomes from the adoption of lean practices lack parity from one case to another. Shah and Ward (2003) imply that inconsistency in reported findings on JIT and TQM performance can be attributed to the failure of the relevant literature to consider the impact of the organisational context in which their implementation takes place.

### **2.6.1 Lean benefits**

Lean practices are commonly linked to world-class manufacturing performance (Sakakibara et al., 1997). Early research investigating the benefits firms can derive from the adoption of these practices is mostly theory-driven. Schonberger (1982b) acknowledges that the most widely perceived JIT benefit related to WIP inventory reduction can in turn improve forecast accuracy, dispatching and communication thus raising customer responsiveness levels. He develops a cause and effect model to illustrate a series of quality-related benefits derived from WIP inventory reduction.

Performance data collected from UK-based manufacturers is reported by Voss and Robinson (1987). Firms participating in this survey were asked to rank a list of JIT benefits. Results indicated WIP reduction followed by increased flexibility and quality as the major benefits resulting from the adoption of JIT. Overall reduction in inventory levels and WIP, increased flexibility and quality improvements are also reported by



Kazazi and Keller (1994) as key JIT benefits. Their research uses data collected from European JIT-adopters and suggests that improvements in time-based performance specifically set-up and lead times, product reliability, productivity and relationships with fewer suppliers were also recognised as JIT-related benefits by surveyed firms.

Zhu and Meredith (1995) identify increased inventory turnover, improved quality, reduced lead times and machine/worker utilisation improvement as commonly reported JIT outcomes in empirical surveys involving US manufacturing firms. A similar questionnaire-based survey carried out in the US by Wafa and Yasin (1998) confirms JIT can improve quality performance, customer service, collaboration with suppliers as well as result in cost savings and higher production efficiency. A point of immediate interest in their findings is the ability of firms where both top management and workforce actively supported the implementation of JIT to reap a higher number of JIT benefits.

Shah and Ward (2003) seek to address a gap in the literature concerned with the impact of LP on operational performance. They claim that previous empirical studies focus on individual lean facets thus ignoring the synergistic effect of these diverse yet complementary practices on performance. Their research organises lean practices into four bundles and studies the effect of their concurrent application on operational performance measured in terms of cycle time, lead time, first pass yield, scrap and rework costs, product unit costs and labour productivity. Data collected from a large-scale questionnaire survey carried out in the US unambiguously affirmed the association between the simultaneous application of multiple lean facets and higher performance.

It is evident that whilst early studies focused primarily on operational efficiency, consideration of the impact of JIT on financial performance was de-emphasised. Upton (1998) explains that removing waste and tightly interconnecting production stages to create a JIT environment calls for the use of appropriate non-financial performance indicators to measure operational improvements. By analysing data collected from a sample of manufacturing firms in New Zealand, his research establishes that almost a third of the surveyed firms used traditional accounting measures to assess benefits resulting from the adoption of JIT. Upton points out that this is an alarming finding particularly since it is commonly recognised in the literature that traditional costing systems can undermine and even encumber the implementation of JIT.

Lewis (2000) uses case data to analyse the impact of LP on overall business performance. Sales and profit figures suggest embracing a lean strategy does not

necessarily improve financial performance. For Fullerton and McWatters (2001) the connection between JIT and profitability is of key importance, as it can influence the decision made by firms to incur the necessary investment costs and adopt JIT. Motivated by the inconsistent findings of studies examining the impact of JIT on financial performance, their empirical research confirms that increased firm profitability was among a range of benefits enjoyed by JIT adopters.

According to Fullerton and Wempe (2009) mixed evidence on the association between LP and financial performance can be attributed to disparity in the adopted methodologies, piece meal adoption of disentangled lean practices, context-specific parameters and the use of non-financial performance metrics to assess the benefits derived from LP. They propose the adaptation of management accounting systems to include non financial performance measures and use structural equation modelling to analyse the relationship between the latter and firm profitability. Their findings corroborate the mediating role of non financial performance measures on the impact that LP has on financial performance.

Recognising that inventory reduction can improve the operational and financial performance of firms and bring greater benefits to the wider national economy by releasing tied up capital, Swamidass (2007) analyses the inventory performance of discrete manufacturing firms in the US. Participating firms are initially ranked into top/middle/bottom-TPS performers using a composite score that measures Return on Assets (ROA), Return on Sales (ROS) and Return on Equity (ROE). Regression models used to analyse trends in inventory performance in terms of total inventory over sales confirm a cumulative and permanent reduction of inventory only in the case of top performers. Conversely, Swamidass acknowledges that the observed inventory growth in bottom performers is an unexpected finding calling for further research on the sources of this problem.

Eroglu and Hofer (2010) challenge an assumption commonly made in the relevant literature which postulates a linear relationship between inventory leanness and firm performance. They develop a measure of inventory leanness which considers the effect of firm size on inventory holdings thus providing a more accurate estimation of the firm's degree of leanness. Using empirical data from the US manufacturing industry they conclude that although the effect of inventory leanness on firm performance is frequently non-linear but positive, in most instances there appears to exist an optimal degree of leanness that once exceeded alters the direction of this relationship. In these cases, firm financial performance in terms of ROS and ROA deteriorates. This survey focuses on medium-sized publicly traded firms but a more important limitation is that it

does not analyse the underlying factors that can lead to a U-shaped relationship between inventory leanness and firm performance.

Inventory turnover is considered a tangible and crucial measure of world-class performance by Demeter and Matyusz (2011) who assess the impact of lean practices on firm inventory turnover. Their research further investigates the effect of various contextual factors on inventory turnover. Using data from manufacturing industries from 23 countries they identify a significant relationship between shop-floor configuration i.e. serial, cellular etc. and the levels of WIP maintained in the system. On the contrary, raw materials inventory and FGI are found to be affected by the type of production system either Make-to-Order (MTO) or Make-to-Stock (MTS). Overall, their research establishes a positive relationship between lean practices and inventory turnover with lean firms maintaining lower levels of inventory. These results appear to contradict Swamidass's findings; however, a direct comparison cannot be performed as their results are based on data representing only high technology industries.

### **2.6.2 Lean metrics and performance measurement systems**

Hallihan et al. (1997) stress the importance of developing appropriate performance measures and systems to guide and assess the implementation of JIT. They explain that using traditional accounting systems to monitor JIT improvements is ill-advised as the former are designed to assess performance objectives that contradict the lean ideology. They develop the JIT implementation pyramid, a model which organises support levers, waste elimination techniques and performance measures into three discrete levels founded on the basis of continuous improvement. Using theoretical data, they populate their model with 19 metrics designed to measure time, inventory, housekeeping, quality, JIT delivery and productivity performance in the context of JIT. This is one of the most extensive lists of JIT-related performance measures proposed in the relevant literature but Hallihan et al. are unwavering that given the diverse nature of JIT a wide range of measures is necessary to assess improvements resulting from it. Their view is consistent with Wan and Chen's (2008) finding that it is practically complex to develop an integrated measure of leanness by synthesising individual metrics.

According to Katayama and Bennett (1999) labour productivity is one of the most commonly cited lean measures, yet associating productivity with investment on automation can compromise the firms responsiveness to demand variations and its degree of leanness as it increases the company's fixed assets. Moreover, utilising this

measure alone provides a short-sighted view of the wider range of benefits resulting from a lean approach.

Productivity decreases during the transition to a LP system and their negative impact on financial performance measures used by traditional accounting systems are viewed by Sánchez and Pérez (2001) as common factors that often discourage, if not hinder, the adoption of LP. In order to overcome these possible shortcomings, they propose the use of intermediate lean indicators which can help adopting firms gauge the impact of changes implemented during the introduction of LP. They further develop a model comprising 36 intermediate indicators related to the elimination of non-value adding activities, continuous improvement, team work, JIT production and delivery, supplier integration and use of flexible information systems.

A leanness metric assessing cycle-time performance is the Manufacturing Cycle Efficiency (MCE) index (Levinson and Rerick, 2002 cited in Wan and Chen, 2008). The index provides a measure of manufacturing efficiency by comparing value adding time in the overall process with cycle time.

Detty and Yingling (2000) use simulation to measure the benefits resulting from the adoption of LM in the case of a consumer electronics production facility. Although their simulation output includes solely quantifiable lean benefits such as inventory levels, lead time, utilisation rates, their work endorses simulation as a potentially powerful incentive system for the introduction of lean practices. They explain that simulation is an adaptable tool that can be used to compare the performance of the existing pre-lean and proposed post-lean system and thus support the decision to adopt leanness prior to any actual investment to facilitate the necessary changes. They point out that by using simulation organisations do not need to base their decision to adopt leanness on the experiences of other firms or theoretical rules of thumb about its potential benefits. This observation is consistent with Fullerton and McWatters's (2001) view that lack of an adequate and effective performance measurement system is one of the main reasons behind management reluctance to adopt JIT.

A classification scheme which systematically organises LM tools and metrics is developed by Pavnaskar et al. (2003). The scheme has a tree-like structure and categorises LM tools into the following seven levels: system, object, operation, activity, resource, characteristic and application. Although not intended for use as a decision-making tool, the classification scheme can be used by organisations to either relate tools with their applications and waste elimination metrics they can help achieve or

conversely match specific production problems and sources of waste with the tool(s) that can eliminate them.

The work of Pavnaskar et al. is subject to a number of serious limitations. First, it is suggested that in order to develop the scheme 101 LM tools reported in the literature were classified. Nonetheless, this work classifies only a very small subset comprising five tools namely, cellular layout, facility layout, load levelling, six sigma and value stream mapping and there is further no indication of the remainder tools considered. Second, the proposed classification scheme is not validated in a real-world industrial setting. Third, the LM tools considered are only related to manufacturing operations/activities directly associated with the production of finished products whereas other organisational functions e.g. product design and development are not accounted for. This is in direct contrast with the lean enterprise ethos according to which lean practices are not limited by the strict boundaries of the shop-floor rather they extend to all business functions of the organisation (Womack and Jones, 2003). Finally, the advocated aim of this classification i.e. to facilitate the selection of appropriate LM tools by organisations adds to the common and disquieting misconception that a piece meal approach to the adoption of tools can lead to an effective and successful implementation of LM. As Shah and Ward (2003) highlighted, lean practices are complementary and unless they are collectively and simultaneously applied their true synergistic potential cannot be realised.

### **2.6.3 Assessment of degree of leanness**

Karlsson and Åhlström (1996) propose a model which can be used to assess the changes taking place during the process of adopting and introducing LP. Despite developing their own conceptualisation of the lean enterprise as a framework comprising lean practices affecting various functions i.e. lean development, lean procurement, LM and lean distribution, their assessment of leanness is primarily focusing on the manufacturing functions of the organisation. Their model for assessing LP changes comprises nine groups of lean determinants and measurements. The determinants are theoretical indicators reflecting changes required to adopt lean principles. Each determinant is subsequently associated with a set of operationalised measurements that have been empirically tested in the case of a manufacturing firm producing office equipment and proven suitable for assessing these changes. Taking the lean principle of zero defects as a case in point, one determinant is worker responsibility for identifying defective parts and a relevant measurement is the number of workers identifying defective parts and stopping the line.

The Lean Enterprise Self-Assessment Tool (LESAT) developed by an integrated team of industrial, academic and government collaborators under the auspices of the LAI at MIT (Nightingale and Mize, 2002) is different from similar lean assessment models in that it aims to assess the maturity level of an organisation in its use of lean practices and principles. The tool can be used to provide assessment in the following three sections: lean transformation/leadership, life-cycle processes and enabling infrastructure. The maturity matrices incorporated in the model identify and organise 54 lean practices under each of these three sections enabling firms to use the tool periodically in order to get snapshots of how the lean transformation is progressing. Although field tested in the aerospace industry, it is argued that the tool can be adapted for use in other industries.

Wan and Chen (2008) use data envelopment analysis and linear programming to develop a quantitative measure of how lean a production system is. Comparing the proposed measure with other lean assessment models they stress that the uniqueness of the measure lies in a set of distinctive features. The measure can provide an integrated leanness index weighing performance in terms of cost, time and value. It is scalable as it can be used to assess the leanness of a single cell, production line or entire production facility. It finally provides a self-contained benchmark. However, the effectiveness of the proposed measure has not been empirically validated.

A relative measure of leanness is developed by Bayou and De Korvin (2008) using a fuzzy logic methodology. Arguing that leanness is a dynamic variable that develops from a lean, to a leaner and ultimately to the leanest state, they insist a relative, dynamic and long-term measure of leanness is necessary to objectively compare the progress made by different organisations aspiring to become lean. Using the Honda Motor Company as the benchmark they empirically test the proposed measure in the cases of Ford Motor Company and General Motors. Nonetheless, the main weaknesses of their approach relate to narrowly focusing on few components of leanness e.g. JIT, Kaizen and quality control and using surrogate measures for these components pooled from financial statements.

Vinodh and Chintha (2010) use a similar approach involving a multi-grade fuzzy logic methodology to develop a model for measuring leanness. The model considers lean enablers, lean criteria and lean attributes and thus aims to assess leanness from three different perspectives. Although empirically tested, this has only been attempted on a single case-study thus preventing safe conclusions about its general applicability and effectiveness.

## **2.7 Common lean misconceptions**

The exploration of extant research on LP reveals three emerging contentious propositions. The first concerns theoretical and empirical attempts to discount lean to a manufacturing tool-box from which practices and techniques can be selected and used on an ad-hoc basis. The second proposition seeks to address whether lean is currently being superseded by rival world-class manufacturing paradigms. The third proposition summarises issues limiting its applicability within and across manufacturing sectors. The analysis of these debates suggests that lean has been both praised and negatively criticised. Most importantly, the last two propositions are of particular relevance to the main premise of this research.

### **2.7.1 Is lean simply a set of tools?**

The work of Voss and Robinson (1987) presents one of the first surveys carried out to assess the level of JIT adoption by UK manufacturers. Their research makes an important contribution to the literature as it is amongst the first to report partial implementation patterns. In particular, Voss and Robinson's primary research findings confirm that despite an admittedly high level of JIT awareness, adoption levels in the UK were very low and indicating a discriminatory preference to techniques which were easier to implement and/or requiring less commitment to JIT.

Ad-hoc selections of subsets of JIT tools were subsequently reported in other works. Westbrook (1988) observes that western manufacturers struggled to embrace Japanese human resources policies promoting worker motivation and development in the context of JIT. Preference to JIT techniques resulting in short-term tangible benefits as well as tools with no major prerequisites are identified as common reasons for selective implementations of JIT by Im and Lee (1989). They present automation as a case in point, explaining that its limited adoption was often linked to a firm's lack of confidence in its preventative maintenance and Quality Control (QC) systems. Lewis (2000) traces the reasons for the piece-meal adoptions of LP in the influential work of Ohno (1988) suggesting that by emphasising the multi-faceted and compact nature of the system, Ohno's work encouraged adopters to decompose the system thus undermining Toyota's 30 years of trial-and-error.

On closer examination of the relevant literature, it becomes clear that reported piece-meal adoptions concern the highly selective use of JIT elements (Keller and Kazazi, 1993), their ad-hoc implementation in specific functions of the organisation (Kazazi, 1994) or in other cases adoptions which were purely tool-focused thus failing to

embrace the lean organisational culture and focus on employee incentivisation and empowerment (Hines et al., 2004).

Piece-meal implementations continued despite alarming evidence that these could compromise a firm's ability to reap the purported company-wide JIT benefits and ultimately improve its competitiveness (Mehra and Inman, 1992). Because of this, non-holistic adoptions were also associated with negative evaluations of JIT. Fullerton and McWatters (2001) provide empirical evidence to support this view and insist that the greater the depth and breadth of JIT implementation, the more significant the resulting benefits.

Shah and Ward (2003) attempt to explain the synergistic effects of diverse yet entangled lean practices on performance. They use worker empowerment and waste elimination to demonstrate this relationship. They explain that the ongoing elimination of waste is dependent on the ability of self-directed work teams to identify and remove non-value adding steps and improve flow between work stages. Coupled with the problem-solving abilities of these teams the removal of inventory buffers can help the identification of hidden equipment problems and minimise machine stoppages, defects and thus improve quality.

The holistic and unifying nature of the LP philosophy is emphasised by Shah and Ward (2007, p.800) who argue that *"viewed separately, none, of the components are equivalent to the system but together they constitute the system"*. Despite theoretical and practical evidence exemplifying the importance of an integrative implementation of lean strategies, extant research confirms that these problems continue to exist. Towill (2007) is adamant that very little emphasis is placed on the cultural changes required to embrace the TPS mindset whilst Jayaram et al. (2010) empirically confirm that piece-meal adoptions of TPS rules and practices impair their synergistic effects on performance.

### **2.7.2 Lean versus rival world-class manufacturing paradigms**

Leanness was broadly recognised as an exemplary model for manufacturing excellence and the relevant literature is replete with lean success stories. Nevertheless, several researchers viewed LP with scepticism. Katayama and Bennett (1996) argue that the success enjoyed by Japanese manufacturers implementing lean strategies in the 1980's and 1990's was mainly due to favourable conditions in the context of the bubble economy in Japan at the time. On this basis, they challenge the viability and robustness of LP in less favourable economic environments.



Lewis (2000) observes that the importance of starting conditions in lean implementation programmes was mostly ignored. He claims the accuracy of the IMVP reports which highlighted the superior performance of Japanese manufacturers in comparison to their US competitors has been seriously challenged and refers to the case of Nissan that under difficult economic conditions failed to achieve similar results. The contention that leanness was overestimated is also supported by Svensson (2001). He remonstrates the origins of leanness can be traced in Ford's production system thus being at odds with those crediting Toyota with these management innovations (Fullerton and Wempe, 2009) and others who accept its Japanese-ness but acknowledge a US influence (Soriano-Meier and Forrester, 2002; New, 2007). Svensson's heaviest criticism of JIT is that, as it offers nothing new it "*only has cosmetic novelty value*" (p. 876).

Claims that the overestimated LP is a passing fad (Lewis, 2000) pushed forward a different research agenda focusing on new challenges faced by manufacturers. Sharifi and Zhang (1999) refer to a new business era which is primarily characterised by rapid change. They assert that to survive and prosper in dynamic business environments, manufacturers need to include proactivity, adaptability and joint ventures with suppliers and even competitors in their strategic objectives. Yusuf et al. (1999) refer to the concept of integration to describe strategic partnering relationships and list speed, responsiveness, proactivity and innovation as key competences which form the basis of competition in 21<sup>st</sup> century manufacturing.

These discussions led to the emergence of new manufacturing paradigms that boosted aspirations to render leanness obsolete. Suri (1998) proposes Quick Response Manufacturing (QRM) as a singular strategy focusing on speed achieved through the reduction of lead time. Originating from Time-Based Competition (TMC) strategies, QRM seeks to shorten response times across the entire supply chain from raw materials to product design and development, fabrication, marketing and delivery. Katayama and Bennett (1999) discuss adaptable production as a new manufacturing approach that combined with fundamental principles of LP can help enterprises compete in rapidly changing markets. They argue that whilst LP focuses on the minimisation of variable costs by reducing resource consumption, adaptable production seeks to optimise the firm's cost performance by appropriately shifting fixed towards variable costs according to demand.

Agile manufacturing is presented by Gunasekaran (1999) as a new manufacturing concept which stresses the importance of achieving flexibility and responsiveness whilst also trying to become lean. He suggests agility is a natural development from leanness. Naylor et al. (1999) argue that lean and agile manufacturing paradigms are

complementary and therefore if simultaneously implemented within an appropriately designed supply chain can maximise the resulting benefits. They present theoretical evidence to demonstrate that LM can be used to produce levelled schedules upstream of the decoupling point i.e. the point in the supply chain whereby strategic stock is held to absorb demand variations. Conversely, they propose the exploitation of the ability of agile manufacturing to satisfy fluctuating demand by using this strategy downstream of the decoupling point.

It is common practice in manufacturing to combine different practices in order to create new management innovations. Flynn et al. (1995) explain that world-class manufacturing paradigms often rely on the synergistic effects of integrated approaches such as JIT and TQM to increase competitiveness. The recently emerged concept of Lean Six Sigma or Lean Sigma (Arnheiter and Maleyeff, 2005) which integrates the LP philosophy with six-sigma techniques is another case in point.

Despite the alleged superiority and newness of these approaches, they were also fiercely attacked. Referring to the case of Time-Based Manufacturing (TBM) Shah and Ward (2007) are highly critical of the work of Koufteros et al. (1998) who practically conceptualise TBM as pull production and develop a TBM framework built entirely on JIT practices including involvement of shop-floor employees in problem-solving, set-up reengineering, cellular manufacturing, quality improvement, preventive maintenance and dependable suppliers. Shah and Ward comment on the impact of this unsubstantiated equation on further research efforts in which the terms pull production and TBM were incorrectly used interchangeably.

Commenting on the novelty of agile manufacturing, Soriano-Meier and Forrester (2002) argue that agility evolved from leanness due to the continuing attention the latter was receiving. Referring to the diverse definitions of leanness which appear in the literature, Wan and Chen (2008) cite the work of other researchers who used the term leanness to introduce and promote leagility. New (2007) provides insight into how the descriptions of manufacturing practices and acknowledgement of their origins are inevitably affected by the agendas and interests of those providing them. He characteristically refers to the case of stockless production developed by Hewlett Packard. Stockless production was first introduced in the early 1980's but is still in use. What is really extraordinary about stockless production is that it is fundamentally based on the principles of JIT although it makes no reference to Toyota. New insists that although many best practice models originate from Toyota there is no general appreciation of Toyota's contribution.

Papadopoulou and Özbayrak (2005) provide a comparative analysis of leanness in relation to newer manufacturing approaches including QRM, adaptable production, agility and leagility. They argue that ambiguity and uncertainty of what leanness really constitutes is the true reason behind the mushrooming of allegedly superior rival practices. However, they deconstruct this argument by suggesting that closer examination of these models confirms an extensive overlap between their main constructs and key lean enablers. They proceed to highlight that although this is not immediately apparent, most rival practices were compared to LM rather than the extended and most recent form of leanness namely the lean enterprise. They conclude that narrow understanding of leanness led to incorrect evaluations of its true potential and falsely supported the superiority of rival approaches.

The review of the work of Yusuf and Adeleye (2002) provides further support to this contention. They compare lean and agile manufacturing and present theoretical evidence which suggests that the lean paradigm is under threat. They argue that LP can simply offer internal efficiency and therefore must be supplemented with agile manufacturing practices. They claim agile manufacturing can provide a responsive supply-chain approach based on networking and strategic partnering, concepts which in fact underpin the lean enterprise model.

### **2.7.3 Lean applicability issues and limitations**

Arguably the most controversial issue about the lean paradigm is its applicability. The first studies investigating the applications of LP concluded that these are entirely situational (Zhu and Meredith, 1995). Flynn et al. (1995) insist that, with no specific recipe available for the adoption of JIT and TQM, their successful application depends on the organisation's ability to review and adapt its culture to the JIT ethos and mindset. They characteristically cite evidence presented by Hall (1983) who described the reluctance of US manufacturing workers to accept the introduction of kanban-based control systems which they regarded as unsophisticated and silly.

Further research evaluating the "context-matters" proposition, identified a range of contextual factors hindering the adoption of LP. The most commonly cited internal resisting forces were incompatible western HR policies (Westbrook, 1988), inflexible hierarchical organisational structures (Bamber and Dale, 2000), workforce issues particularly unionisation and impact on already established production strategies (Lewis, 2000). Nonetheless, this is yet another area where findings are mixed. Shah and Ward (2003) analysed the impact of three internal contextual factors namely level

of unionisation, plant age and firm size on the adoption of a wide set of LP practices in US manufacturing. Contrary to widespread belief that unionisation is encumbering, their empirical findings did not support this contention for the entire set of LP practices. The association between plant age and level of adoption was also unclear. However, their results substantiated a positive association between firm size and adoption clearly indicating that large firms are more likely to afford the investment and resources required to introduce LP.

The identified need for change concerned not only the organisation's culture, business functions, operating procedures and structure but crucially also involved physical changes to adjust plant layouts and re-arrange production facilities to support JIT flow (Voss and Robinson, 1987; Black, 2007). On the other hand, relationships with suppliers (Wafa and Yasin, 1998) and the broader socio-economic conditions in which firms operated (Nakamura et al., 1996) were viewed as significant external factors that could limit the applicability of LP.

The amalgamation of these hindering factors constitutes what Lee and Jo (2007) recognise as the contingency perspective in research exploring the transferability of the TPS. This perspective is adopted by researchers who accept the superiority of TPS yet identify a number of preconditions to its transferability similar to the factors identified above. Lee and Jo explain this perspective is a compromise between two diametrically opposed schools of thought. According to the structuralist perspective, the TPS is a production model confined to Toyota and therefore followers of this school of thought deny its portability. Conversely, the convergence perspective views LP as the ultimately superior production model that can be universally transferred anywhere. The convergence perspective essentially reflects the central hypothesis postulated in the IMVP research which is summarised by Womack et al. (1990, p.278) in the following statement *"LP will supplant both mass production and the remaining outposts of craft production in all areas of industrial endeavour to become the standard global production system of the twenty-first century"*.

The universality of the TPS was examined at various stages of the lean evolution. Pegels (1984) is adamant that the TPS is only applicable in the assembled goods industries and completely rejects its transferability to the process industries including oil refineries and steel mills. Cooney (2002) examines the universality of LP and challenges the assertion presented by Womack et al. that LP will supersede both batch and craft production. It is intriguing that despite the historical distance between their studies, Pegels and Cooney are equally emphatic that LP cannot have a universal application.

The premise for Cooney's contention is that LP relies heavily on JIT's unique approach to product flow, arguing that if this flow cannot be attained in every production system then neither can JIT and LP. Cooney focuses specifically on batch and craft HVLV production systems claiming that producing highly diversified product mixes in low volumes poses serious difficulties in establishing standard production times and in turn achieving production levelling, the main prerequisite for JIT flow.

The central importance of production levelling (smoothing) for the efficient operation of JIT is also recognised by Yavuz and Akçali (2007) who argue that due to the fact that JIT originated from assembly line systems, reported analytical models seeking to address the Production Smoothing Problem (PSP) mainly concentrated on flow-shop and final assembly systems. However, Yavuz and Akçali insist that contrary to common misconception, JIT is a viable option for HVLV systems and call for consideration of the PSP in the context of HVLV manufacturing environments.

These findings suggest that although Cooney succeeded in accurately identifying production levelling as the main cause for the problematic application of JIT in HVLV systems, his dismissal of production levelling as a problem that can be resolved in HVLV production was flawed. Further support to this contention can be found in the work of Cruickshanks et al. (1984) who develop a mathematical model to address the PSP in the case of a HVLV system. The production environment under investigation is a MTO job-shop which operates as a stockless system allowing no uncommitted FGI to be stocked and tolerating no late deliveries. Although not specifically referring to JIT, it is clear that the job-shop considered in their study operates as a JIT system.

Despite initially receiving little research attention (James-Moore and Gibbons, 1997), interest in the application of LP in HVLV systems started to grow with the first investigations reported in the late 1990's. Jina et al. (1997) develop a framework for the application of LM principles in various functions of HVLV systems. They recommend the categorisation of parts into runners, repeaters and strangers based on associated levels of demand and the creation of focused cells, dedicated job-shops and flexible job-shops to process each category respectively. They test the application of LM principles in two HVLV organisations, a manufacturer in the aerospace sector and a manufacturer of specialist machinery and conclude that the specific selection of LM principles is contingent upon the specific circumstances of the adopting HVLV organisation.

Fullerton and McWatters (2001) report the adoption of JIT by HVLV manufacturers across different manufacturing sectors including industrial equipment, electronics, food

and textile. Sorriano-Meier and Forrester (2002) set out to validate the hypothesis proposed by Womack et al. (1990) that LP can be applied to craft production systems. Focusing on the tableware sector of the UK ceramic industry, their study concludes by confirming the successful application of LP in craft production. Doolen and Hacker (2005) describe how highly specialised and organisationally dispersed functions in electronics manufacturing may limit the applicability of LM practices. Using empirical data from the electronics industry, they conclude that despite these challenging conditions, many electronics manufacturers implemented LM to a certain extent.

Papadopoulou and Özbayrak (2005) review previous research concerned with the extension of LP to HVLV production environments. They identify several works which specifically concentrate on the adaptation of job-shop systems primarily through plant reconfigurations to facilitate JIT scheduling. Interestingly, the first studies exploring the applicability of LP in HVLV manufacturing systems were published back in 1985. Abdulmalek and Rajgopal (2007) recognise the lack of documented applications of LM in the process sector and the common yet false perception that this sector is less conducive to the adoption of lean techniques as the main reasons for the limited applications of LM in continuous process industries. Their research presents a methodological approach for the introduction of a hybrid push/pull system and TPM programme in a large integrated steel mill. Their findings demonstrate significant improvement in terms of lead time and inventory performance and evidence from this case-study is used to substantiate the assertion that LM is a feasible improvement programme for the process sector.

Theoretical and empirical evidence presented in documented attempts to extend LP in production environments initially considered less amenable to lean practices support the argument that not only has LP passed the transferability test but its tailored applications in HVLV production settings can yield a range of performance benefits. These findings validate the universal applicability of LP declared by Ohno (1988) and Womack et al. (1990) and urgently call for the notions of academics' and practitioners' around this key issue to be reframed.

## **2.8 Recent industrial applications of the lean paradigm**

There is a plethora of empirical validations of LM in the recent literature and this is a reflection of the currency of the lean paradigm. Most of the studies which investigate the implementation of LM in real-life manufacturing systems employ a case-based approach.

Considering the central importance of waste and value in LM, several studies focus on how value stream mapping tools can be used to drive the transition of manufacturing systems to a lean state. The application of lean visual process management tools on three aerospace manufacturing firms is investigated by Parry and Turner (2006). Through their integration with resource planning software, maintenance and control systems, visual process management tools are confirmed as key lean enablers for world-class manufacturing performance.

Chen et al. (2010) create an integrated value stream mapping and continuous improvement/Kaizen tool to investigate the benefits resulting from the adoption and implementation of LM. The tool is tested in the case of a US electrical manufacturer. Value stream maps are created to illustrate the firm's pre-lean operation status and desirable post-lean state. The value stream maps are also used to identify barriers to the implementation of LM. Kaizen tools and principles are proposed to overcome these. Results obtained from this case study demonstrate that the company considered achieved significant improvement of process efficiency and product quality coupled with the reduction of inventory levels.

Research carried out by Wee and Wu (2009) aims to empirically validate the importance of value stream mapping in creating and continuously improving a lean supply chain. The proposed value stream mapping tool is tested in the case of the Ford Motor Company plant in Taiwan. Their findings highlight the resulting performance improvement across a number of supply chain management criteria including Overall Equipment Effectiveness (OEE), total working time and value versus non-value adding activities. The impact of lean manufacturing and supply chain management practices on the business performance of MTO and MTS firms is the main focus of Olhager and Prajogo (2012). Using data from 216 Australian manufacturing firms they conclude that whilst MTO firms emphasise more on supplier integration, MTS firms seek to improve their business performance through investment on internal lean practices and rationalisation of their supply chain. This finding is consistent with the widespread view that lean practices are mainly applicable to MTS systems.

Several empirical studies considering the adoption of LM emphasise on its production scheduling and control tools. Mukhopadhyay and Shanker (2005) study the implementation of pull control and the Kanban system in the production line of a tyre manufacturing plant. The first stage of their approach involves housekeeping techniques, employee training, set-up time reduction, layout improvements and finally the implementation of quality and visual control. With the supporting infrastructure in place, the second stage of the implementation process concerns the adaptation and

introduction of the Kanban system. The benefits resulting from the adoption of LM are cost reduction resulting from reduced WIP levels, increased output, minimisation of machine downtime and number of defects.

The implementation of lean manufacturing principles in a mass customisation boat manufacturer is considered by Stump and Badurdeen (2012). Product flows through linear fabrication and assembly stages and the company implements an assemble-to-order policy. They find pure pull production control difficult to implement throughout the entire process due to high product variety. Instead a hybrid push/pull production control mechanism is tested through simulation and found to lead to significant lead time and WIP efficiencies.

Van der Krogt et al. (2009) employ constraint-programming to model the application of lean scheduling tools in two industrial case-studies. The first case-study concerns the implementation of pull production control tools in a manufacturing company producing health care products. However, the proposed constraint-based reasoning model is only applied to a simple two-stage process. The effect of manufacturing cells is explored in the second case study which involves a telecommunications manufacturer. Apart from underscoring the strength of scheduling and the need for its full integration into any lean system, these case studies confirm significant reduction in inventory levels following the implementation of lean scheduling tools. The synergistic effect of LM and cellular layouts is also considered by Pattanaik and Sharma (2009). Their empirical research focuses on a case study involving a manufacturing unit which assembles missile components. Their approach involves the formation of part families based on routing commonality, the determination of flow rate and workload balancing. Their findings suggest that though the optimisation of intra/inter-cell flow, lean cells exhibit reduced flow, transportation and waiting times.

Anand and Kodali (2009) develop an Analytic Network Process (ANP) model designed to compare LM with Computer-Integrated Manufacturing (CIM) across a range of criteria which concern productivity, quality, cost, delivery, morale, flexibility and innovation. The ANP is applied in the case of an Indian HVLV valve manufacturer. The case study findings confirm that the firm can increase its competitive advantage by embracing LM.

Lean practices are recognised as significant operations management tools for manufacturing firms in emerging economies. Panizollo et al. (2012) carry out empirical research which considers four Indian manufacturing firms which adopted LM. A wide range of performance criteria is considered. Quantitative criteria mainly concern



internal factory performance e.g. throughput time, WIP, set-up time requirements, scrap and rework. Qualitative measures are used to assess the firms' external performance and relationships with suppliers and customers. Their findings demonstrate significant performance improvement in all the four industrial applications considered.

From the above it becomes evident that most empirical research on LM showcases successful lean implementations. However, there are a few studies which discuss failed attempts to implement leanness. Scherrer-Rathje et al. (2009) consider the adoption of lean manufacturing by a food processing machines manufacturer. The initial attempts to implement leanness prove to be unsuccessful. Their detailed analysis of lessons learned demonstrates the failure of the lean implementation project which lacked clear mission, coordination, the commitment of senior management and employee engagement. These shortcomings were addressed in a second implementation project in which lean manufacturing delivered the expected benefits of reduced throughput time and manufacturing costs.

Turesky and Connell (2010) study the unsuccessful implementation of LM in a UK manufacturer of pumping components. They argue that failure to systematically plan lean change initiatives coupled with lack of communication, insufficient investment on employee training, weak management support and resistance to change led to the derailment of the lean implementation project.

The findings of studies considering unsuccessful implementations of LM are fairly consistent. Failure is attributed to the deficiencies of the implementation project, not the lean paradigm itself. Reviewing the low adoption rates of lean manufacturing in the UK and the number of failed implementations, Bhasin (2012) seeks to shed light on the main barriers preventing successful lean transformation. Empirical data collected through a questionnaire survey and case studies attribute most of the unsuccessful lean implementation to inertial forces resisting change. The study highlights the need for a strategic approach which promotes cultural change, top management support, employee buy-in and a strong sustainability focus.

## **2.9 Chapter summary**

LM has central importance to this thesis which aims to test the applicability of lean production scheduling and control techniques in non-repetitive production systems. For this reason, Chapter 2 presents extensive research into the lean paradigm with the

view to develop strong insight into its overarching principles and constituent components.

Initially, the chapter sets to investigate the origins of leanness in the post World War 2 Japanese manufacturing. This historical overview is intended to unveil the context in which the precursor of LM, i.e. the TPS was conceived. The discussion identifies challenges faced at the time by Toyota and its focus on waste elimination as a means of securing their future in an extremely volatile economic environment. A wide spectrum of TPS tools is reviewed drawing attention to JIT production and its prerequisites including mixed model sequencing, workload balancing, production in small batches, set-up time reduction and the notion of production flow.

TPS and JIT were initially confined within Toyota and its supply chain until the second oil crisis. It was mainly then that Toyota's resilience and ability to sustain its growth attracted the attention of its national and international competitors. The chapter points to landmark publications by Sugimori et al. (1977) and Ohno (1988) and their influential role in the dissemination of TPS in Japan. It proceeds to explore initiatives led by research and professional groups including the IMVP and RMG which boosted the diffusion of TPS in western manufacturing and led to the emergence of the LM paradigm.

The evolution of leanness is reviewed by analysing the central notions of waste and value within the context of the lean enterprise, that is, the most contemporary form of the lean paradigm. In line with the lean enterprise model, lean thinking is extended beyond the shop-floor into the product design, R&D, HR and other departments of a manufacturing company as well as its procurement, marketing and sales functions which govern the relationships with its supply chain and customer base.

The chapter further focuses on the implementation of LM and attempts to identify the complete array of goals, principles and techniques that need to be adopted to allow a successful lean transformation. The analysis reveals a plethora of schemes proposed for the classification of TPS, JIT and LM tools and practices. The rationale for embarking on the lean transformation journey is discussed by exploring the financial and manufacturing performance differential resulting from the adoption of leanness. Performance metrics and benchmarking schemes introduced to support manufacturers in assessing their degree of leanness are also considered.

The final sections of this chapter review empirical research concerned with recent industrial applications of LM. These overall produce overwhelming evidence in favour of the success of LM although a few failed implementations are also identified. Despite

the growing support LM received over the years, it is surrounded by a number of misconceptions. These relate to the overall scope of the lean paradigm, its potential in relation to rival manufacturing paradigms and limitations of its applicability within manufacturing and other sectors.

Overall, the extensive review of LM has produced a number of significant findings, which are summarised below:

1. The continuing interest of academics and practitioners in LM and the considerable number of recent industrial applications of LM is a clear manifestation of its currency.
2. Originally introduced as the TPS, leanness has not remained static. This would be incompatible with one of its fundamental principles, i.e. that of continuous improvement. Leanness itself has over the years evolved into its current form represented by the lean enterprise model.
3. The precise nature of LM is still a subject of great controversy. Leanness has been described as a set of goals, methods, processes, tools as well as a philosophy, strategy, program and mindset. Through the historical overview presented in this chapter, it becomes evident that leanness is both a manufacturing and business philosophy. This philosophy sets the long term strategic objectives e.g. waste elimination, value maximisation, continuous improvement etc. that lean adopters strive to achieve.
4. From the operational perspective and similarly to its precursor i.e. the TPS, LM is a multi-faceted production system. Its true power lies in the synergistic effect of its complementary constituents. Several of these constituents, for instance, setup time reduction using SMED techniques are important prerequisites for its successful operation as they provide the necessary infrastructure for other key lean elements, e.g. JIT production and pull control.
5. LM is still being discounted to a manufacturing toolbox. This myopic approach has led to many piece-meal and ad-hoc implementations of some of its techniques. Unless LM is embraced holistically, its true potential is compromised.
6. A lot of the controversial aspects of LM can be attributed to the lack of a universal definition of what constitutes leanness. Definitions and classification schemes attempting to differentiate between lean principles and techniques and organise them into clusters abound and cause further ambiguity.
7. This chapter has produced considerable evidence showing that in most cases, failed implementations of LM are due to poor organisation and planning of the

lean implementation project. This review highlighted the importance of clear mission, top management support, employee engagement and training, good communication, commitment to the lean ethos, continuous improvement and cultural change as key preconditions for a successful lean transformation.

8. Reported failed implementations of LM are also attributed to the lack of perseverance and strong sustainability focus. The lean implementation project does not have a definitive end. It is an ongoing journey.
9. There still exists anecdotal evidence regarding the suitability of leanness for non-repetitive manufacturing systems. HVLV production is argued as one of the areas in which leanness has limited applicability. This thesis aims to test the transferability of LM into non-repetitive, non-serial HVLV production systems.
10. The comparison of LM with rival manufacturing paradigms has shown that in their majority the latter are founded on the principles of LM. Rival systems are also often compared to an outdated and narrow perception of the lean production model.
11. The numerous successful implementations of LM provide substantial evidence in support of its world-class manufacturing status and ability to sustain strategic competitiveness.

As production scheduling and control are at the focal point of this thesis, Chapter 3 reviews hierarchical production planning and control systems and the functions performed in their context. The impact of shop-floor layout, product diversity and demand response policy on scheduling and control decisions is considered. Scheduling and production control systems designed for repetitive flow-shops are reviewed and contrasted to those suitable for non-repetitive job-shops. The most prevalent forms of production control, namely push and pull are discussed and applications of pull control in non-repetitive production lines are examined in detail.

### **3 Production Planning and Control (PPC)**

Manufacturing firms rely on schedules to satisfy customer demand. Failure to meet promised due-dates compromises the quality of customer service and can lead to irreversible loss of customer confidence. Effective schedules allow firms to utilise their resources efficiently. They can free up capacity which in turn enables firms to be versatile and agile in the way they respond to customer orders. Good scheduling brings competitive advantage in a fast-changing manufacturing sector facing the immense pressures of globalisation.

Scheduling problems aim to satisfy multiple conflicting objectives. Their combinatorial nature results in a vast solution space. Scheduling is performed in volatile production environments where unexpected events can cause deviations from established production plans. Control is a function integrated with scheduling to monitor the execution of plans. It ensures work flows through work centres as planned. One of the latest innovations in the area of operations control concerns pull control mechanisms introduced in the context of JIT. Similarly to JIT, pull control was designed specifically for repetitive mass production systems. The purported success of pull control is the main driver behind the investigation of the feasibility of its extension to non-repetitive production systems.

This chapter reviews the scope of operations scheduling and control and discusses functions performed in their context. Initially, it draws attention to product volume and variety and the impact these have on the way manufacturing organisations configure their production systems and schedule their operations. Scheduling is reviewed in the context of repetitive and non-repetitive production systems, i.e. flow-shop and job-shop environments respectively. The review points out intriguing commonalities. It further enables the development of a conceptual job-shop scheduling framework. The discussion extends to main forms of production control, focusing mainly on pull control. A detailed analysis of the operating principles of pull control mechanisms is presented followed by a comparative review of their performance. The review identifies three pull control mechanisms which remain at the focal point of research to date. The main contribution of this chapter to the thesis is the conceptual scheduling framework which coupled with the three identified pull control mechanisms provide the design parameters for the agent-based simulation model developed in chapter 5 to test the operation of pull control in job-shops.

The next section reviews three main factors, namely nature of demand, order fulfilment policy and shop-floor configuration which influence scheduling practice. Section 3.2 provides an overview of the planning and control hierarchy in which operations

scheduling is carried out. It stresses the reliance of scheduling on outputs generated by MRP planning systems. Loading and sequencing formulated as optimisation problems in the flow-shop and job-shop scheduling literature are discussed in section 3.3. Section 3.4 provides an in-depth analysis of the mechanics and performance of pull control policies. Finally, section 3.5 draws conclusions to this chapter.

### **3.1 Contextual factors influencing production scheduling**

Scheduling affects every aspect of human venture from simple everyday tasks to complex operations and services across most industrial sectors. Gupta (2002) admits that due to its multifaceted nature, scheduling classifications and definitions abound.

According to Kempf et al. (2000, p. 204) in manufacturing settings, production scheduling is concerned with *“assigning scarce resources to competing activities over a given time horizon to obtain the best possible system performance”*. Manufacturing resources generally comprise machines, tooling, material handling systems, human operators etc. however, this analysis will specifically focus on machines. Activities are manufacturing operations that require processing on machines. They are determined by decomposing the products that need to be manufactured within a certain time period into their respective sets of operations. These need to be processed in predetermined sequences (imposed by technological constraints) on certain types of machines. Due to resource limitations that characterise every production system, activities that require scheduling often have to compete for specific machines especially those which tend to be heavily utilised. The output of the scheduling process, namely, an operations schedule, specifies the timings and order that activities need to be carried out by machines and influences the manner in which WIP will flow through the system.

Scheduling generates allocations of activities to available machines. The aim in scheduling is to optimise system performance with respect to a wide range of often conflicting objectives e.g. on-time completion of operations to meet due dates, maximisation of machine utilisation, and minimisation of WIP levels within the system (Wiendahl et al., 2005). Managing the trade-offs between these objectives and seeking an optimal or near optimal scheduling solution in a vast solution space has led to the recognition of the intrinsic complexity of scheduling problems and their classification as Non-Polynomial (NP) hard (Leung, 2004).

Wild (1994) suggests that the nature of scheduling problems and associated solution techniques are influenced by the following three factors: (i) the nature of demand for manufacturing products, (ii) the orientation and order fulfilment policy of the production

system (iii) the type of manufacturing process and its effect on shop-floor configuration. These influencing factors are discussed further in the following sections.

### **3.1.1 Nature of demand**

Manufacturing is the physical transformation of raw materials (inputs) to goods (outputs) sold to customers. Goods such as automobiles, electrical appliances, personal computers etc. are end products with complex structures consisting of various sub-assemblies, components and parts. Demand for such integral components is dependent on the demand for finished products. In contrast, demand for finished goods is independent and cannot be established based on demand information already available (Martinich, 1996).

This distinction is particularly relevant in scheduling. Operations scheduling is primarily concerned with manufacturing activities associated with dependent demand items, that is, the processing of raw materials and their progressive transformation into parts, components and sub-assemblies of products. Independent demand inventories are normally controlled by periodic review policies designed to replenish inventory when levels reach predetermined reorder points. Conversely, the high number of dependent demand items handled in any given factory setting call for an entirely different approach, in fact one which is capable of handling large volumes of data. MRP is the computerised system typically used to manage dependent demand inventories (Jacobs and Weston, 2007).

### **3.1.2 Production orientation and order fulfilment policy**

Demand for finished goods is either generated externally in the form of orders placed by customers or created internally so that manufacturing products can be stocked to meet future customer orders. In the first case, customers specify the range of goods to be manufactured and timing of production. Internal scheduling is performed to ensure production of the goods ordered is completed on time to meet due dates and is therefore directly influenced by external demand. Wild (1994) classifies such scheduling systems as externally oriented and contrasts them to internally oriented systems where production is scheduled on a speculative basis in anticipation of future demand. It is evident that whilst externally oriented systems need to be able to respond to demand quickly, there is higher flexibility in internally oriented systems where scheduling of activities is not time-limited.

Proposing a similar classification, Markland et al. (1998) maintain that production systems can be differentiated based on the amount of processing they perform following receipt of orders. Manufacturing companies where procurement of raw materials and fabrication of parts are only instigated once customers have placed firm orders are known to implement a MTO policy. Such a policy of responding to demand is mostly appropriate for companies capable of customised production offering a wide range of “tailor-made” products.

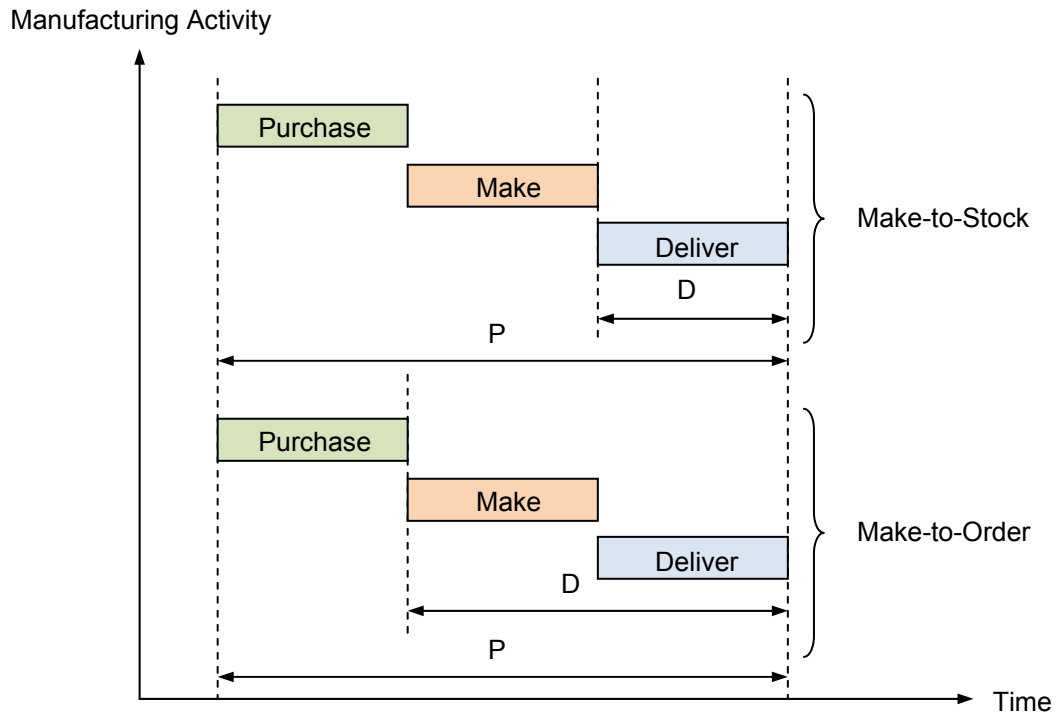
A policy diametrically opposed to MTO is MTS, adopted by manufacturing companies which offer a limited range of highly standardised products. As the name of the policy suggests, production aims to create an inventory of finished goods which is used to fulfil customers’ orders. Therefore, the processing performed by such systems is not associated with firm but rather anticipated demand. MTS companies rely heavily on forecasting models which use historic sales data to estimate the product mix and volume as accurately as possible. Following the convention proposed by Wild (1994), MTO production systems can be classified as externally oriented whereas MTS factories and their scheduling operations are internally oriented.

Porter et al. (1999) identify three more classes of order-driven policies. Assemble-to-Order (ATO) implies that adopting firms produce standardised modular components which are assembled according to customers’ specifications to offer model variations of the same finished product. The two main types of operations performed by ATO systems are fabrication and assembly, with the first aiming to create stock and the second initiated in response to customer orders. This explains why ATO systems are considered to be a compromise between MTS and MTO. Engineer-to-Order (ETO) and Design-to-Order (DTO) systems are less frequently adopted. They rely on more customer input into product development and customisation. ETO companies produce a standard product range with optional modifications which are available upon request. DTO systems allow individual clients to get involved in the research and development of products thus maximising their uniqueness. Nevertheless, ETO and even more so DTO result in considerable lengthy design and production lead times.

Focusing on this particular point, Slack et al. (2010) study the underlying differences between these policies by examining the total amount of time customers need to wait between placing an order and receiving the finished products. In doing so, they compare the production throughput time (P) which is the total time required to procure raw materials, manufacture and deliver the product with demand time (D) that is, the length of time between placing an order, processing and transporting it to the customer.



As illustrated in Figure 3.1, the graduation from make-to-stock to make-to-order results in a lower P:D ratio pointing to longer customer waiting times.



Source: Slack et al. (2010)

Figure 3.1 P:D ratios in different demand response policies

### 3.1.3 Manufacturing process and shop-floor configuration

Stevenson (2006) recognises that scheduling functions are highly dependent on the volume of production, which in turn largely determines the type of manufacturing operations (processing) performed at a given manufacturing facility. He broadly categorizes production systems into high, intermediate and low volume. There is an inverse relationship between production volume and product variety. Low volume systems are mainly associated with high mix production whereas high volume systems are dedicated to the production of small ranges of goods. Mass production lines are typical examples of high volume systems.

Reviewing developments in contemporary manufacturing operations management, Gunasekaran and Ngai (2012) highlight a shift of focus from medium level and variety production in the 1970s to maximising variety and minimising volume from 2010 onwards. This is remonstrated on the basis of the challenges facing modern manufacturers with competitive advantage linked to product individualisation as opposed to customisation. In order to accommodate different levels of production

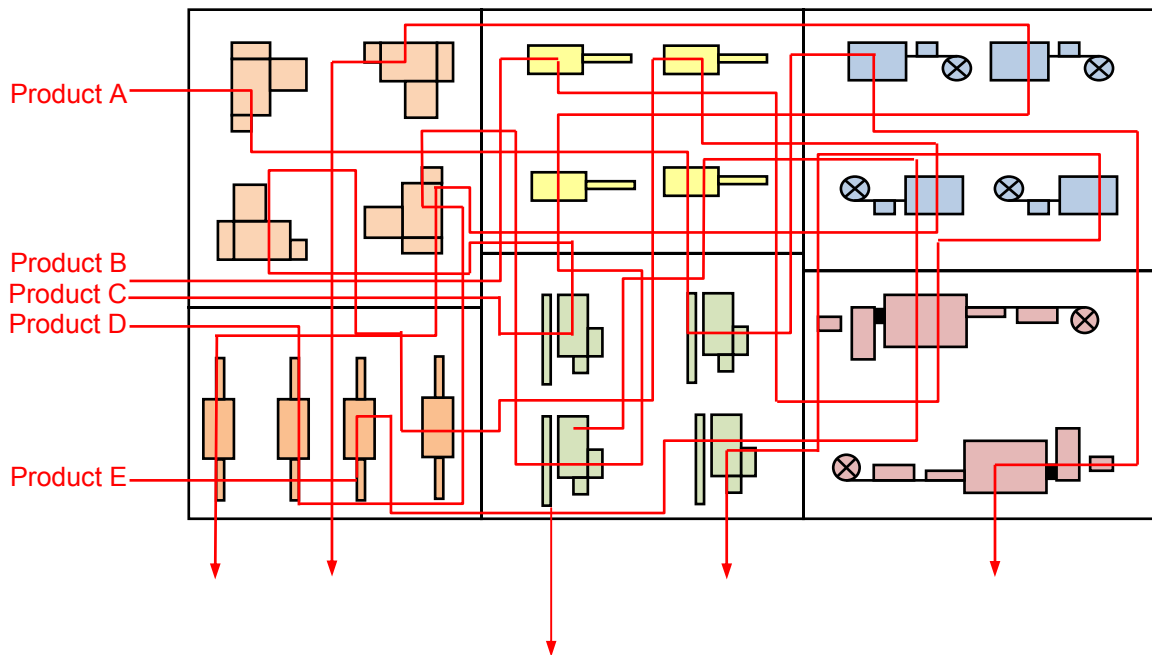
volume and variety, processing equipment needs to be physically arranged into appropriate production layouts.

Layouts by fixed position are typically encountered in project settings. Project manufacturing is concerned with the protracted production of unique (often one-off) large scale and high value products (Smith, 2008). The production of heavy machinery, aircrafts and ships is undertaken in such layouts. Due to the nature of the end product, the latter remains stationary whilst all necessary resources e.g. labour, raw materials, equipment etc. move around its fixed position.

Process manufacturing requires functional layouts, typically job-shops where general purpose machines performing similar processing operations e.g. drilling, milling, grinding etc. is grouped together in discrete workstations occupying designated areas of the shop-floor. Products typically manufactured in job-shops are machine tools or components for a wide range of industries including aerospace (Scallan, 2003). Contrary to project manufacturing, in process layouts the product moves through different sections of the factory in batches and the actual routing is determined by the sequence of processing steps that need to be completed on different workstations. Such shop-floor configurations allow great flexibility as they can accommodate the production of a high variety of products requiring work of a jobbing nature in small quantities. Processing large product ranges practically results in disorderly workflow patterns which are quite complex to control. Furthermore, whilst some machines are idle, in other workstations queues may be formed by jobs competing for machines. Such occurrences are typical of the intermittent production that takes place in process layouts. Figure 3.2 illustrates the flow diversity of five products manufactured in a job-shop comprising six workstations with parallel machines.

Product layouts are commonly associated with flow line systems. Flow lines are series of tightly connected workstations arranged sequentially according to the operations that need to be completed to manufacture a small variety of products in very large volumes (Stevenson, 2006). These products are fairly standardised i.e. there are few variations between them and subject to high but stable demand. Production in this type of systems is often uninterrupted and quite simple to control. Flow systems can be dedicated to the production of one product (single model production lines) or a small range of products (mixed model assembly lines). Product layouts are typically used for the mass production of electrical appliances and automobiles. Extreme cases of this type of layouts are continuous flow systems dedicated to the production of large quantities of non-discrete products. Continuous manufacturing is encountered in a wide

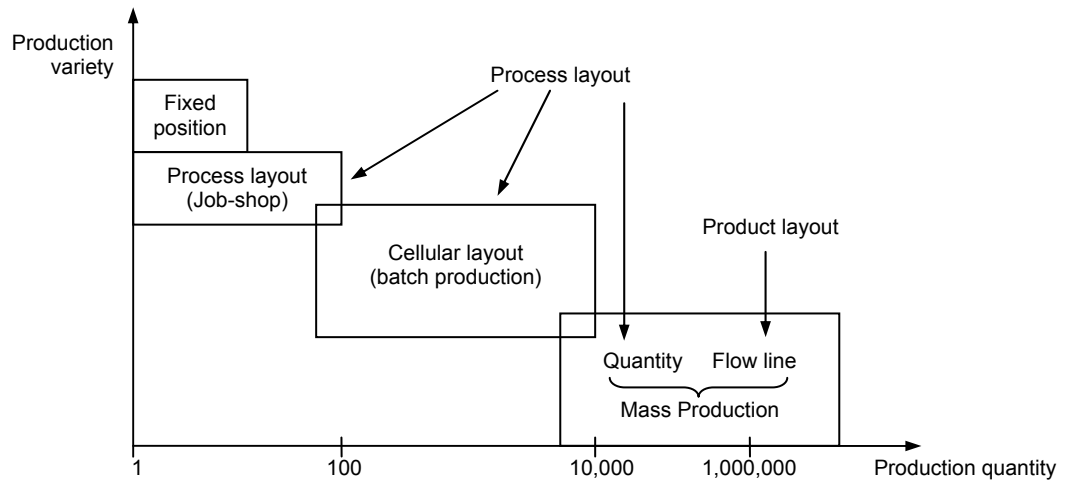
range of applications including petroleum refineries, chemical processing, pharmaceutical, steel making, paper and food processing.



**Figure 3.2 Workflow diversity in job-shops**

Cellular configurations offer a compromise between process and product layouts. Cells are groups of workstations used for the production of a small family of similar products (Dos Santos and De Araújo, 2003). Product families are typically formed in line with the classification principles of GT (Özbayrak and Papadopoulou, 2004). Cellular layouts are capable of producing greater volumes than job-shops whilst they can also handle a greater product variety than mass production. GT alleviates various shortcomings of job-shops such as the requirement for specialist supervision as typically one operator can oversee the operation of an entire cell.

Groover (2008) maintains that the boundaries between the four production layouts discussed above are not fixed. He suggests that low production systems manufacture products which do not exceed 100 units per year. Medium volume production is associated with outputs ranging from 100 to 10,000 units per product and year, whereas high volume systems are capable of processing quantities exceeding 10,000 units per product annually. The effect of production volume and variety on system configuration and processing accommodated is depicted in Figure 3.3. A brief overview of the characteristics and performance of process and product configurations is presented in Table 3.1.



**Source: Groover (2008)**

**Figure 3.3 Types of production layouts suitable for different levels of production volume and variety**

The premise of this thesis is that scheduling and control techniques originally designed for repetitive mass production systems are transferrable to non-repetitive manufacturing systems. The analysis of factors influencing the selection of the most suitable production system is relevant to this investigation. It was demonstrated that the order fulfilment policies adopted by manufacturers to respond to demand determine the type of production system and its shop-floor layout. Repetitive manufacturing systems configured as flow lines are conducive to mass production. They are mainly preferred by manufacturers implementing a MTS policy. In contrast, non-repetitive manufacturing systems, specifically job-shops, provide the necessary flexibility to support customised production and the operation of a MTO policy.

The section drew a distinction between dependent and independent demand. It was pointed out that independent demand is forecasted to determine long-term production plans based on which detailed requirements for dependent demand items are subsequently ascertained. Whilst independent demand affects the decisions made in production planning discussed in section 3.2, scheduling and control are solely concerned with dependent demand items. The shop-floor configuration of flow lines and job-shops were reviewed in detail in this section. The discussion contributes to our understanding of their characteristics and implications for the scheduling and control functions analysed in section 3.3.

**Table 3.1 Comparison of process and product layouts**

<b>Characteristics</b>	<b>Process layout</b>	<b>Product layout</b>
Application (production volume/variety)	Large product range. Low production volumes.	High production volumes. Limited product range.
Product type	Customisable.	Standardised.
Processing type	Intermittent batch production. Non-repetitive.	Continuous mass production. Repetitive.
Configuration	All similar equipment grouped together.	Serial arrangement of equipment according to product routing.
WIP flow pattern	Jumbled.	Common for the same product type.
Machine type	General purpose, semi-automatic.	Special purpose, automatic.
Machine set-ups (tooling change and reprogramming)	Frequent. Required every time a different product is processed.	Remain unchanged over longer periods.
Required labour	Specialist operator.	Semi-skilled labour.
Provision of services (water, power, waste removal)	Simple as similar equipment with same requirements placed in the same area of the shop-floor.	More complicated
Material handling requirements and systems	High requirements and costs. Fork-lifts and hand-carts.	Low requirements and costs. Closely interlinked workstations allow the use of conveyor belts.
Machine break-downs and preventative maintenance	Breakdowns can be tolerated as multiple equipment of the same type is available. Maintenance can be performed during production hours.	Breakdowns will halt production in downstream sections of the line.
<b>Performance</b>		
WIP	High levels of WIP resulting from jobs queuing in front of workstations.	Low.
Throughput time	Long.	Short. Less transportation time and no queuing of jobs.
Machine utilisation	Average.	Quite high.

### 3.2 PPC framework

Groover (2008, p. 796) asserts that *“PPC is concerned with the logistics problems that are encountered in manufacturing, that is, managing the details of what and how many products to produce and when, and obtaining the raw materials, parts and resources to produce those products. PPC solves these logistics problems by managing information. The computer is essential for processing the tremendous amounts of data involved to define the products and the manufacturing resources to produce them and to reconcile these technical details with the desired production schedule”*.

PPC is typically performed in three levels. Long-term (strategic) planning determines the organisation’s capacity and product design and mix for a period ranging from 1-5 years ahead. It involves decisions about the expansion of existing facilities and location

of new factories, selection of production technology, design of product and work systems. Intermediate-term (tactical) planning is concerned with determining the level of aggregate production and inventory. Decisions regarding workforce levels and work patterns e.g. overtime, additional shifts, subcontracting are also made during this planning stage and concern the next 3-12 months. Martinich (1996) points out that the intermediate planning effectively interfaces strategic long-term plans with day-to-day operations schedules. The latter consist of workforce schedules and machine loading charts developed in short-term (operational) planning. Machine schedules are established by determining production lot sizes, assigning jobs to machines and defining the order in which operations will be carried out.

Heizer and Render (2004) explain that strategic plans requiring capital expenditure are typically authorised by executive managers with input from peripheral departments including finance, Research and Development (R&D), HRM, marketing and sales. Tactical plans are developed by middle-level operation managers who liaise with procurement, production and logistics. Finally, short-term plans are determined and overseen by shop-floor supervisors and foremen.

Guinery and MacCarthy (2009) argue that PPC hierarchies are consistent with the logic pertaining to every planning system. Top level aggregate plans are broken down into more detailed and accurate plans as they cascade into lower levels of the hierarchy. In this manner, the PPC hierarchy provides an effective structure for decomposing information and knowledge available to the highest levels of the hierarchy into instructions that determine decisions made lower down. This presumes that some element of autonomy is delegated to lower level functions responsible for the implementation of plans in daily operations. Communication, close interfacing and coordination between different hierarchical levels becomes even more important when it comes to dynamically and rapidly responding to unforeseen external changes which call for plans to be revised.

Planning and control are often treated as integrated functions (Groover, 2008). Slack et al. (2010) subscribe to this view. They argue however that the scope of planning differs significantly from that of production control. Plans are developed using specific information on a wide set of parameters. They imply an intension - not certainty – to run production in a predetermined way. Unexpected changes to internal parameters, e.g. machine breakdowns or disruptions imposed externally, e.g. late delivery of raw materials render some of the assumptions on which plans were developed invalid. Corrective action is necessary to respond to variability, adapt plans to changes and bring operations back on track to meet production objectives. Production control

ensures schedules are closely monitored and re-aligned to preset objectives (Koeningsberg and McKay, 2010).

### **3.2.1 Aggregate planning and the Master Production Schedule (MPS)**

Intermediate planning establishes optimum production output levels such that total demand for all products can be met by utilising effectively the total available capacity resources (Timm and Blecken, 2011). In order to match demand with capacity, intermediate planning reviews inventory status, production patterns (regular/overtime/subcontracting), workforce levels and order fulfilment strategies (backordering). Liang (2007) stresses the importance of intermediate planning which informs decisions made in lower levels of the planning hierarchy.

Plans for the intermediate planning horizon concern aggregated product categories. For this reason, intermediate planning and aggregate planning are two terms used interchangeably. Aggregate plans concern major lines of current and new products (Groover, 2008). Stevenson (2006) justifies the need for aggregate planning by highlighting the importance of reconciling demand with capacity before developing detailed plans. He further admits that as the planning horizon in intermediate planning is considerably long, it is not possible to estimate accurately and with certainty the volume and timing of demand for individual items. Even if sophisticated forecasting models were used to facilitate this, it would not be desirable to compromise the system's flexibility and responsiveness to the changing needs of the market.

Markland et al. (1998) explain that aggregate planners commonly adopt two strategies in order to balance capacity and fluctuating demand. A chase strategy aims to reactively align production output with the profile of demand over the planning period. To achieve this, planners adjust production rates and workforce levels or use subcontracting. This strategy makes limited use of inventory to meet demand requirements. A level strategy on the other hand, ensures product rates and workforce levels are stable throughout the planning period. Inventories accumulated in periods of low demand are used to fulfil a backlog of orders created during peak demand. Moreover, hybrids of the aforementioned two strategies may be more appropriate in certain production systems. Sophisticated mathematical and simulation-based approaches have been developed to generate aggregate plans. However, it is accepted that simple trial-and-error techniques are more frequently used in practical applications (Stevenson, 2006).

Martinich (1996) adopts an aggregation model reported in the relevant literature according to which end products with similar processing and machine set-up requirements can form product families which in turn can be aggregated according to their cost structures, inventory holding costs and demand patterns to form product types. Following this convention, product types may for example comprise refrigerators and freezers for commercial or household use whilst different families may include integrated refrigerators/freezers, frost-free refrigerators/freezers, simple refrigerators etc.

However, the aim in aggregate planning is to roughly balance capacity with demand. Once this is achieved, it is important to disaggregate plans so as to interface firm and projected demand with short-term operational schedules (Jamalnia and Feili, 2011). Martinich (1996) discusses some of the overarching principles of disaggregation by suggesting that when disaggregating plans at the product family level, it is important to plan production of families with expensive or time consuming set-ups less frequently than those which are less expensive to set-up. Disaggregation of families into end items should take into account current and projected inventory levels so that near stock-out items are produced with priority.

The intermediate production plan is decomposed into a very specific schedule known as the MPS. The MPS determines the timing and quantities of specific end items produced every week. It combines firm customer orders, forecasts, urgent orders, and inventory status reports to re-evaluate total demand requirements. If these can be met by the available capacity, production orders are entered into the MPS with urgent orders placed in the earliest possible available slots (Gaither and Frazier, 2002).

The MPS takes into account the cumulative lead times comprising procurement of raw materials, fabrication of components and assembly of end items. For several products, lead times can be substantial and therefore the MPS needs to be a medium-term schedule with a planning horizon spanning from a few weeks to several months (Groover, 2008).

Waller (2003) explains that the MPS is typically divided into four stages or zones called time fences. Changes to the first couple of weeks of the schedule can be quite disruptive as most of the available capacity resources are already committed. This period is often described as frozen, a term indicative of the rigidity of the production plan. Permission from higher levels of the organisation will need to be sought to modify plans during this period. Schedule stability is also important in the next two to three weeks when the schedule is considered to be fixed. Modifications can still be



disruptive, but are allowed under exceptional circumstances. The third segment of the MPS is viewed by operations managers as full, meaning that all the available capacity is now fully allocated. This covers the next couple of weeks where the impact of changes is less dramatic. The final (fourth) stage of the MPS is open. This implies that capacity is available to accommodate the production of new orders which are generally slotted in this phase of the schedule. Tang and Grubbström (2002) admit that the quality of the MPS is determined by the scheduler's ability to select an appropriate rolling horizon and replanning frequency.

### **3.2.2 Material Requirements Planning (MRP)**

Having decomposed the aggregate plan into an MPS, that is, a specific production plan which determines the timing and quantities of individual final products, the next step in the planning process involves converting the MPS into a detailed schedule for the raw materials and parts required to produce these. Stevenson (2006, p. 640) describes MRP as a computer-based system designed to achieve this. Starting from the due dates for final products, MRP works backwards and uses lead times and order policies to compute the required quantities of dependent-demand inventory and when these need to be manufactured or ordered from suppliers. He further admits that MRP is *“as much a philosophy as it is a technique, and as much an approach to scheduling as it is to inventory control”*

It is evident from the above definition that the aim of MRP is to develop effective production and purchasing schedules. This practically means that MRP determines materials to be procured from external suppliers, materials that need to be manufactured internally and when to place orders for these inside and outside respectively. The primary input of the MRP system is clearly the MPS. Once the MRP processor acquires information on which final products are required, when and in what quantities, it needs additional data in order to convert these into material requirements. This data is contained within the BOM file. The BOM provides information on the product structure listing in detail the exact quantities (number of units) of subassemblies, components and raw materials which make up the final product. Barnes (2008) points out that this information needs to be combined with up-to-date inventory status reports. The inventory records file (or item master file) provides time-phased records of inventory status taking into account on-hand inventory, scheduled receipts and planned order releases (Groover, 2008). It is further recognised that MRP needs to be interfaced with capacity planning so that the generated schedules do not exceed the available production capacity. The accuracy of the aforementioned files is

crucial as errors and outdated information will compromise the quality of MRP schedules.

The MRP processor explodes the end product requirements contained within the MPS into successively lower levels of the product structure using information from the BOM. However, the quantities of materials computed at this stage represent gross requirements as they do not take into account current and projected levels of inventory. A procedure called netting is implemented at this point. The sum of on-hand inventories and scheduled receipts (quantities on order) is subtracted from the gross final product requirements specified in the MPS to compute the net material requirements in line with the formula below:

$$\begin{array}{rcccc} \text{Net} & & \text{Gross} & & \text{On-hand} & & \text{Scheduled} & & \\ \text{requirements} & = & \text{requirements} & - & \text{inventories} & - & \text{receipts} & & \end{array} \quad (3.1)$$

(Akillioglu and Onori, 2011)

Another complicating aspect of MRP is that netting generates delivery requirements which must be offset, taking into account lead times to derive a time-phased material plan (Wacker and Sheu, 2006). Offsetting is the process by which planned order releases are determined. These are instructions to place orders for materials planned by the MRP system. If the materials needed are raw parts from suppliers, the planned order releases are purchase orders whereas work orders need to be released to authorise the manufacturing of parts produced internally. Planned order releases offset net requirements by the respective procurement and manufacturing lead times (Stevenson, 2006).

MRP systems produce material schedules using time periods of equal intervals known as time buckets. In this manner, MRP considers time as discrete intervals. All MRP computations including gross requirements, inventory status, planned order releases are specified for the time buckets of the planning horizon. Teo et al. (2011) draw attention to the dynamics of job-flow movements which take considerably shorter times than the typical MRP time buckets.

Jonsson and Mattsson (2008) argue that in addition to lead times, other important planning parameters in MRP systems are order quantities and safety stocks. Ho and Chang (2001) review lot sizing policies proposed for MRP including the Economic Order Quantity (EOQ) used in inventory management systems for independent demand and Lot for Lot (LFL), Fixed Order Quantity (FOQ) and Period Order Quantity (POQ). These appear to have different advantages in terms of minimising set-up times,

inventory holding costs and achieving economies of scale through shipping discounts, with none of the policies outperforming others under all the aforementioned conditions.

Gaither and Frazier (2002) justify the need to use safety stock due to uncertainty in demand and lead times. They acknowledge that although the requirements for lower-level items are generated internally and correspond to dependent demand, these can still be affected by changes to independent demand specified in the MPS. Safety stock is included in MRP computations as shown in the following formula:

$$\text{Net requirements} = \text{Gross requirements} - \text{On-hand inventories} - \text{Inventories on order} - \text{Safety stock} \quad (3.2)$$

It can be inferred from the literature that planned order releases are the most important instructions generated by MRP systems; however, the latter produce a variety of primary outputs. These include reports outlining planned order releases in future periods, rescheduling notices (instructing revisions of due dates of open orders), cancellation notices of open orders, inventory status reports. Secondary outputs include performance reports (indicating levels of item-usage, deviations from planned lead times etc.), exception reports (in case of late orders, defective items) and projected levels of inventory (Roy, 2005).

MRP systems are powerful computer applications able to keep track of large volumes of data. However, Barnes (2008) stresses that real-life applications of MRP systems operate in unpredictable dynamic environments. Changes such as late deliveries by suppliers, cancellation of existing orders or receipt of rush orders by customers, worker absenteeism, machine downtime and scrap can cause a chain reaction affecting all open MRP orders. Moscoso et al. (2010) describes this situation as MRP nervousness and admits that problems occur when the volume of rescheduling messages generated by MRP is such that shop-floor control is unable to react.

Advances in computing and information technology facilitated the expansion of the scope of MRP. Petroni (2002) admits the term MRP broadly encompasses all subsequent versions. Feedback loops were introduced in closed loop MRP to test the validity of plans against the available capacity. The next generation of MRP systems were databases for manufacturing resources planning known as MRPII (or MRP2) (Gupta and Kohli, 2006). These were used to plan and control a wider range of resources required for manufacturing including workforce and equipment. ERP systems are the most advanced MRP-based programs. Core business functions including procurement, marketing, accounting, finance, logistics, operations control are

interfaced into one seamless database increasing productivity and adding value to the quality of service provided (Ngai et al., 2008).

### **3.2.3 Capacity planning**

Capacity is the throughput in number of units per period produced by a facility (Heizer and Render, 2004). Obviously, the overall capacity of a production system is the cumulative productive capability of every single manufacturing resource e.g. operator, machine, workstation etc. that makes up the entire facility.

Markland et al. (1998) differentiate between design, effective and actual capacity. Design capacity as the term suggests, is the maximum output a production facility is designed to achieve. Operating conditions e.g. product mixes that require multiple changeovers and therefore increase machine idle times often limit the output a system can practically achieve, resulting in an effective capacity which is lower than its design capacity. Finally, the actual capacity achieved is a reflection of dynamic working conditions prevailing during a certain period, e.g. shortage of materials, worker absenteeism, machine failures which reduce the productive capability of the manufacturing system below its effective capacity. Therefore, the capacity an organisation can achieve is not only a function of the number and how technologically advanced resources are, but it depends heavily on how well the latter are operated and maintained.

As discussed in section 3.2, strategic capacity planning entails decisions which aim to adjust production capacity in the long term. These usually require significant capital investment and long lead times. Groover (2008) provides examples of such decisions:

- Investment on new equipment. Procuring more machines or replacing old machines with more advanced, higher productivity models can increase capacity.
- Construction of new plants or acquisition of existing plants from competing firms. Apart from increasing revenue and market share such decisions can lead to a significant increase of an organisation's overall capacity.

Nevertheless, with reference to the second point above, Barnes (2008) warns that capacity is location-specific, remonstrating it is not safe to aggregate the capacities of different facilities belonging to the same organisation. In other words, attempting to compensate for heavily utilised factories using the excess capacities of other under-utilised facilities may have serious transport cost implications. He also identifies that

capacity can be affected by constraints in all types of manufacturing resources e.g. material handling systems, storage space and so forth and therefore investment plans should also focus on the latter.

Waller (2003) recognises that operations managers use the following tools to increase capacity in the short-term:

- Adjusting permanent workforce levels. This may involve hiring additional workers.
- Hiring temporary workers. A tactic frequently used to cope with peak (or seasonal) demand.
- Altering the number of work shifts. Adding one or more shifts can increase nominal capacity but not necessarily productivity (mainly valid in the case of night shifts).
- Changing labour hours. This can be achieved by allowing weekend work or overtime.
- Increasing inventory levels. Stockpiling raw materials sourced from suppliers or WIP produced internally allows capacity to be stored and used in future periods.
- Subcontracting. An equivalent term for subcontracting is outsourcing. This implies letting of work to subcontractors. Although this may relieve some of the strain in the client organisation's resources it may also limit their control over quality and timely delivery of the products to customers.
- Allowing backordering. This involves accepting customer orders which however will be fulfilled with delay in periods when capacity cannot cope with high demand.

Capacity planning is an iterative process performed at all stages of production planning and control. Further to the strategic capacity planning discussed above, Ravindran et al. (2011) recognise that aggregate planning is performed in tandem with some element of medium-term capacity planning. This involves aggregating the capacity of the system in terms of man hours, machine hours and inventory available. The aim at this stage is to determine if aggregate demand can be met without violating existing capacity limitations.

Following this preliminary checking of capacity requirements, the aggregate production plan is converted into an MPS. The MPS produced at this stage may not necessarily be feasible. It is standard practice to run the MPS through the MRP processor in order to determine the corresponding resource requirements. This is a procedure known as Rough-Cut Capacity Planning (RCCP). RCCP aims to identify periods in which

capacity is exceeded (overloading occurs) (Gaither and Frazier, 2002). Despite being more systematic than the initial capacity calculations carried out when the aggregate production plan is determined, Tenhiälä (2011) accepts that the RCCP ignores lower level inventories, set-ups, routings and batch sizes. Consequently, even at this stage, there is no guarantee that the MPS can be met as due the intrinsic limitations of RCCP, there is limited insight into the loadings of workstations on the shop-floor.

Follow-up capacity calculations known as Capacity Requirements Planning (CPR) are carried out once the MRP schedule is available. Planned receipts generated by MRP coupled with routing, operation and set-up times are used to produce machine loading reports and forecast the total capacity required to achieve the MRP schedule (Segerstedt, 2006). If there is insufficient capacity, either the MPS must be revised or capacity adjusted using one of the short term tactics identified above.

The findings from the above discussion culminate in the simplified illustration of the planning and control hierarchy presented in Figure 3.4. This section adds to the thesis by pinpointing where scheduling fits in this hierarchy. It argues that scheduling does not function in isolation.

Quite on the contrary, scheduling decisions affecting short-term production plans depend on strategic and tactical planning performed at higher levels of a manufacturing organisation. The discussion centres on MRP and the computations performed in its context. It identifies the most salient information produced by MRP, that is, order releases which provide the necessary trigger for the scheduling process.

The review extends on aggregate planning and capacity planning which both feed into the MRP process. Capacity considerations are particularly relevant in the ensuing discussion of scheduling presented in section 3.3. Line balancing performed in the context of flow-shops and job-shop loading both concern the uniform distribution of workload to the available production capacity. Both push and pull production control analysed in section 4.4 aim to ensure work flowing through work centres does not exceed its available capacity.

### **3.3 Scheduling in manufacturing systems**

Chen and Ji (2007) argue that MRP merely generate order releases. Although these trigger production at the shop-floor level they provide no information about operation sequences. Moreover, MRP does not ensure a feasible production plan exists nor can handle the monitoring of the plan once this is put into implementation. These are typical functions performed in the context of scheduling which is carried out at the shop-floor

level. In section 3.1 it was stressed that scheduling is influenced by the nature of the production system. The following subsections present an overview of the scheduling performed in the process industries and project manufacturing prior to a more in-depth review of scheduling in flow systems and job-shops.

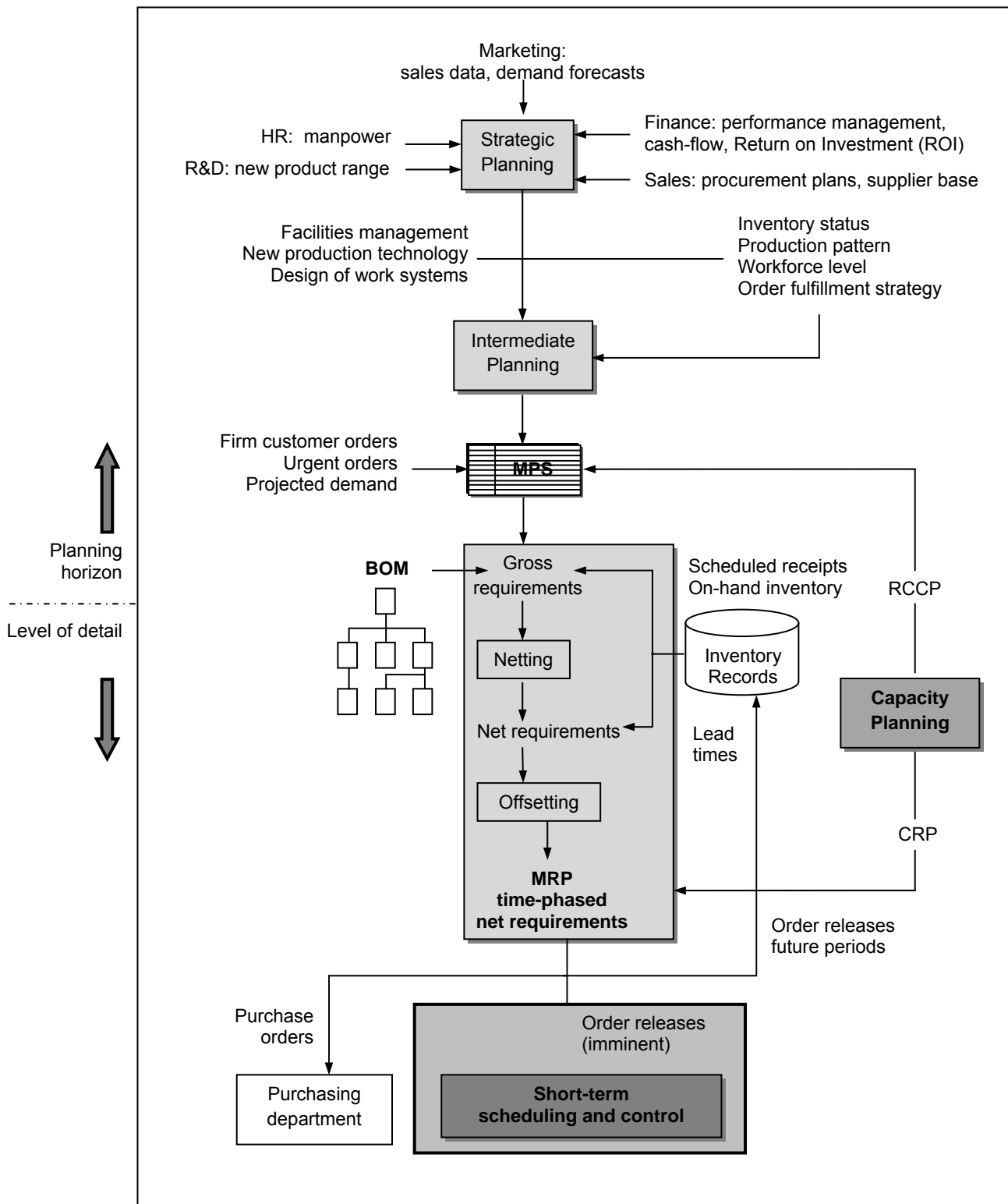


Figure 3.4. Planning and control hierarchy

### 3.3.1 Scheduling in the process industries

The terms “continuous manufacturing” and “process manufacturing” are often used interchangeably to describe a production system which operates 24 hours a day for long periods of time without a halt (King, 2009). The term process manufacturing further implies that the transformation of raw materials into final products is often the result of chemical reactions supported by other physical or mechanical means (Scallan, 2003).

In such systems, changeovers, i.e. switching between different products are lengthy as processing units may need to be flushed and inspected before they can be reconfigured to process the next batch (Rappold and Yoho, 2008). It is therefore not surprising that scheduling in continuous manufacturing is concerned with the determination of the product mix and production order quantity that minimises changeovers (Russell and Taylor, 2009).

Kallrath (2002) points out that scheduling process industries also involves planning shutdowns of production facilities in order to carry out maintenance. The Critical Path Method (CPM) is a technique developed in the 1950s by Remington-Rand and Dupont to schedule such maintenance shutdowns (Mouhoub et al., 2011). Maravelias (2012) accepts that CPM served as the precursor of network-based approaches currently employed to model process facilities. Floudas and Lin (2004) review processing networks capable of handling the complex modelling requirements of dissimilar production recipes. They discuss State-Task Networks (STN) where state nodes are used to denote raw materials, WIP and final products whilst task nodes represent the form of processing (separating, mixing or forming) products undergo.

David et al. (2006) examine the application of ERP-based scheduling systems in process manufacturing. They point out the significant limitations of these systems in coping with distinctive features of process manufacturing, namely the diverse nature of material flow, requirement for multiple product synchronisation in operations and shipping tolerances. Although their study is mainly focused on aluminium conversion industries, they argue their findings are relevant to other process industries. The limitations of scheduling systems developed specifically for discrete manufacturing are also noted by Maravelias (2012). He maintains early scheduling approaches treated continuous processing in a similar fashion to the way discrete jobs receive processing at different workstations, disregarding the need to mix and split batches in between processing steps.



Reviewing different scheduling solution methodologies for process manufacturing, Li and Ierapetritou (2008) highlight the limitations of discrete time approaches which apart from their restricted accuracy increase the complexity of the associated mathematical programming formulations. They discuss alternative models based on continuous-time representations which are more effective approximations or real-life applications. Mixed Integer Linear Programming (MILP) and Mixed Integer Non-Linear programming (MINLP) are the most commonly used modelling and solution methodologies for the scheduling of process facilities (Li et al., 2009).

### **3.3.2 Scheduling in project manufacturing**

Sharon et al. (2011) acknowledges that the main scheduling methodologies underpinning commercial software commonly used in the project manufacturing industry are the CPM and Programme Review and Evaluation Technique (PERT). Both techniques rely on the use of diagrams used to graphically represent projects as complex networks of activities. The basic constructs of such diagrams are nodes and arrow connectors denoting activities and their precedence relationships respectively.

The fundamental difference between the two methodologies concerns the variability of activity durations. More specifically, activity durations are considered to be deterministic in the CPM whilst PERT deals with probabilistic activity lead times. The underlying assumptions made with regard to the stochastic nature of activity durations influence the scope of these two methodologies. On the one hand, CPM aims to establish an overall project duration, which is in turn determined by the duration of the critical path. PERT on the other hand, seeks to determine the probability that the project can be completed within a given timescale.

Samaranayake and Toncich (2007) examine the limitations of MRP2 software packages in project-based manufacturing applications. They explain that ERP systems address these limitations by interfacing their databases with project management constituents including CPM which support activity control and resource loading.

Azaron et al. (2011) combine Markov chains and PERT analysis to develop a model for determining minimum cost due dates. Networks of queues are formed to represent activities competing for shared resources and processing times are assumed to be stochastic. The model is employed to determine the project completion time in new product development and solve the due date assignment problem in mixed model assembly lines.

Hasgül et al. (2009) develop an agent-based architecture for scheduling tasks assigned to mobile industrial robots. Robots require substantial investment and are therefore considered to be scarce resources. Associated scheduling problems are considered to be resource-constrained. The proposed scheduling framework performs the CPM and resource levelling to determine task allocations. It is further capable of rescheduling when deviations from the plans occur.

The critical chain methodology introduced in the 1990's is the most contemporary alternative to traditional project scheduling techniques such as CPM and PERT (Goldratt, 1997). The critical chain is determined by modifying the critical path so that resource constraints are not violated (Blackstone et al., 2009). Another distinguishing feature of the critical chain methodology relates to the use of time buffers. These additional time allowances are strategically placed in the project network to protect the critical chain itself (completion and feeding/convergence buffers) or resources (resource buffers) used by it against delays.

Huang and Yang (2009) employ simulation to test the performance of CPM and Critical Chain Project Management (CCPM) in reducing project lead time. They apply both methodologies in a manufacturing project entailing the installation of a slab sizing press which is used as the test-bed in the simulation experiment. Their findings confirm the superior performance of critical chain theory in compressing project completion. Robinson and Richards (2010) review industrial applications of CCPM in the aerospace industry, citing successful implementations of CCPM software packages by Boeing.

### **3.3.3 Scheduling in flow systems**

Flow systems are highly automated and therefore their installation and configuration require strategic planning and significant capital investment (Topaloglu et al., 2012). Dolgui (2006) suggests that designing assembly lines is performed in tandem with scheduling. Becker and Scholl (2006) share this view and explain that every time that either a new assembly line needs to be set up or an existing line reconfigured to accommodate a modified production plan, the work produced on the line needs to be re-scheduled. Design decisions involve the selection of equipment, formation of workstations, determination of the line's production rate etc. and set the capacity of the line (Boysen et al., 2007).

On the other hand, scheduling decisions concern the allocation of operations (tasks) to workstations so that precedence (sequencing) constraints are not violated and the resulting workload is fairly uniform. This scheduling problem is referred to in the extant

literature as the Assembly Line Balancing Problem (ALBP) (Sholl and Becker, 2006). Once both the required number of workstations and workload assignments have been ascertained, workstations are interlinked by means of a material handling/transportation system e.g. a conveyor belt to form the assembly line. The design of the line may result in various layouts for instance serial, parallel, U-shaped etc.

The line balancing problem is in fact extended to include the determination of the sequences and batch sizes of different models assembled on the same line (Hop, 2006). Balanced flow systems can bring about certain benefits including effective utilisation of resources (both operators and machines) and minimisation of workstation idle time which is one of the main requirements of lean production (Askin and Chen, 2006).

Workstations producing at slower rates than others can cause idle time in the latter and further create build-ups of WIP. Cohen et al. (2006) identify that starvation of faster workstations and blockages caused by queuing WIP in front of slower workstations are the direct effects of idle time in unbalanced assembly lines. As stressed by Levitin et al. (2006), it is generally accepted that being a multi-criteria scheduling problem means that line balancing often aims to simultaneously optimise a set of performance objectives e.g. minimisation of idle time, set up time, maximisation of output rate etc.

A preliminary step in the formulation of the line balancing problem concerns decomposing the work that needs to be performed along the assembly line into a set of elemental tasks (Scholl and Becker, 2006). At this stage, the aim is to break down the work into manageable portions. However, as the durations of the latter may be quite short and impractical for one resource to perform, a bottom-up approach is also required to combine elemental tasks so groups with approximately equal time requirements are formed and assigned to workstations.

A determining factor in the context of line balancing is cycle time. Peeters and Degraeve (2006) explain that cycle time is the interval between the production of two consecutive batches assembled on the same line. This effectively suggests that the cycle time determines the output rate of the line. In the case of a single-model assembly line, Stevenson (2006) differentiates between the minimum and maximum cycle times which are established by the longest task time and the sum of all the task times respectively. According to Sotskov (2006) the cycle time is the ratio of the available operating time and desired output (as specified by customer demand) for the same production period.

Boysen et al. (2007) recognise that further to the workstations that form the assembly line and the multi-objective function that line balancing problems seek to optimise, formulations of the latter comprise a third essential component. This relates to precedence graphs (diagrams) used to visually illustrate the order in which tasks can be performed. Precedence graphs use nodes to indicate tasks, whilst the weights of the nodes (superscripts) represent task times. Arrows are also used to connect nodes to their direct successors, i.e. tasks that can be performed immediately after.

Lambert (2006) asserts that precedence relationships among tasks are defined by a range of constraints. Soft constraints concern the availability of resources. Hard constraints are imposed by the product structure and are therefore impossible to relax. He further stresses that precedence diagrams are quite rigid in the sense that they cannot depict alternative sequences related to product (model) variations. In order to overcome these shortcomings, Topaloglu et al. (2012) propose a rule-based model which relies on if-then rules to map out alternative precedence relationships. Their model is developed by constraint and integer programming.

Pastor and Ferrer (2009) declare that the most widely researched ALBP is the Simple Assembly Line Balancing Problem (SALBP). Given that, investigations of SALBP focus on assembly lines dedicated to the production of a single product (model), it can be argued that these constitute the simplest form of the ALBP. Taking into account the objective(s) SALBP problems aim to optimise and the prevailing constraints, Eswaramoorthi et al. (2012) organise these problems into the following types:

- SALBP-1. Objective: Minimisation of number of workstations.
- SALBP-2. Objective: Minimisation of cycle time.
- SALBP-3. Objective: Maximisation of workload smoothness.
- SALBP-4. Objective: Maximisation of work relevance.
- SALBP-5. Multiple objectives.
- SALBP-E. Objective: Maximisation of line efficiency.
- SALBP-F. Objective independent, seeking to produce a feasible line balance.

Indicatively, as originally defined by Baybars (1986) in SALBP-1, the cycle time is given and the objective is to minimise the number of workstations, whilst the reverse problem i.e. minimising the cycle time for a given number of workstations is referred to as SALBP-2. SALBP-F merely seeks a feasible solution given the cycle time and number of workstations.

Slack et al. (2010) recognise that solution methodologies cannot guarantee optimality as it is practically impossible to achieve a uniform (in terms of time requirements)

distribution of tasks to workstations. Considering the SALBP, they review simple heuristic rules to solve the line balancing problem. The first proposed technique relies on the selection of the task with the largest work content that can fill in the remaining time at a given workstation. According to the second technique, the selection is based on the task with the greatest positional weight or number of followers, that is, succeeding tasks.

Two measures commonly used to assess how well a line is balanced are the percentage of idle time (or balance delay) and the line efficiency (Stevenson, 2006). Essential for the computation of these measures is the notion of idle time. The latter is determined for each workstation by subtracting the sum of all the allocated task times from the cycle time. The following formulas apply:

$$\text{Percentage Idle Time} = \frac{\text{Idle time per cycle} \times 100}{\text{Number of workstations} \times \text{Cycle Time}} \quad (3.3)$$

$$\text{Line Efficiency} = 100 - \text{Percentage Idle Time} \quad (3.4)$$

Despite their inherent limitations and distance from real-life applications, SALBP problems continue to spark growing research interest. Blum (2008) develops a hybrid methodology combining metaheuristic ant colony optimisation and beam search to solve the SALBP-1. They test their approach on a set of benchmark problems. Their findings point out a superior and similar performance of the proposed model compared to other metaheuristics and exact methods respectively.

Pastor and Ferrer (2009) propose a mathematical model to solve SALBP-1 and SALBP-2. Their model is founded on existing theory concerning the feasible workstation interval determined by the earliest and latest workstations to which tasks can be assigned. It further builds in additional constraints to the mathematical formulation of the line balancing problem by linking this interval with the upper bound of the number of workstations and upper bound of the cycle time for the case of SALBP-1 and SALBP-2 respectively. Toksari et al. (2010) study a special case of the SALBP-1 which considers the effect of learning (workstations performing tasks faster after certain repetitions) and linear deterioration (task times are increasing functions of their start times).

A SALBP which seeks to simultaneously minimise the smoothness index (this measures the uniform distribution of workload into work centres) and design cost of a line with parallel machines and stochastic task times is considered by Cakir et al. (2011) who test a modified simulated annealing methodology on a set of test problems.

Hamta et al. (2012) examine the case of a single-model assembly line where task times are affected by learning and set-up times are sequence-dependent. They develop a hybrid metaheuristic approach which integrates particle swarm optimisation and variable neighbourhood search and is used to minimise cycle time, equipment cost and smoothness index in the single-model assembly line balancing problem. Their algorithm is compared to a multi-objective genetic algorithm proposed in the literature using a range of test problems. Experimentation findings confirm its superior performance in terms of solution quality and computational time.

Although researchers find the combinatorial nature and computational challenges of SALBP intriguing, Simaria et al. (2010) admit that the assembly lines more frequently encountered in industrial settings fall into the following categories:

- Mixed-model lines. In contrast to single-model lines (considered in the context of SALBP), mixed-model assembly lines are able to accommodate the assembly of a number of products, albeit these need to have an element of homogeneity, i.e. effectively be model variations of the same product. A key consideration concerns the determination of the production sequences of different models.
- Lines with parallel workstations. The main advantage here is that the output rate of the line can be increased by increasing the capacity of workstations. In addition, it is expected that line stoppages are prevented, as workstation failures can be compensated by the remainder parallel workstations.
- Two-sided lines. These imply that assembly operations can be performed on both sides of the assembly line. This gives greater flexibility in the assembly of large products as it is the case in automobile manufacturing.
- Flexible U-lines. The most characteristic design feature of these lines is that their entry and exit ends are in the same position. Due to their shape these lines can be manned by fewer human operators, thus allowing the production volume and variety to be flexibly adjusted to accommodate changes in demand.

U-lines were introduced in the context of JIT production with the aim to support multi-model production in a cost-effective manner (Monden, 1983a). They have been treated as a special type of cellular manufacturing systems. The line balancing and sequencing problem of mixed-model U-lines were further investigated by Monden (1998) and received continuing research attention since.

Kara and Tekin (2009) claim they have developed the first mathematical formulation of the line balancing problem in mixed-model U-lines. Their model aims to minimise the

number of workstations under various constraints. They further produce a new heuristic solution procedure which is validated on a set of large-scale test problems. A significant limitation of their approach is that it does not address the model sequencing problem.

Li et al. (2012) investigate mixed-model two-sided U-lines used for the cyclical assembly of minimum product sets. They develop a branch-and-bound algorithm and simple heuristic which seek to optimise model sequences whilst minimising work overload. They find that in approximations of real-life applications, it is mainly the simple heuristic which produces near-optimal solutions. They further argue that the complexity of simultaneously addressing the line balancing and sequencing problems stems from the fact that the aforementioned planning issues have different planning horizons.

Nevertheless, the joint line balancing/model sequencing problem is undoubtedly an emerging theme in recent literature. The line balancing and sequencing problem in mixed-model U-lines with stochastic task times is investigated using a genetic algorithm developed by Özcan et al. (2011). Lian et al. (2012) propose a modified colonial competitive algorithm to simultaneously balance and sequence mixed-model U-lines. Their computational approach is tested using a range of test-bed problems. Comparisons with other metaheuristic approaches proposed in the literature reveal the superior performance of the modified colonial competitive algorithm.

Hamzadayi and Yildiz (2012) develop a simulated annealing based fitness evaluation approach which they integrate with a priority based genetic algorithm to jointly address the line balancing and model sequencing problems in mixed-model U-lines where zoning constraints are imposed to restrict the number of parallel workstations.

Due to the proliferation of the investigations of the mixed-model assembly line balancing and sequencing problems in the academic literature, a number of comprehensive reviews have also been presented. Boysen et al. (2007) develop a classification schema for the research carried out in the field of assembly line balancing. They present this following the tuple notation  $\alpha|\beta|\gamma$  which has its origins in machine scheduling. In this notation,  $\alpha$  denotes precedence graph characteristics,  $\beta$  station and line characteristics and finally  $\gamma$  optimisation objectives. Each classification criterion is represented by a number of attributes.

A key attribute of the first classification criterion, namely  $\alpha$ , concerns the homogeneity/multiplicity of the products assembled on the line. According to this, further to the single-model and mixed-model assembly lines (discussed above), the

proposed classification includes multi-model assembly lines. As the difference between mixed-model and multi-model assembly is not self-evident, it should be clarified that the former produce model variations of similar products whereas the latter a range of diverse products.

According to classification criterion  $\beta$ , assembly lines are differentiated into paced and unpaced depending on whether a cycle time limits the available time at workstations. Unpaced lines can be either synchronous (coordinated movement of materials to/from workstations) or asynchronous (completed workpieces are placed in buffers). Line parallelisation and shape are other important attributes. The final classification criterion  $\gamma$ , identifies objectives related to the minimisation of the number of workstations, cycle time, line efficiency as well as the maximisation of profit, smoothing of station times and optimisation of scores covering several other line efficiency metrics.

Boysen et al. (2008) extend the above taxonomy by supplementing it with three additional classification categories. The first distinguishes between line balancing performed when the line is first installed or reconfigured (to accommodate production programme changes or replacement of obsolete equipment) and therefore needs to be re-balanced. It is accepted that the line re-balancing problem has attracted little attention by researchers despite its increasing relevance in industrial applications. Furthermore, automated production lines are differentiated from manual lines. The former typically rely on industrial robots so tooling selection is an important consideration. In contrast, manual lines are partially or fully controlled by human operators so task time variability due to both physical and psychological conditions is inevitable. The manufacturing setting, e.g. automobile manufacturing, consumer electronics industries etc. in which the line balancing problems are addressed is proposed as the final criterion for their classification.

Focusing explicitly on mixed-model sequencing in assembly lines, Boysen et al. (2009) classify the different approaches proposed in the relevant literature into classic mixed-model sequencing, car sequencing and level scheduling. The key differentiating factor in these approaches concerns the objective(s) set in each of these sequencing problems. It is accepted that these generally comprise the minimisation of work overload and achievement of JIT-related objectives.

The proposed classification assumes that whilst work overload is of the essence in mixed-model sequencing, car sequencing seeks to control work overload through the introduction of rules which aim to minimise the occurrences of work-intensive model options. Conversely, sequencing decisions made in level scheduling are driven by JIT



objectives related to supply chain integration, frequent deliveries in small quantities and minimisation of safety stocks. As a result, the central aim in level scheduling is producing sequences for mixed-model assembly which smooth out material requirements over the production period.

Emde et al. (2010) admit that line balancing and model sequencing are two planning tasks which involve long/medium-term and short-term decisions respectively. They argue however, that these problems are in fact interdependent as both unbalanced workload resulting from poor line balancing and incorrect model sequencing can lead to work overloads. Their work reviews investigations of the line balancing/sequencing problem and groups these according to the approach followed into successive planning and simultaneous planning. The distinction is clearly drawn by the fact that the former ignore any interdependencies and rather deal with each of these two problems in isolation, whereas the latter seeks to simultaneously derive an optimal line balance and model sequence. They propose an intermediate approach which relies on the anticipation of line imbalances that may result from certain model sequences. The basis for the development of such an approach comes from earlier works promoting horizontal balancing, that is, workload smoothing performed at each workstation and for all models.

The application of cross-trained utility workers as a means for compensating work overloads in sequencing mixed-model lines is investigated by Boysen et al. (2011). The underlying assumption in supplementing regular workforce with utility workers is that the latter will step in and support the former in increasing the processing speed of the workstation and ensuring work is completed on time. Contrasting such a work protocol coined side-by-side with what they describe as the skip policy, in which overloaded cycles become the sole responsibility of utility workers, they argue it is the latter which is most widely implemented in practice. They develop a binary linear program to address the resulting sequencing problem.

According to Emde et al. (2010) planning mixed-model assembly lines involves two tasks: long-term line balancing and short-term sequencing. In terms of how these tasks can be performed they identify three different approaches namely, successive, simultaneous and anticipation. The successive approach segregates these tasks whilst simultaneous planning resolves them jointly. The third approach, anticipation, postulates a compromise between the aforementioned two and promotes line balancing solutions which seek to eliminate short-term sequencing overloads.

It is accepted that the methodologies employed to solve line balancing and model sequencing problems, are broadly classified as heuristics, metaheuristics and exact methods (Lambert, 2006). Heuristics rely on simple (intuitive) algorithms and provide good but not necessarily near-optimal solutions. They are mainly applicable to specific product configurations. This shortcoming is overcome with the use of metaheuristics which additionally can generate near-optimal solutions. Simulated annealing, tabu search and genetic algorithms are typical examples. Finally, exact methods rely on mathematical programming e.g. linear, binary linear, integer etc. and normally guarantee near-optimality.

Ribas et al. (2010) review solution methodologies proposed for the scheduling of hybrid flow shops with parallel machines. Their proposed classification comprises heuristics, exact methods and simulation/decision support systems. They differentiate between constructive heuristics which aim to develop an initial solution and improvement heuristics used to improve an existing solution. Interestingly, they consider metaheuristics as a special subclass of improvement heuristics. Their findings suggest that branch-and-bound and mixed integer programming are the most commonly employed exact methods. With regard to heuristics they observe a trend in available studies to combine problem decomposition and parallelisation of heuristics in order to obtain flow shop scheduling solutions. They finally note that simulation/decision support systems are primarily used to model and analyse the performance of real-life systems, however ignoring in most applications real-time changes.

Soft computing approaches employed to solve the NP-hard line balancing and sequence planning problems are analysed by Rashid et al. (2012). Their research reveals that the most commonly adopted methodologies are genetic algorithms, ant colony optimisation and particle swarm optimisation. They maintain that although the aforementioned computational techniques have been found to produce near optimal solutions, there appears to be a great distance between the problems considered in such applications and those encountered in real-life industrial environments. It is also stressed that the minimisation of the computational costs remains yet another challenge.

### **3.3.4 Scheduling in job-shops**

MRP systems are primarily used in production planning and do not perform any scheduling functions. They generate order releases which serve as the main inputs to the short-term scheduling performed at the shop-floor level (Selçuk et al., 2009). Once

an MRP job list is created, decisions need to be made to determine the workstations orders can be routed to, the assignment of jobs to specific machines (in case of parallel machines) and the prioritisation of jobs awaiting processing in front of the same machine. These are typical decisions made in the framework of job-shop scheduling.

Job-shop scheduling is a complex combinatorial problem. Due to its complexity, attempts to reach solutions have taken two discrete stances. The difficulty in obtaining high quality solutions by employing optimisation approaches gave rise to approximation techniques aiming at determining near optimum solutions. According to Sha and Hsu (2006), the demonstration of the Job-Shop Scheduling Problem (JSSP) as NP-hard is credited to Garey et al. (1976). Despite years of intensive research and advances in solution search methodologies and computational power, there is substantial evidence to suggest JSSP remains an intractable NP-hard problem (Arasteh et al., 2011; Meeran and Morshed, 2012; Shen and Buscher, 2012).

The complexity of the JSSP stems, in part, from the volume and variety of production typically accommodated by job-shops. More specifically, although the volume of orders is quite low, the range of jobs can be quite diverse. In addition, most of these have distinctive routings and processing requirements. Gaither and Frazier (2002, p.625) analyse the variety/volume characteristics of production and shop-floor configuration in jobs-shops in relation to their implications on scheduling. They highlight that:

- In order to be able to cope with product diversity, substantial preproduction planning is required so that all job design and processing data including routings and processing times can be determined.
- Job-shops employ versatile machines and multi-skilled operators able to process a variety of jobs and operations. However, this flexibility comes at the expense of accumulating WIP inventory in between processing steps.
- Decoupled workstations and product diversity increase the number of jumbled flows through the system. This leads to unbalanced machine workloads with some machines being heavily utilised whilst others have ample idle time. The complexity and magnitude of job routings creates a pressing need for tight control of material flow.

Baykasoğlu and Durmuşoğlu (2012) reinforce this final point and argue that machines, routings and queues are intricacies which define job-shops. They stress that the trade-offs between manufacturing flexibility and system complexity need to be carefully balanced by implementing elaborate queue management and workflow control systems.

The multiplicity of scheduling objectives is another complicating factor in JSSP. Globalisation and increasing demands for customisation, leanness, agility and responsiveness strengthened the prevalence of job-shops. However, Chong et al. (2006) admit that modern manufacturing challenges created contradictory objectives. Inventory management is a case in point. Pressures for cost minimisation require inventory levels to be kept to a minimum, but on the other hand inventories facilitate shorter delivery times and thus drive competitive advantage.

Scheduling objectives are integral parts of the JSSP formulation. They have a dual function. First of all, they are built into the objective functions that the JSSP problem seeks to optimise (Wang and Shen, 2007). Moreover, they are the measures used to assess the performance of the schedule or in other words, the quality of the generated solution. The mathematical formulation of the scheduling objectives relies on key job input/output data.

The following notation provided by Sipper and Bulfin (1997) is used to represent job input data:

- $n$  = number of jobs processed
- $m$  = number of machines in the system
- $p_{ij}$  = time job  $i$  spends on machine  $j$  (processing time for that step or operation)
- $r_i$  = release date for job  $i$  (time job arrives at the system – earliest time its processing can start)
- $d_i$  = due date of job  $i$  (delivery/shipping deadline promised to the customer)
- $w_i$  = weight of job  $i$  (importance of job  $i$  relative to other jobs in the system – a priority factor)

T'Kindt and Billaut (2006, p. 13&328) classify scheduling optimality criteria into *minimax* and *minisum* with the former aiming to minimise a maximum function and the latter seeking to minimise a sum of functions. They provide the listing of commonly cited scheduling objectives presented in Table 3.2.

Krajewski and Ritzman (2005) identify that WIP and machine utilisation are also among the most frequently encountered job-shop scheduling objectives. They explain that:

- WIP (expressed in product units) is a broad measure which considers all jobs in the shop. In addition to jobs being processed, those queuing in front of machines or being in transit between workstations are treated as WIP.
- Utilisation is the ratio of the machine's productive work time over the total work time they are available.

It is therefore clear from the above that job-shop scheduling objectives are related to completion time (speed), due-date (customer waiting time), WIP and utilisation.

**Table 3.2 Performance objectives in operations scheduling**

	<b>Measure based on:</b>	<b>Mathematical expression</b>
<b>Minimax</b>	Makespan (completion times)	$C_{max} = \max(C_i)$
	Flow time	$F_{max} = \max(F_i)$
	Lateness	$L_{max} = \max(L_i)$
	Tardiness	$T_{max} = \max(T_i)$
	Earliness	$E_{max} = \max(E_i)$
	Machine idleness	$I_{max} = \max(I_k)$
	Completion times	$\sum C_i$ Total completion time
		$\frac{1}{n} \sum C_i$ Average completion time
		$\frac{1}{n} \sum w_i C_i$ Average weighted completion time
<b>Minisum</b>	Flow times	$\sum F_i$ Total completion time
		$\frac{1}{n} \sum F_i$ Average completion time
		Average weighted flow time is the same as the Average weighted completion time
	Number of late jobs	$\sum U_i$ Total number of late jobs
	Tardiness	$\sum T_i$ Total completion time
		$\frac{1}{n} \sum T_i$ Average completion time
		$\frac{1}{n} \sum w_i T_i$ Average weighted completion time
	Earliness	$\frac{1}{n} \sum E_i$ Average earliness
		$\frac{1}{n} \sum w_i E_i$ Average weighted earliness

Source: T'Kindt and Billaut (2006)

whereby:

$C_i$  = completion time of job i

$F_i = C_i - r_i$  flow time of job i

$L_i = C_i - d_i$  Lateness of job i

$T_i = \max(0, C_i - d_i)$  tardiness of job i

$E_i = \max(0, d_i - C_i)$  earliness of job i

$I_k$  = sum of idle times on machine k

$U_i = 0$  (if job i is not late) or 1 (if job i is late)

Pinedo (2009) suggests that in its most basic form, the JSSP comprises n jobs which may visit each of the m machines (nxm) only once. A variation of this problem is the job-shop with recirculation, where jobs are allowed to visit the same machine(s) more than once. Hasan et al. (2009) outline the main assumptions underlying the classic (basic) JSSP. Some of the least evident are summarised below:

- Job routings are predetermined. The processing times are deterministic, that is, not subject to variations.
- Preemptions are not allowed. Once a machine has started processing a job, its operation cannot be interrupted in favour of another job e.g. a rush order.
- Setup times and costs are ignored.
- Machines operate at peak efficiency. This practically means there is no downtime for planned or reactive maintenance.

An extension of the JSSP is the Flexible Job-Shop Scheduling problem (FJSSP). Its main differentiation from the JSSP is that workstations comprise multiple (parallel) machines. It is generally recognised that the FJSSP is a better approximation of “real” industrial applications. Xing et al. (2010) consider the additional challenges encountered in scheduling flexible job-shops. They develop a new optimisation approach for the FJSSP. Their solution methodology named Knowledge-Based Ant Colony Optimisation (KBACO) integrates knowledge and heuristic searching modelling approach. Results obtained from their experiments show that KBACO performs better compared to other heuristic and metaheuristic approaches. Their study focuses on the optimisation of a single objective, the makespan.

The FJSSP with the makespan minimisation criterion is also considered by Li et al. (2011). They use a tabu search-based model equipped with a critical block neighbourhood structure mechanism which is used to enhance the exploitation capability of their hybrid approach. This was found to perform well in benchmark problems considered in their simulation experiments. Zhang et al. (2011) propose a Genetic Algorithm (GA) for makespan optimisation in the FJSSP. Their algorithm is found to outperform other genetic algorithms in terms of computational speed and quality of the generated solution.

The multi-objective FJSSP with makespan, total weighted earliness and tardiness minimisation criteria is studied by Kachitvichyanukul and Sitthitham (2011). Their 2-stage approach applies parallel genetic algorithms to optimise each objective function in stage 1 and combines solution populations in stage 2. Moslehi and Mahna (2011) integrate particle swarm and local search algorithms to optimise makespan and machine workload objectives in FJSSP. The local search algorithm improves the solutions generated by particle swarm optimisation by rescheduling operations. Their directions for further research identify scope for enhancing the computational efficiency of their approach. Li et al. (2011) argue they develop the first hybrid approach combining an artificial bee colony algorithm with an external Pareto archive set. Results from their experiments point suggest that the proposed algorithm competes well with

other approaches reported in the relevant literature. They acknowledge however that future work is required to improve its convergence speed.

Lei (2009) reviews the multi-objective scheduling literature and concludes that in their majority, investigations of the JSSP and FJSSP tend to ignore the constraints e.g. set-up times, machine breakdowns, WIP limitations etc. which typically prevail in job-shops. It is suggested that interestingly, such simplifications of the job-shop scheduling problem appear to be made even when considering single-objective functions.

There appears to be some confusion in the relevant literature concerning the semantics of job-shop scheduling. According to Hill (2012), scheduling is an operations control activity which specifies the start and finish times of operations. However, scheduling is not performed in isolation; on the contrary, it is interlinked with loading and sequencing decisions.

Greasley (2008) subscribes to this view and explains that the loading activity determines the assignment of jobs to machines. Loading further involves balancing production volumes with the available capacity at each workstation. Sequencing prioritises orders once these have been assigned to specific machine(s). It is highlighted that loading and sequencing need to precede scheduling.

Russell and Taylor (2009) suggest that shop-floor control and job-shop scheduling are two terms used interchangeably. They further suggest that production control encompasses loading, sequencing and monitoring. Chary (2009) agrees and explains that the monitoring function aims at reviewing the schedule and identifying corrections in case of deviations.

It is obvious that views are divided. One school of thought views scheduling as a function integrated with loading and sequencing in the broader context of production control. This approach is contrasted to that of those who identify scheduling and control as identical functions. In line with the discussion in section 3.2, scheduling must be distinguished from shop-floor control. Scheduling is mainly concerned with the execution of short-term plans whilst monitoring and control primarily focus on checking planned versus actual progress. In doing so, they rely on established schedules and use these as benchmarks.

#### **3.3.4.1 Loading**

Loading decisions are capacity-related. Hill and Hill (2011) recognise that there are two approaches to machine loading. Infinite loading assigns jobs to machines without

consideration of their available capacity. The opposite approach, referred to as finite loading, provides a mechanism that restricts job allocations to machines so that capacity limitations are not violated. Crowson (2006) points out that the maximum time a machine or workstation is available relates to its theoretical capacity which in reality tends to diminish as a result of other unavoidable activities e.g. changing over from one product to another, cleaning operations, planned maintenance or unexpected machine breakdowns.

Schönsleben (2007) differentiates finite loading into operations-oriented and order-oriented. The aim of operations-oriented loading is to accelerate the completion of individual operations and therefore the production of the complete order. This situation is typically encountered in job-shop production. Several order-oriented approaches are identified. In contrast to operations-oriented finite loading, order-oriented seeks to maximise throughput and machine utilisation whilst ensuring low levels of WIP. The fundamental principle here is that orders are scheduled according to their priorities and capacity overloads are dealt with by loading production orders which have non-negotiable due-dates or deferring others when capacity cannot be raised over certain production periods.

Finite loading appears to be feasible in certain types of services and operations where it is possible or even necessary to limit the load e.g. passengers allowed on an aircraft (Slack and Lewis, 2011). In contrast to this, Slack et al. (2010) refer to the case of a machine shop in an engineering company to provide an example of a production environment where capacity constraints complicate finite loading to an extent which does not justify the time and computational power required to prevent overloads.

#### **3.3.4.2 Sequencing**

Loading results in the assignment of several jobs to each machine. The order in which queuing jobs will be processed must be specified; this prioritisation is called sequencing (Swamidass, 2000). Sequencing is carried out by utilising simple dispatching rules. These effectively convert MRP order releases i.e. the jobs arriving at the shop-floor into dispatching lists containing prioritised orders ready for processing at the nominated machines (Russell and Taylor, 2009).

Despite the simplicity of the majority of dispatching rules, Jayamohan and Rajendran (2000) accept that applying dispatching rules is a complex matter. They attribute this to the multiplicity of possible job/operation sequences. There are  $n$  factorial ( $n!$ ) ways in which  $n$  queuing jobs can be prioritised. In addition, the dynamic conditions prevailing



at a job-shop can affect the performance of the employed dispatching rule. Wild (2002, p. 367) suggests two different criteria which can be used to classify dispatching rules. The first criterion relates to the locality of job data. Using this criterion, dispatching rules are grouped into:

- i. Local rules. These prioritise jobs using data concerning the jobs queuing at a particular machine.
- ii. General rules. These utilise data related to all the jobs in the system, including those queuing in front of other machines or workstations.

The second classification criterion examines the impact of time on job data and subdivides dispatching rules into:

- i. Static rules. These assume that the priority indices of jobs (irrespective of which queues these temporarily reside in) do not change with the passage of time.
- ii. Dynamic rules. If these rules are selected, job priority is a function of time and therefore, indices need to be updated accordingly.

A further classification is proposed by Pinedo (2009). This is based on the number of objectives dispatching rules seek to achieve. Basic dispatching rules are intuitive heuristics employed to address simple objective functions. Such basic rules can be combined to form more elaborate rules, referred to as composite rules used to optimise more complex objective functions. Some of the most commonly cited dispatching rules and the reporting sources from the operations scheduling literature are presented in Table 3.3. The full definitions of these dispatching rules and the criteria used to assign priority to jobs are presented in Appendix A.

Some of the first investigations of job-shop sequencing provided comprehensive surveys of the wide spectrum of dispatching rules available (Blackstone et al. (1982); Haupt, 1989; Ramasesh, 1990). Dynamic jobs-shops appear to be on the focus of more recent studies. Rajendran and Holthaus (1999) evaluate the performance of 13 dispatching rules with respect to flow time and tardiness criteria. They consider both flow-shops and job-shops with dynamic arrivals and random job routings. Jayamohan and Rajendran (2000) develop seven new composite rules, namely FDD, PT+PW, PT+PW+FDD, PT+PW+ODD, OPFSLK/PT; FDD, OPSLK/PT; ODD and AVPRO. They use a simulated open job-shop to compare their performance against that of nine benchmark dispatching rules. They use a selection of performance criteria including mean flow time, tardiness and number of tardy jobs.

**Table 3.3 Commonly cited dispatching rules and reporting sources.**

Dispatching rule	Reporting source										
	1	2	3	5	6	7	8	9	10	11	12
COVERT				*		*					
CR	*	*	*				*	*	*	*	
EDD	*	*	*	*	*	*	*	*	*	*	
FASFS				*		*					
FCFS	*	*	*	*	*	*	*	*	*	*	*
FOR									*		
LCC	*									*	
LCFS							*			*	
LFJ					*						
LPT		*			*		*			*	*
LTWK				*		*					
LWKR				*		*			*		
MOPNR						*					
MS	*			*	*		*		*	*	*
MWKR						*					*
ODD				*							
RUSH							*	*			
S/RO			*	*		*		*			*
SIRO					*	*			*		*
SPT	*	*	*			*	*	*	*	*	*
SSD											*
SST					*						
WINQ				*	*	*					
WSPT				*	*						

(1) Gaither and Frazier (2002); (2) Heizer and Render (2004); (3) Krajewski et al. (1999); (4) Morton (1999); (5) Pinedo (2009); (6) Roy (2007); (7) Russell and Taylor (2009); (8) Stevenson (2006); (9) Vollmann et al. (1997); (10) Waller (2003); (11) Wild (2002)

Dominic et al. (2004) create an enlarged set of basic and composite rules which they test across several flow time and tardiness performance criteria in a dynamic job shop environment where job arrivals, routings and number of operations are generated randomly. They conclude that the MWKR\_FIFO and TWKR\_SPT rules performed well across most of the performance measures considered.

A new dispatching rule named Enhanced Critical Ratio (ECR) is proposed by Chiang and Fu (2007). The rule is a combination of the SPT, EDD and LRPT rules. Moreover, the rule's computational efficiency is enhanced by a job candidate reduction mechanism. They use simulation to study its performance against 18 benchmark rules

sourced from the relevant literature and conclude this is a promising rule for job-shops with due-date objectives.

El-Bouri and Shah (2006) argue that applying local dispatching rules as opposed to adopting a common rule across the shop-floor can lead to improved performance. They develop two neural networks trained to achieve makespan and flow time minimisation. These identify a suitable rule from the following set: SPT, LPT, LWKR, MWKR, PT+WINQ every time a machine becomes available. Their research confirms that combined rules identified by the two neural networks outperform single dispatching rules implemented globally. This is a significant point of departure from the perception that local rules are myopic as they fail to account for conditions generally prevailing at the shop-floor and can therefore limit scheduling performance (Sarin et al., 2011).

Ouelhadj and Petrovic (2009) argue that owing to the fact that no single rule can guarantee optimum performance across a range of criteria, investigations of existing and development of new rules proliferate in the extant literature. In this direction, Baykasoğlu and Özbakir (2010) analyse the relationship between dispatching rules and system performance. Their study considers the following rules SPT, EDD, MWRT, LWRT, PDR, ERD, MS and LNS which are applied to flexible job-shops with four different machine flexibility levels. They find that the effect of dispatching rule is stronger in lower levels of job-shop flexibility.

Pickardt and Branke (2011) focus on setup-oriented dispatching rules. Further to pure and composite rules, they identify a third category comprising family-based rules. The latter are further subdivided into exhaustive and truncated rules depending on whether they permit changeover to a new family prior to the exhaustion of the current family. Their findings suggest that composite rules result in good due date performance whilst family-based rules satisfy setup and flow time reduction objectives.

Vinod and Sridharan (2011) employ simulation to study the interaction between dispatching rules and due-date assignment methods including dynamic processing plus waiting time, total work content, dynamic total work content and random work content. The two dynamic due date assignment methods were found to improve performance in terms of flow time and tardiness of jobs. The results obtained from this study apply to specific job-shop conditions which ignore a dynamic job arrival pattern and disruptions caused by machine failures etc.

Russell and Taylor (2009) assert that due to the complex and dynamic nature of scheduling, simulation has been favoured over analytical methods in investigations aiming to produce general guidelines for situations in which dispatching rules are

mostly appropriate. In a similar vein, Sarin et al. (2011) review dispatching rules that can be applied in the general process and specific operations within wafer fabrication. They reaffirm the prevalence of dispatching rules over other order release policies that rely on computationally intensive mathematical programming models or expert systems and AI applications which are laborious to develop and customise.

### **3.3.4.3 Forward/backward scheduling**

There are two general approaches to scheduling. These are referred to as forward and backward scheduling (Heizer and Render, 2004). The differentiating factor between the two is the point in time scheduling of operations starts from.

Vonderembse and White (2004) explain that in forward scheduling jobs are scheduled to start as close to the present time as possible. This theoretically means that processing can commence immediately after a job has been released to the shop-floor. In practice this is not always the case as machines can be busy processing other jobs. Backward scheduling works in the reverse order. The defining point in determining a time line is the job's due date. The last operation of each job is scheduled first so that it can be completed by the desired due date. Accepting the due date as the job completion time, the start time for the final operation of the job can be derived by deducting its processing time. The start time of the last operation determines the completion time for the one preceding it and so forth.

Slack et al. (2010) argue that the selection between forward and backward scheduling depends on the type of application. They point out however, that both MRP and JIT utilise backward scheduling techniques. They consider the merits of these two approaches and the following points can be drawn from their analysis:

- Forward scheduling:
  - The approach leads to high resource utilisation. Operators and machines become engaged as soon as work arrives at the system.
  - Completing the work as soon as possible can reduce throughput time.
  - In addition, there is slack time to accommodate the production of urgent orders and cope with unexpected events.
- Backward scheduling:
  - Costs associated with raw materials and WIP can be minimised as these can be ordered and handled respectively right before they are needed.

- The system can easily adapt to early modifications/cancellations of customer orders.

It is clear from the above analysis that both approaches expose production systems to risk. In forward scheduling production systems are vulnerable to order cancellations whilst backward scheduling allows limited time buffer to cope with sources of variability e.g. operator absenteeism, machine breakdowns, late delivery of materials.

Concluding the discussion of the scheduling functions presented in the above subsections, scheduling can be viewed as a process which converts various inputs into a set of outputs. These outputs provide tangible production schedules, machine loadings and job prioritisations but also define the performance and efficiency of the overall process. Scheduling uses mechanisms to drive the process whilst taking into account a range of relevant constraints. Scheduling does not take place in isolation but rather within a very dynamic context which transcends the boundaries of the internal production setting and reaches the external supply chain. This context creates sources of variability which can compromise the quality of scheduling decisions. The conceptual scheduling framework illustrated in Figure 3.5 provides an overview of the scheduling process.

This section builds on the premise of section 3.1 which argued that the scope and functions of scheduling vary depending on the type of manufacturing system. Starting from scheduling performed in the context of process industries and project manufacturing, the section is gradually narrowing its focus on flow-shops and job-shops. The discussion reveals that despite several fundamental differences, scheduling performed in all four types of manufacturing systems adheres to some common principles. These concern both the scheduling objectives and tools used. Precedence graphs and makespan are cases in point, as both are found to be relevant in all four types of manufacturing systems.

The most contentious issue in this thesis concerns the extension of scheduling and pull control designed for flow lines into job-shops. It was therefore prudent to examine the fundamental differences in scheduling applied to these two diametrically opposed systems. The review pointed out that line balancing is the primary focus of scheduling in flow-shops. The discussion extended into dominant concepts of line balancing including, uniform loadings, stable schedules and mixed-model production which are acknowledged in section 3.4 as key prerequisites for the implementation of pull control.

The section provides an insight into job-shop scheduling. Backward scheduling is distinguished from forward scheduling and found to imitate the reverse scheduling logic

of pull control. Emphasis is also placed on dispatching which is identified in section 3.4 as one of the key issues affecting the operation of pull control mechanisms.

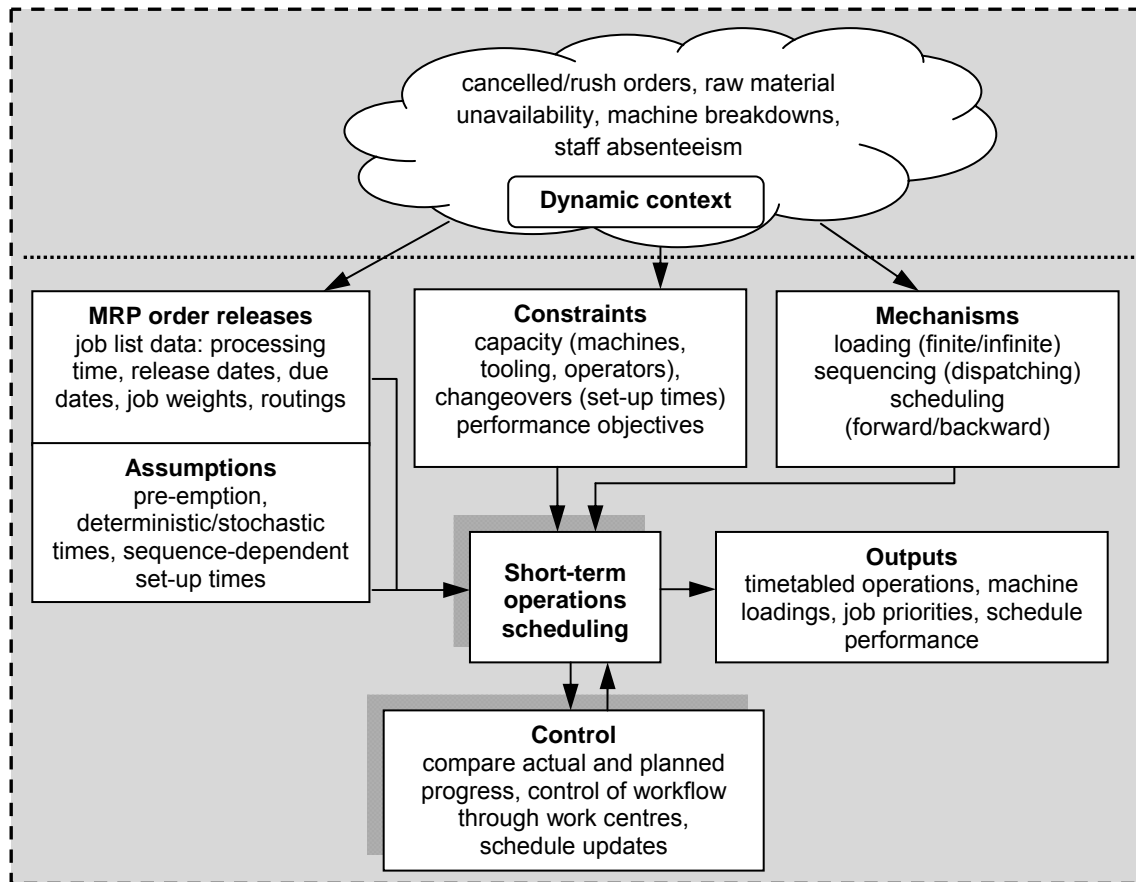


Figure 3.5 Conceptual job-shop scheduling framework

### 3.4 Control

Shop-floor control is concerned with the monitoring of orders and acquisition of real-time information on the progress and status of these orders. In other words, shop-floor control deals with the management of WIP. Production control is of the utmost importance in job-shops where a diverse range of products, each with its own priority, need to be tracked effectively.

Key to tracking inventory is the monitoring of jobs flowing through work centres, an activity referred to as Input/Output (I/O) control (Gaither and Frazier, 2002). This allows operations managers to assess if work flowing through a work centre is according to the preset production plan. Inventory building up in front of a given work centre is an indication that jobs coming into this work centre exceed the plan and the work centre's available capacity. Such a situation results in downstream work centres starving for

jobs. Idle work centres in turn suggest that work flowing through them is less than what was originally planned.

Heizer and Render (2004, p. 564) provide an example of an I/O control report. Actual input and output data representing jobs flowing into and out of each work centre are converted into their respective capacity requirements in labour-hours. These are then compared against benchmarked job planned data and the work centre's available capacity. Gantt charts are commonly used to map out the start and finish times of operations. These are horizontal bar charts enumerating operations in the vertical y-axis whilst the horizontal x-axis provides a time line (Naylor, 2002). A Gantt chart is as much a planning as it is a control tool. Initial production schedules are baselined and reviewed in the context of I/O control to identify deviations and the necessary corrective measures to bring production back on track.

### **3.4.1 Push production control**

MRP systems determine order release dates and times by propagating backwards planned procurement and fabrication lead times (Jodlbauer and Huber, 2008). The underlying assumption is that lead times remain constant. In reality however, lead times can be inflated by the WIP flowing through the system. Work is also loaded assuming infinite capacity and it is only through the MRP feedback loop that exceptions are reviewed and corrected by schedule regeneration.

Luh et al. (2000) accept that in order to moderate the effects of these oversimplifications, MRP systems utilise longer lead times to cope with shop-floor uncertainties and inaccurate capacity estimates during the planning process. They further acknowledge that longer lead times mean higher levels of WIP within the system. This is a long-recognised truth with Berger (1987; cited in Plenert, 1999) pointing out that through their everyday use, MRP were turned into labour efficiency oriented systems that use lead times and inventory build up as means for maximising labour efficiencies.

Waller (2003, p. 460) suggests that once planned order releases are generated, MRP plans driven by firm and projected orders, are set into motion and products get pushed through the production pipeline. MRP is presented as a system promoting the push culture "*where products are manufactured, pushed through the supply chain where it is then up to the sales personnel to find clients*". The less evident side of such a plan-push system relates to WIP accumulation and the risk of overloading the entire factory with more work than it can possibly handle.

### 3.4.2 Pull production control

Contrary to rigid MRP supply-push systems, JIT systems are recognised as flexible demand-pull systems where actual demand pulls finished products off the fabrication line by triggering production in the last stage of the process (Barnes, 2008). When the last stage finishes work, it transmits a message to the preceding centre requesting another batch to work on. Effectively, each stage generates demand for the stage upstream and every work centre produces only what is required by the process downstream in the next time period.

Hill and Hill (2011) explain that in MRP systems, demand information dictated by MRP order releases travels through the system in the same direction in which materials flow from the raw materials inventory (upstream) to the finished goods storage (downstream). In contrast to this, in pull production, orders determined using actual demand information follow the reverse path travelling upstream through the system. They further highlight that pull production control encompasses the JIT principle according to which, each stage in the process is fed by the preceding stage at the right time. JIT practically means that operations should be completed neither late nor early, as in the first case customer service will be compromised whilst in the latter WIP will pile up and clutter the shop-floor.

Curry and Feldman (2010) point out that JIT pull production control addresses some of the shortcomings of push systems by attempting to control order releases using information about the existing shop-floor conditions. Groover (2011, p. 959) views JIT production as the Japanese alternative to the mass production mentality promoted for several decades by US manufacturers. He coins the latter as a "*just-in-case*" philosophy which uses large WIP inventories to cope with production problems and uncertainties.

The above discussion highlights that material flow and inventory control are in fact intertwined. JIT is as much a production control as it is an inventory management system. JIT promotes an entirely different mindset as far as inventories are concerned. Instead of using inventories to mask sources of variations, JIT regards inventories a waste of resources. Paton et al. (2011) insist that instead of holding inventory to cope with uncertain demand, lean systems fix inventory so that it comprises WIP directly driven by customer demand.

Plenert (1999) provides an informative précis of the key MRP-push and JIT-pull differences across six criteria. MRP is clearly associated with systems able to accommodate the production of diverse product ranges as opposed to JIT which is



mainly encountered in the repetitive production of narrow product varieties. In that respect, it is also suggested that shop-floor layouts appear to be quite flexible in MRP but very restricted in JIT systems. MRP systems can cope with the scheduling needs of high variety production, whilst JIT pull systems offer no scheduling functionality. However, in terms of order tracking, MRP systems appear to be quite demanding compared to pull systems where control of orders is built into the core JIT logic. Data accuracy and computational power appear to have almost no significance in JIT, whereas MRP systems heavily depend on these factors.

#### **3.4.2.1 Kanban system operating principles**

In the process of masterminding JIT production and pull control, Ohno introduced stores between work centres which were used to hold inventory (Liker, 2004). These stores were deliberately small so that their restricted capacity could control the level of inventories. This storage capacity effectively specified a predetermined re-order level. Ohno further needed to devise a mechanism which could be used to signal that a production stage had used its part and needed more. Such a signalling mechanism was provided by a system of cards and WIP containers, referred to as the kanban system. A kanban card was attached to an empty container and sent back to the preceding process each time a station needed more materials for processing. Sarin et al. (2006) admit that the Kanban system is the earliest and most widely adopted pull control mechanism.

Considering that JIT pull control transcends the boundaries of the shop-floor and extends across the entire supply chain, several types of kanban cards were introduced. Huang and Kusiak (1996) classify these into five categories depending on the message transmitted and travel distance covered by the cards. The cards mainly used for shop-floor control are primary kanbans. These cards are circulated within the limits of the shop-floor. Primary kanbans are further subdivided into production and withdrawal kanbans depending on the type of authorisation they are used to provide. Production kanbans are used to initiate production in an upstream station whenever the downstream stage consumed an inventory portion. Withdrawal kanbans on the other hand provide the necessary authorisation for the transportation of parts between stages.

In its early applications, kanban was implemented as either a single-card or dual-card system. A detailed description of the operation of the single-kanban system is provided by Yang (2000). The only cards in circulation within a single-kanban system are

withdrawal kanbans. The system is set up so that each station has an incoming and outgoing card posts used for the temporary storage of kanbans as well as an input and output WIP storage points. The only exception is the last station which only has an outgoing card post. A material handler periodically removes the cards that have accumulated in the outgoing post of a station and posts them onto the incoming post of the preceding station, ensuring the original order of the cards is maintained. The material handler performs a second round during which all full containers with attached withdrawal cards are collected from the output storage of a preceding station and transferred to the input station of its succeeding station. At each station, production of a certain part can only commence if a card for that part is available at the incoming card post and can be matched with a full container from the station's input storage. The station's operator will detach the card from the full container which is about to be processed and post it into the station's outgoing card post. After its processing, the full container will be matched with the card (from the incoming card post) which was used to trigger its production and moved to the output storage of the station.

The operating principles of the dual-kanban system were first discussed in Schonberger (1982) and Monden (1983a). According to the analysis presented by Horng and Cochran (2001) a production system controlled by the dual-kanban mechanism is configured so that each station has two parts buffers, two production kanban posts and two withdrawal kanban posts. Each of these is placed in the input and output points upstream and downstream the station respectively. Production kanbans circulate between the input and output production kanban posts of the same station. On the contrary, withdrawal kanbans travel between stations. A production kanban at the input production kanban post of a station awaits a full container of parts to arrive at its input storage buffer. Assuming the station is idle, the machining process commences. At the end of processing, the full container and production kanban which has followed the batch through the machining process are placed in the station's output buffer and output production kanban post respectively. Production kanbans are moved from the output production kanban post of a station into its input production kanban post at each reorder point. Withdrawal kanbans travel through the system in the following manner. A withdrawal kanban in the output withdrawal kanban of a station awaits a full container to be processed before they are both moved to the succeeding station. There, the withdrawal kanban is stored in the input withdrawal kanban post until the full container is put through processing again. When processing commences, the kanban is moved into the output withdrawal kanban post of the same station and the same process repeats itself.

Ramanan and Rajendran (2003) suggest withdrawal kanbans are used for part requisition in material-handling operations performed by dedicated transportation workers. They argue that given the finite input and output storage buffers of each station and periodic frequency by which material handling is carried out, station blocking is unavoidable. In line with the requirements of kanban operation, a station is forced to remain idle when its output storage buffer is replete with full containers. Processing can resume once material handlers have removed at least one full container. However, such an operation presumes that there is space available in the input storage buffer of the succeeding station.

Walters (2003) accepts that several variations of the kanban system exist, however all these comply with the following basic operating principles:

- Kanbans are used to communicate the need for upstream stages to supply materials to downstream stages
- Materials are transported in standard containers which have predetermined capacity
- Only full containers can be produced and transported
- Containers can be transported only if they have kanbans attached
- Containers have a reasonably small size, generally this can be equivalent to 10% of the daily production
- By limiting the number of containers and kanbans, a more rigid control of material flow can be imposed.

One of the most important parameters in establishing a kanban system concerns the number of cards to be put into circulation. The first guidelines in this direction were provided by Sugimori et al. (1977) who developed a formula for setting the number of kanbans between two connected stations. The formula was further endorsed by Monden (1983a,b) and Shingo (1987) who analysed the trade-offs between the level of inventory within the system and number of cards. The proposed formula manifests a linear relationship between the maximum stock held within the system and number of kanbans. The goal of JIT is inventory minimisation and therefore the number of cards should be the minimum feasible number possible. The formula originally proposed by Sugimori et al. is still accepted as one of the key rules in running the kanban system (Stevenson, 2006; Paton et al., 2011). It has the following form:

$$\text{Number of kanbans (and containers)} = \frac{D \times (T_w + T_p) \times (1 + SS)}{C} \quad (3.5)$$

Where:

D = Planned usage rate (demand) for the part

$T_w$  = Part (fraction) of the cycle the kanban spends waiting and moving empty/full containers to the supplying/demanding station

$T_p$  = Part (fraction) of the cycle to produce a full container of parts

SS = Safety stock factor, determined by management as buffer against production instability. Normally set at 10%.

C = Capacity of the container, this can be determined on a production order size EOQ basis.

Shingo (1987, p 180) provides an example of kanban cards analysing the information they were used to convey. Despite being introduced as a card/token-based system, in subsequent applications, the physical form of kanbans varied between different organisations. Waller (2003) refers to different applications in which paper tokens were replaced by magnetic tags and more recently bar code systems and laser readers utilised to record the information contained within kanbans into a central inventory database.

Chase et al. (2006) present kanban squares as an alternative to the traditional token-based kanban system. Kanban squares are marked zones on the floor where stacks of containers are stockpiled. Empty zones provide the signal to supplying stations to commence the production of parts. They further refer to the case of Kawasaki which replaced kanban cards with coloured golf balls in one of its engine plants. Station operators would roll a coloured golf ball down a pipe to inform the supplying station that more parts are needed.

Huang et al. (2008) suggest tracing and tracking WIP is a painstaking task the complexity of which becomes enormous in functional layouts producing a high variety of products. They propose the use of wireless manufacturing technology supported by Radio Frequency Identification (RFID) sensors and a wireless information network for the collection and capturing of WIP data in real time. They are emphatic that this form of job-shop reengineering is a promising alternative to reconfiguration. They argue converting functional into cellular layouts inevitably restricts the operational flexibility of job-shops.

### **3.4.2.2 Kanban supporting infrastructure**

The synergy between JIT practices and supporting infrastructure is analysed by Ahmad et al. (2003). They argue tangible elements of JIT including schedule stability, pull control and set-up time reduction need to be supported by an array of infrastructure practices covering quality management, human resource management policies,

product technology and work integration so that their full potential can be exploited. A detailed discussion of the JIT supportive practices is presented in Chapter 2. This section will draw attention to some of the key elements for the implementation of pull production control.

One JIT pull control requisite is uniform machine loadings which is necessary so that operations are balanced. Balanced operations result in smooth flow of materials with the output of each station meeting the requirements of the following station. A balanced line is further one where no machines starve or get blocked and therefore WIP accumulation between stages is minimised. Hill (2012) argues that the key difference between MRP and JIT lies in the principle used to load jobs to machines. More specifically they suggest that whilst loading under MRP is based on EOQ driven by due-dates, under JIT, loadings are determined by throughput rates. Using a throughput rate, that is the actual material consumption rate practically means that the time-phased principle of MRP is substituted by a rate-based principle in JIT production.

A central notion in JIT is that of cycle time. This establishes the rate at which units of finished products exit the production line (Waller, 2003). A common analogy used to describe the importance of cycle time in JIT production is one which compares the cycle time with the takt time, a term denoting the rhythm imposed by the conductor in an orchestra. In JIT production, the takt time is determined by the customer's demand rate. Liker and Meier (2005) identify several deviations which can cause undesirable cycle time variations. These include lack of operator skills, material and tool shortages and defective parts all reinforcing the need for quality management, close relationship with suppliers and other forms of infrastructure to be in place to support JIT.

JIT pull control relies heavily on stable production schedules. All deviations from normal operations e.g. unscheduled product changeovers and machine breakdowns will inevitably decrease the established JIT production rate. Groover (2011) explains that such deviations will be amplified in upstream operations thus preventing the smooth flow of work and causing major line imbalances. Another major JIT pull control requisite is mixed-model production. Chase et al. (2006) claim that building the full mix of products into the daily production schedule is a JIT response strategy to demand variations. Clearly, mixed-modelling requires production in small batches which in turns needs to be facilitated by shortened set-up times.

### 3.4.3 Theory of Constraints (TOC)

In its broad sense, a constraint is anything that hinders an organisation's performance. The sheer realisation of this fact was the inspiration behind the development of the TOC in the early 1980's by a physicist named Goldratt. The theory gained popularity after the publication of a book entitled "The Goal" (Goldratt and Cox, 1984). In addition to the novelty of the TOC, the book sparked the attention of manufacturers by presenting the challenges faced by production managers in the form of a novel.

A point of reference in the book was the goal of any manufacturing organisation, which in line with TOC should be to generate profit now and in the future. Key to achieving this goal is the organisation's performance across three measures (Markland et al., 1998). These are throughput, inventory and operating expense. All these have a distinct meaning in the context of TOC. Throughput measures the rate in which an organisation generates cash through sales rather than production output. The concept of inventory is extended to include apart from raw materials, WIP and FGI, the organisation's assets i.e. buildings, land and machinery. Inventory is capital tied up in anything an organisation could potentially sell. Operating expense represents money spent in transforming inventory to throughput e.g. salaries, wages etc. even scrap. By introducing this terminology, Goldratt's aim was to challenge the traditional accounting and costing thinking and practices and inspire production managers to think of innovative productivity measures and performance incentives.

Synonymous with TOC are the terms Optimised Production Technology (OPT) and synchronous manufacturing (Russell and Taylor, 2009). OPT is the name of the scheduling software developed by Goldratt based on the concepts of TOC. Applications of TOC by General Motors and other manufacturers were referred to as synchronous manufacturing. Slack et al. (2010) suggest the central idea behind TOC is planning to known bottlenecks rather than producing unrealistic plans that simply overload capacity constrained parts of the factory. They summarise some of the overarching principles of TOC in the following points:

- 1) The aim in TOC is to balance flow, not capacity.
- 2) The level of utilisation of a non-bottleneck is not determined by its own capacity, rather another constraint in the system.
- 3) Machine utilisation and activation are not equivalent terms.
- 4) An hour lost at a bottleneck resource causes an hour to be lost in the entire system.
- 5) System throughput and inventory are governed by bottlenecks.

- 6) Process batches should be variable and clearly differentiated from transfer batches.
- 7) Lead times result from the schedule and should not be predetermined, i.e. fixed.
- 8) Schedules should be produced by examining all the bottlenecks simultaneously.

TOC uses a technique called Drum-Buffer-Rope (DBR) to exercise production control at the shop-floor level (Stevenson, 2006; Gonzalez-R. et al., 2010). In this metaphor, the drum is the capacity constrained resource i.e. the bottleneck, which is beating the rhythm of production to establish the throughput rate for the entire system. The buffer element is inventory placed in front of the bottleneck to ensure it does not suffer from material shortages. Finally, the rope represents the input control mechanism used to synchronise the sequence of operations so that the bottleneck can be effectively utilised. This establishes a communication link between the bottleneck and preceding stations ensuring the latter are not running ahead of the bottleneck schedule.

Following its introduction, TOC was often compared with JIT. Johnson and Malucci (1999) present an interesting view suggesting both TOC and JIT are applicable in a variety of production settings including job-shops. Gupta and Snyder (2009) carry out a comprehensive review of the literature comparing TOC, JIT and MRP. Their findings suggest the underlying principles of TOC can be embedded into an existing MRP system. The comparison of TOC with JIT points out that both systems perform equally well, however it is noted that JIT is mainly associated with repetitive production. It is also identified that more empirical studies are needed to examine the performance of TOC.

#### **3.4.4 Kanban variants and hybrids**

The introduction of pull control stimulated extensive investigations of the original kanban system and numerous variations and hybrids were proposed by experts in the field. One of the earliest variations of the kanban system is CONWIP proposed by Spearman (1988). CONWIP is a single-card system, which aims to maintain a constant maximum level of WIP in the system. This is achieved by limiting the total number of cards circulating within the system. A new job cannot be released into the system unless there is a corresponding card that can be attached to it. Once the job is released for processing, the card travels with the job through the system. When the processing of the job is completed at the last station, the card is detached from the job and sent back to the beginning of the process where it awaits to release a new job. During the resting phase of the system, there is FGI at the last station but all other

station buffers are empty. CONWIP can be viewed as a single stage kanban system which exercises pull control at the end of the process and push control at its beginning.

Buzacott (1989) makes one of the first references to the Base Stock mechanism. This pull control policy establishes a base level of finished parts in the output buffer of each stage (Bonvik et al., 1997; Zipkin, 2000). A new customer order is immediately broadcasted to all production stages. It effectively triggers the release of FGI from the output buffer of each stage to the input buffer of the succeeding stage and simultaneously provides the authorisation for the replenishment of the Base Stock of each station. The key advantage of this method is that it is quite reactive to customer demand but its drawback is that it cannot limit the maximum level of WIP in the system.

Generalised Kanban Control System (GKCS) was an umbrella term used by Buzacott (1989) to describe three different pull control policies. These appear in the original work of Buzacott under the names, special (conventional) kanban, reserve stock kanban and back-ordered kanban. The GKCS uses two parameters. These concern the total WIP and maximum inventory at the output buffer of each stage. Depending on the specific values of these parameters, GKCS is reduced to one of these three control policies. Duri et al. (2000) suggest that despite the different names proposed by Buzacott, these effectively correspond to the kanban, CONWIP and Base Stock systems.

Otenti (1991) developed a variation of the kanban system which made it applicable to a semiconductor company. The proposed system, named Modified Kanban System (MKS) groups operations into centres. MKS uses signals to control the inventory of these centres thus minimising the level of monitoring required. The extension of pull control to non-repetitive production environments was also considered by Chang and Yih (1994) who proposed the Generic Kanban System (GKS). GKS uses generic cards which are not associated with specific part types. A fixed number of cards is provided at each station. The system divides production time into two distinct cycles used for kanban acquisition and fabrication. A new job cannot enter the system unless it is matched by a kanban from every station. As soon as the job finishes its processing at a station, the kanban is released and made available for a new request. The system is found to be simpler than conventional kanban and more flexible than CONWIP is dynamic settings.

The Extended Kanban Control System (EKCS) is theoretically developed by Dallery and Liberopoulos (1995). The EKCS uses two design parameters per production stage, namely the number of kanbans and Base Stock of parts at the output buffer of each station. Similarly to the operation of the Base Stock control system, customer demand



is instantly communicated to every stage in the process. The difference however is that parts are released from a stage upstream to a downstream stage only if there is kanban available to provide such authorisation as it is the case in the classic kanban system. Therefore, the EKCS is practically a combination of the kanban and Base Stock systems.

Gupta and Al-Turk (1997) design the Flexible Kanban System (FKS) to address the blocking and starvation of stations typically encountered in production systems controlled by a fixed number of kanbans. FKS employs an algorithm to dynamically adjust the number of cards and is found to outperform the kanban system in dynamic environments affected by variable processing times and uncertain demand. Mohanty et al. (2003) propose the Reconfigurable Kanban System (RKS) as an alternative to the kanban system which is mostly suitable for production systems operating under unstable demand. The RKS adjusts the total number of cards according to the difference between customer demand and the production rate for each part.

Kumar and Panneerselvam (2007) regard variations to special cases of the conventional kanban system. They present a survey of the JIT and kanban related literature. This focuses on different blocking mechanisms developed for the operation of the kanban system and performance measures employed to test its effectiveness. Their survey further classifies previous studies in this area into theoretical and empirical and critically evaluates the effectiveness of proposed solution methodologies and modelling approaches.

Junior and Filho (2010) identify 32 variations of the kanban system. They review these by analysing characteristics of the original kanban system that were retained in the developed variation models. They further discuss the operational aspects of the variations and evaluate their main merits and demerits compared to the original kanban system. They find most of these variations are developed by manipulating the use and number of kanban signals. They argue several promising variations were merely developed as theoretical frameworks and therefore lack empirical testing. Their survey identifies several variations which were found to perform well in dynamic non-repetitive environments.

Kanet and Stößlein (2010) claim the success of Japanese manufacturers and emphasis on waste elimination was not merely a significant contradiction to the costly inventory-centred approach of MRP but the main incentive behind the exploration of push-pull hybrid models. The emergence of the QRM strategy by Suri (1998) drew attention to a hybrid push/pull system named Paired-Cell Overlapping Loops of Cards

with Authorisation (POLCA). POLCA was promoted as the production control component of QRM and was used to synchronise and balance material flow in manufacturing cells. In POLCA, a high level production planning system such as MRP authorises the release of jobs into the cell. However work on a specific order cannot start until there is a corresponding POLCA card. Despite an initial similarity with the kanban system, there are fundamental differences (ibid). As the name of the system suggests POLCA cards are assigned to pairs of cells not different product types. A POLCA card remains attached to a job as the latter travels between two cells and when the processing of the job is completed in the downstream cell the card returns to the upstream cell of the pair. Since its introduction, POLCA was promoted as a suitable hybrid push/pull system for HVLV and customised production carried out in cellular manufacturing systems.

Martinich (1999) describes DBR as a hybrid which implements a pull strategy in stages preceding the bottleneck and push control in all subsequent stages. It is also suggested that MRP, JIT and synchronous manufacturing should not be regarded as mutually exclusive systems. In fact, it is argued that DBR can be combined with JIT production to facilitate pull control in job-shops where JIT scheduling is hindered by the high variety of products and great diversity of their routings.

Nagendra and Das (1999) design a hybrid push/pull architecture that combines the planning functions of MRP with the shop-floor control mechanism provided by kanban-type systems. The proposed framework comprises a kanban card controller which uses the planned order releases generated by MRP to determine the number of kanbans for each time period. Lead times are estimated taking into account the dynamic shop-floor conditions. Dispatching is performed by a module called kanban prioritiser which allows the hybrid system to be applied in non-serial production settings. Despite its good performance in experiments, this is merely a framework which needs further development and practical testing.

Cochran and Kaylani (2008) design a hybrid push/pull control system for multiproduct multistage serial systems. They also develop a genetic algorithm for optimising the safety stocks and number of kanbans of the push and pull elements respectively. They use simulation to test their model. From a practical viewpoint, they find that the implementation of the proposed hybrid system requires layout changes but this integrated approach results in cost savings compared to either pure push or pull control. They further investigate a range of design issues which mainly concern the position and number of junction points i.e. the points which signify the transition from push to pull subsystems.

González and Framinan (2009) develop a hybrid pull system named Customised Token-Based System (CTBS). CTBS establishes control points between all pairs of stations and uses a token-based system to regulate the entrance and flow of jobs to each pair. They use this as the basis for the formulation of a combinatorial problem which seeks to determine for each loop the number of corresponding cards that optimises the system's performance across several criteria. They point out that the application of CTBS in realistic settings will significantly increase the size of the solution space.

### **3.4.5 Kanban design and modelling**

Several studies have considered the design of the kanban system and its variations by analysing key features including the number of cards and job prioritisation. This section will further review studies employing simulation, queuing theory and Markovian chains to model kanban systems.

Martin et al. (1998) develop a tabu search algorithm to optimise the number of kanbans and lot sizes in a generic kanban system operating under dynamic conditions. They argue these two design parameters can directly impact the WIP and cycle time performance of the system. Their algorithm performs well in terms of computational time but has limited success in optimising the multi-objective function considered in their study. They recommend the use of a more sophisticated tabu search algorithm as well as further experimentation with genetic algorithms and simulated annealing.

Dispatching in a small cell producing several different product types is considered by Thesen (1999). Two push rules, namely random and rotation and one pull rule are compared in terms of their throughput rate performance. The implementation of the push rules is quite straightforward however; they require preliminary design work and real-time information in order to determine the proportion of parts to release into the cell. Their performance is significantly compromised in the absence of large buffers. The third rule uses cards to pull parts into the cell. Its performance is good but depends on good fine tuning at system initiation.

The study of Herer and Shalom (2000) can be clearly differentiated from other investigations considering the issue of kanban card setting. Their work adopts a probabilistic and averaging approach in simulating the use of a non-integer number of kanbans in serial and non-serial production systems. Their findings confirm that not only is it possible to operate complex applications of pull control by using a non-integer number of cards but in fact such an approach results in cost savings. This is due to the

fact that inventory holding and shortage costs as well as set-up costs are reduced since less WIP exists in the system compared to when the number of kanbans is rounded up to the next integer.

Yang (2000) reviews four parameters namely, priority rules, number of cards, withdrawal cycle and transfer policy which define the design of single-kanban, dual-kanban and CONWIP systems. The FCFS and Maximum Number of Cards (MNC) priority rules are mainly associated with the operation of the single-kanban and CONWIP systems. The prioritisation mechanism varies in the case of the dual-kanban system as operators make a selection based on the incoming withdrawal cards first before considering the cards posted on the production card post. The withdrawal cycle determines the frequency of rounds performed by material handlers to move withdrawal cards and empty containers to preceding stations and full containers to the stations downstream. Furthermore, two transfer policies are considered. The immediate transfer policy requires the instantaneous transit of full containers and attached withdrawal cards to the station downstream before the imminent withdrawal cycle takes place. In contrast, full containers and attached withdrawal cards are stored and await the next visit by the material handler under the periodic transfer policy.

Framinan et al. (2000) consider input control mechanisms and dispatching rules in manufacturing cells. They compare the WIP and service level performance of input control using a single card count for all product types (S-CLOSED) with that of an alternative input control mechanism which sets individual card counts for each product type (M-CLOSED). They find the combination of M-CLOSED with the SPT dispatching rule exhibits superior performance. However, this comes with the significant practical complexity of establishing a card count for each different product count. In order to benefit from the simplicity of S-CLOSED and good performance of M-CLOSED they develop a hybrid named S-CLOSED/Min(WIP) which follows the single card count operating principle of S-CLOSED but prioritises jobs according to the existing WIP in the cell. The performance of the hybrid mechanism is comparable to that of M-CLOSED but the former is found to be more sensitive to demand variability.

Framinan et al. (2003) identify a significant gap in the CONWIP literature which focuses mainly on card setting and lot sizing whereas other important decisions relating to the operation of CONWIP control appear to be receiving little attention. A number of design parameters which can impact the performance of the CONWIP system are identified. In addition to card setting and prioritising these include the production quota, maximum amount of work in the system, capacity shortage trigger and forecasting of backlogged orders. Their research points out a pressing need for the development of a cohesive

framework that will model and compare the relative importance of these operational parameters on the overall performance of the CONWIP system. It is argued this will help address some of the contradictory results reported in previous comparative studies.

The cyclic sequence in which parts are loaded into a deterministic flow line is considered by Sarin et al. (2006). They develop a new policy which releases products into the line as late as possible ensuring the idle time of the combined bottleneck station that is, the station with the largest sum of processing times for all products is minimised. A new dispatching rule that reduces the size of queues in front of stations is designed to support the new policy. The latter is found to outperform CONWIP and the workload regulation control mechanism, a policy which seeks to maintain a constant level of WIP before bottlenecks when both throughput and WIP are considered.

Bahaji and Kuhl (2008) investigate optimum combinations of order release mechanisms and dispatching rules by analysing their performance in an experimental wafer fabrication setting accommodating HVLV production. They produce a collection of 10 benchmark dispatching rules which they further supplement with four new composite rules and test these under push and CONWIP control. Their simulation results confirm that one of the proposed dispatching rules, namely  $Wt(PTpWINQ)/XF$  demonstrates superior flow time and due date performance irrespectively of the order release mechanism with which it is combined. Their directions for further research recommend further exploration of the potential of these new rules in other HVLV and job-shop settings.

Dispatching rules used in systems controlled by the DBR mechanism are discussed by Gonzalez-R et al. (2010). They argue dynamically switching to a new dispatching rule contradicts the simplicity of TOC and therefore a more meaningful approach would be to select a robust rule that exhibits good performance under variable conditions. They identify nine rules which they test in several scenarios considering the utilisation of the bottleneck, set-up times and machine breakdowns. WIP and lateness measures are used to assess the local and global performance of these dispatching rules. Their findings give useful directions regarding the suitability of the rules under several conditions of variability. However, these results are produced in experimentations performed in a flow line comprising five stages so they are necessarily valid in other production settings.

Modelling pull control systems is an important step in studying their operation, applicability and most importantly assessing their performance. Buzacott (1989)

designed a queuing network to study the behaviour of the GKCS. The GKCS is modelled as a series of multiple server queues and an appropriate blocking mechanism is introduced to describe the three different special classes of the GKCS. The output and input storage points of two sequential stations, the upstream and downstream respectively are represented by linkage stations which are split into two different queues where cards and containers are temporarily stored.

Frein et al. (1995) present a queuing network with a synchronisation mechanism for a 3-stage GKCS. They identify two important GKCS design parameters. The first limits the WIP whereas the second specifies a target for the products that can be stored at the output buffer of each stage. In their model, the linkage between two successive stations comprises two synchronisation stations, each of which has two queues. The queues of the first station represent the storage of finished parts from the upstream station and kanban authorisations to transfer these to the downstream stage. The queues in the second synchronisation station correspond to demands from the downstream stage and kanban authorisations to broadcast these into the stage upstream.

Liberopoulos and Dallery (2000) use queuing networks with synchronisation stations to provide an insightful review of the operational differences of base stock, kanban, GKCS and EKCS. They further model CONWIP which they treat as a special case of the single-card kanban system. Kanban, Base Stock and CONWIP are the three pull control mechanisms considered in the simulation study presented in Chapter 5.

Nomura and Takakuwa (2004) employ simulation to model the operation of a multi-stage flow line controlled by a dual-kanban system. Their model built in Arena 8.0, comprises a workstation module that performs the production and conveyance functions. A match module associates production and conveyance authorisations with WIP whereas station input and output buffers are represented by queues created by a hold module. Experiments are conducted assuming stochastic arrival and processing times but the model is only tested in a small flow-shop that produces only one part.

A continuous-time stochastic Markov chain is developed by Al-Tahat and Rawabdes (2008) to model CONWIP control in a multi-stage multi-product system. The premise for using Chapman-Kolmogorov mathematics is so that the steady-state behaviour of the system can be analysed and interactions between its key design parameters can be observed. The developed model is capable of capturing the performance of COWNIP in relation to multiple performance measures including average makespan, machine

idle time and throughput rate. Comparisons with queuing and simulation models are suggested so that the accuracy of this modelling approach can be verified.

### **3.4.6 Performance comparison of pull control mechanisms**

Duri et al. (2000) present a comparative study of kanban, Base Stock and GKCS. The three policies are assessed according to their service level performance in a production system where customer demand can be satisfied immediately or be backordered. Their research points out that when no delay is allowed in filling orders the three policies offer the same level of performance. The kanban system is therefore seen as the preferred option since it is simpler than GKCS and has better WIP performance than CONWIP. If orders can be fulfilled with delay, the GKCS and Base Stock outperform the kanban policy.

A generic model that simulates the operation of kanban, CONWIP and hybrid kanban/CONWIP is designed by Gaury et al. (2000). An evolutionary algorithm is integrated into the model to optimise its performance by adjusting the number of cards. The hybrid system is that originally proposed by Bonvik et al. (1997) and is one that combines the good throughput of CONWIP with the tighter WIP control of kanban. In this integrated approach, kanban cells are built into the CONWIP mechanism to prevent the release of components into the system beyond a certain limit and thus restrict the total amount of WIP. Simulation experiments are designed to test the operation of these three policies into a single-product serial line. The generated results point to the superior performance of the hybrid system.

In a similar vein, Sharma and Agrawal (2009) integrate simulation with Analytic Hierarchy Process (AHP) in order to compare the performance of kanban, CONWIP and hybrid Kanban/CONWIP in an analogous production setting operating under stochastic demand. Criteria including machine utilisation, WIP, service level, throughput, unsatisfied demand and total cost are assigned different weights and used to assess the performance of the three pull control policies considered. Due to the conflicting nature of the performance objectives, the simulation results fail to identify a single best pull control policy. The performed sensitivity analysis further reveals that the obtained results are specific to the exact weightings assigned to the performance objectives in each experiment and therefore hardly generalisable. The overall model is experimental and limited to single-product serial lines.

Liberopoulos and Koukoumialos (2005) investigate the diffusion of Advanced Demand Information (ADI) in a single-product multiple-stage system controlled by a Base Stock

or hybrid base stock/kanban mechanism. These mechanisms are adapted to offset demand information by a fixed supply lead time at each production stage. Their main findings suggest that in the presence of ADI, supply lead times and number of kanbans can be minimised whilst Base Stock levels can be progressively reduced until they drop to zero.

Kanban and CONWIP performance is compared by Khojasteh-Ghamari (2012). Their approach uses activity interaction diagrams and critical circuits. Activity interaction diagrams represent processes and queues. Circuits are chains of activities, with the longest circuit (critical) determining the maximum cycle time in the system. Their results indicate that Kanban implemented on a serial production line can achieve a given throughput rate with less WIP compared to CONWIP. This superior performance however is affected by process characteristics such as processing times.

Riezebos et al. (2009) review the operational differences of kanban, CONWIP and POLCA. These are analysed across several elements including signal type, level of automation and applicability to MTS and MTO production environments. An overview of their performance is presented in the sidelines of their research. More specifically they consider the workload balancing performance of these mechanisms which is determined by their ability to minimise throughput by minimising WIP before bottlenecks and releasing orders in a manner that avoids bottlenecks. They argue POLCA is superior in that respect as the other two mechanisms do not have this capability. However, this finding is drawn from a theoretical analysis and is rather short-sighted. It contradicts the findings of empirical studies such as that of Khojasteh-Ghamari (2008) which clearly suggest otherwise.

Jodlbauer and Huber (2008) use simulation to compare the service level performance, stability and robustness of MRP, kanban, CONWIP and DBR. Their approach employs an evolutionary algorithm that optimises operational parameters of these control mechanisms, for example, lot sizing and safety stock for MRP, number of cards and container sizes for kanban, the WIP cap and work-ahead window of CONWIP and finally shipping and bottleneck buffer in the case of DBR. The performance of the four control mechanisms is assessed under various conditions of variability concerning demand, set-up times and machine availability. Their simulation results indicate that the best service level performance is obtained under CONWIP control which however does not maintain its robustness when variability comes into play. The main weakness of the kanban system concerns the optimisation of its parameters which is found to be quite laborious. CONWIP and DBR are simpler in that respect. Interestingly, MRP is



found to be the system offering the highest stability. Their research produced significant findings which are nevertheless limited to flow-shops.

The above review leads to a number of interesting observations. Firstly, there is not always a tightly argued justification for the selection of those specific pull control mechanisms the performance of which is put under the spotlight. Different studies appear to be focusing on different sets of original pull control policies and/or hybrids. The approach adopted in these investigations varies. It is noteworthy that several develop theoretical models which are not empirically tested. Conversely, empirical studies consider simplistic experimentation settings which are far from real-life applications. In the absence of benchmark problems and a comprehensive unified framework for these comparisons, their findings are not safely comparable. Finally, in their majority comparative studies consider serial systems thus revealing an obvious gap that can be filled by analysing the performance of pull control mechanisms in job-shops.

### **3.5 Chapter summary**

This thesis posits that pull production control introduced in the context of lean manufacturing for repetitive flow systems can be extended to non-repetitive job-shops employing functional layouts. This transferability test is proposed on the basis of the proclaimed superiority of pull control and indisputable success of JIT production as discussed extensively in chapter 2.

This chapter provides an in-depth analysis of the configurations and shop-floor layouts of flow-shops and job-shops. The aim was to identify characteristics which make these systems suitable for mass and customised production respectively and further examine practical implications of their features for scheduling and control. The analysis highlighted the complexity of production control in job-shops. This complexity stems from the routing diversity of the products manufactured in these systems. It is further exacerbated by the disconnected production stages typically encountered in job-shops. The thesis aims to contribute to research and scholarship by validating whether, despite the inherent complexity of job-shops, it is possible to extend pull control in these systems. It further seeks to determine whether the implementation of pull control will bring about a significant performance differential.

Scheduling was considered in the broader context of the planning and control hierarchy. Production planning is often closely associated with MRP systems. This review pointed out that order release systems including MRP do not negate scheduling

and control. In fact, both systems must co-exist. The planning function generates order releases which in turn drive the scheduling process.

This chapter stressed that scheduling and control are interwoven practices. Therefore, an experimentation model developed to test the application of pull control mechanisms in job-shops should simultaneously imitate the scheduling and control functions performed in their context. For this reason, an extensive review of production scheduling was carried out. The review identified common dominant concepts in flow-shop and job-shop scheduling.

Loading is a scheduling function performed in flow-shops and job-shops which seeks to allocate jobs to work centres. Loading is tightly associated with the line balancing problem in assembly lines. It was found that line balancing is crucial in serial lines with closely interlinked stages. This is due to the fact that work centre blockages caused by workload imbalances can disrupt and even bring the entire line to a halt. The repercussions of sub-optimal loading are less severe in job-shops. Sequencing was identified as the second scheduling function performed in flow-shops and job-shops. Sequencing concerns the prioritisation of jobs assigned to work centres. Given the high variety of products manufactured in job-shops, sequencing in this type of production systems can become an increasingly complex task. The review revealed the following shortcomings of the research carried out to date:

1. The scheduling literature, irrespectively of its focus, either on flow-shops or job-shops, appears to be primarily concerned with solving scheduling optimisation problems. In seeking to address NP-hard problems, it mainly relies on simplistic problem formulations which have limited practical value from an industrial viewpoint. It is further subject to other limitations, for example it fails to consider dynamic scheduling conditions.
2. In an attempt to simplify the scheduling problems considered, loading and sequencing decisions are treated in isolation. This is found to be the case particularly in the flow-shop scheduling literature, where studies attempting to simultaneously solve the joint line balancing and sequencing problem have emerged only recently.
3. Scheduling problems are treated separately from production control, despite the integrated nature of these two practices.

Through the review of the literature, a conceptual framework for job-shop scheduling was developed. This comprises job lists and due dates. It also includes backward scheduling, loading and dispatching protocols. Constraints imposed by the capacity of the production system itself and availability of raw materials are also considered. The framework views scheduling in its dynamic context, and accepts that schedules can be

disrupted by unexpected events such as machine breakdowns and order cancellations. It further comprises a list of performance metrics used to assess the quality of the generated schedules. This conceptual framework, supplemented by three pull control mechanisms forms the basis for the design of the job-shop scheduling system modelled in chapter 5.

The final part of this chapter captures the essence of production control which forms the core of this thesis. Initially, the discussion focuses on the underlying logic of push and pull control, linking the former with MRP and the latter with the waste elimination ethos of JIT. DBR proposed in the context of TOC is recognised as another form of production control which combines the push and pull production control policies.

The Kanban system is identified as the first mechanism introduced to exercise pull control in assembly lines. The mechanics and operating principles of Kanban are discussed in detail drawing attention to several hybrids and variants of the system. Simulation and queuing networks are identified as the main approaches adopted to model the operation of pull control mechanisms. This is vital evidence in support of the selection of simulation as the modelling technique employed to answer the research questions posed in this thesis. The specific type of simulation employed in this research, that is, agent-based will be further appraised in chapter 4.

The review of pull production control pointed to a substantial stream of literature presenting comparative studies of the performance of different pull control mechanisms. Their analysis led to a number of intriguing observations:

1. There is not always a tightly argued justification for the selection of specific pull control mechanisms the performance of which is put under the spotlight. Different studies appear to be focusing on different sets of original pull control policies and/or hybrids. There is a tendency however to focus on Kanban, CONWIP and base stock. This finding is of significant value as it provides the main justification for the incorporation of these three pull control mechanisms into the simulation model developed in chapter 5.
2. The approach adopted in comparative investigations varies. It is noteworthy that several develop theoretical models which are not empirically tested. Conversely, empirical studies consider simplistic experimentation settings which are far from real-life applications. In the absence of benchmark problems and a comprehensive unified framework for these comparisons, their findings are not safely comparable.
3. In their majority comparative studies consider serial systems thus revealing an obvious gap that can be filled by analysing the performance of pull control mechanisms in job-shops. This finding reinforces the rationale for this research.

The next chapter reviews scheduling approaches. Simulation modelling is differentiated from descriptive techniques which are used mainly to solve scheduling optimisation problems. The chapter justifies the selection of agent-based simulation by arguing it is a technique capable of handling the modelling complexity of pull control applied to job-shops. It proceeds to survey applications of agent-based simulation in the scheduling and control of production systems.

## **4 Application of simulation techniques in Production Planning and Control**

Modelling approaches have supported decision-making in various fields within the discipline of Operations Research (OR). Ranging from simple table-top physical replicas to advanced mathematical programs and sophisticated computer-based models, the power of modelling has been tested extensively in academic research and industrial practice. Computer simulation is a distinct form of modelling employed to observe the behaviour of real-world systems. Driven by gains in computing power, simulation provides a powerful yet cost-effective tool allowing critical decisions to be made by studying and experimenting with the model, that is, an abstracted form of the system, not the system itself.

Advances in the field of Artificial Intelligence (AI) have led to a new form of simulation built on the fundamental notion of intelligent software agents. Due to their distributed nature, agent-based simulation models are particularly suited for decentralised environments. Given the inherent decentralised nature of production control, the latter serves as an appropriate testbed for agent-based modelling.

The chapter begins by discussing the use of modelling in decision theory. It differentiates between two dominant modelling approaches, namely, prescriptive and descriptive. It argues in favour of the suitability of descriptive modelling approaches for applications where the main objective is to study, not optimise, the modelled system's behaviour. A further distinction is drawn between conventional and non agent-based simulation. The chapter highlights the computational efficiency and suitability of agent-based simulation for distributed environments. It proceeds to review applications of simulation in the wider context of production planning and control, gradually focusing on the use of conventional and agent-based simulation in the introduction of pull control in job-shops.

This chapter adds to the thesis in three respects. Firstly, it identifies the need to use agent-based modelling in order to achieve the main research objective of this study, that is, to assess the feasibility of extending pull control techniques designed for repetitive production lines into non-repetitive job-shops. Since this is not an optimisation problem but rather one concerning the operation and behaviour of production systems under certain control policies, descriptive modelling and simulation in particular is the most suited approach. Agent-based is argued as the preferred form of simulation on the basis of its ability to cope with the intricacies of distributed systems as is the case in decentralised pull production control. Secondly, the chapter reviews

key stages of the modelling process and discusses how other chapters of the thesis feed into these stages. Thirdly, by presenting an extensive review of applications of conventional and agent-based simulation in production planning and control it identifies major limitations and research gaps in existing attempts to apply pull control in job-shops and in doing so reinforces the rationale of the thesis.

The chapter is organised as follows. Section 4.1 outlines the scope and purpose of prescriptive and descriptive modelling. It discusses the main components of simulation models, key stages in the modelling process and appropriateness of simulation for behavioural decision-making. Conventional simulation techniques are reviewed in section 4.2 which draws particular attention to discrete-event simulation. Section 4.3 reviews the characteristics of intelligent agents and appraises the suitability of multi-agent systems for modelling distributed production control environments. Applications of simulation in production planning and control are reviewed in section 4.4. This provides a critical evaluation of the limited research concerned with the use of simulation in extending pull control in job-shops. The conclusions of this chapter are presented in section 4.5.

#### **4.1 Modelling in decision-making**

Management science and major fields under this banner including Operations Research, decision theory, systems thinking and management disciplines have been dominated by quantitative modelling as the main form of research tool. Mingers (2006) accepts that quantitative modelling broadly encompasses mathematical programming, combinatorial heuristics, statistical methods and simulation.

As the term suggests modelling relies on the fundamental notion of models. A model is generally regarded as an abstraction of real-world events, processes, facilities, systems and phenomena. Models constitute representations of the real-world that scientists use to study aspects of the systems they emulate. These particular aspects may be related to the structure of the system, conformity with scientific laws, principles of operation as well as behaviour under specific conditions. Giere (2004) proceeds to suggest that models used in science can be quite heterogeneous.

Indeed, models can take various different forms. They can be physical replicas, presented as full-scale or subscale versions of a real-world system (Kelton, et al., 2007). Subscale models of an aircraft in a wind tunnel used to test the aerodynamics of the aircraft before it is built, dummies employed in crash-tests to assess safety performance and flight simulators used for training are a few examples. Physical

models further extent to blueprints and specifications used heavily in engineering (Brockman, 2009). Models can be logical or mathematical relying on simple algebraic equations, complex differential calculus or statistical techniques. Weather forecasts and econometric models are cases in point in this category. Similarly, spreadsheets used in risk modelling and analysis or more advanced virtual approximations of real-world systems, for example a simulation of a distribution network comprising factories, warehouses and transportation links are instances where computer-based simulation can be used (Buede, 2009).

Although models are not a panacea in management science, their use is incumbent for a number of reasons. Ragsdale (2008) recognises that models result in the simplification of most often complex real-world problems. Provided that models are valid, they provide simplified versions of reality which are practically easier to study and investigate. Simplification results in cost, time and risk minimisation. Replicas are also less expensive to build. They are used to identify design flaws, which in turn can be addressed in a cost-effective manner. So the risk of design errors and abortive work is minimised. Models can be created considerably faster than the real-world systems they imitate. Therefore, they ensure vital information about the structure, operation and behaviour of systems is collected on a timely basis.

Pidd (2009) emphasises replication and safety as two additional benefits of modelling. Models allow a recursive experimentation approach, which facilitates the replication of statistical variation. In other words, several sets of experiments can be carried out using the same model in order to test its behaviour under variable conditions. Quite importantly, the use of models ensures human subjects are not exposed to health and safety hazards. Irrespectively of the purpose for which models are used, either in prototyping or experimenting with existing systems, modelling allows the scientific study and investigation of real-world systems in an unobtrusive way.

#### **4.1.1 Prescriptive versus descriptive modelling**

The selection and suitability of the available modelling and solution approaches are determined by considering what these will be used for, that is, the aim of the scientific investigation. Modelling approaches are broadly classified into prescriptive and descriptive. Shapiro (2007) argues that prescriptive modelling adopts an authoritarian approach by seeking to prescribe how a decision can be made to reach the best possible outcome in a given situation. Another synonym for prescriptive modelling is

normative modelling. What is inferred by the use of the term normative is the set of norms or axiomatic rules that decision-makers need to follow to achieve optimality.

Prescriptive models therefore seek to solve primarily optimisation problems. T' kindt and Billaut (2006) explain that optimisation problems can be mathematically formulated by determining three components. Firstly, an objective function that the model aims to minimise or maximise. Secondly, a set of decision variables that in turn determine the value of the objective function. Finally, optimisation problems require a set of constraints that allow decision variables to take only certain values and hence serve as bounds for the solution space.

Depending on whether constraints exist or not in an optimisation problem, the latter is said to be constrained or unconstrained. Giordano et al. (2009) propose a classification framework for optimisation problems. In line with this, the presence of one or more objective functions results in single and multi-objective problems respectively. Depending on whether decision variables take integer, real (time-dependent) or mixed-integer values the resulting optimisation problems are termed integer, continuous and mixed-integer.

Brandon-Jones and Slack (2008) point out that due to the polynomial computational complexity of most classes of optimisation problems, satisficing techniques have emerged aiming at identifying good quality sub-optimal solutions. Simple rule-of-thumb techniques used to solve such approximation problems without guaranteeing convergence and optimality are referred to as heuristics. Sarker and Newton (2008) present an overview of widely adopted heuristics including hill climbing, simulated annealing, tabu search, genetic algorithms, ant colony optimisation and memetic algorithms. Jain and Meeran (1999) provide a detailed taxonomy of optimisation and approximation techniques that have been specifically applied to the deterministic JSSP.

By contrast to prescriptive modelling, the descriptive approach is mainly suited for behavioural decision-making. Its main function is to clarify the behaviour of the system under specific conditions (Parnell et al., 2008). The failure of human beings to follow normative axiomatic rules leading to often erroneous judgment led to the emergence of the behavioural decision theory paradigm. Hodgkinson and Starbuck (2008) explain that descriptive modelling aims to capture actual, suboptimal behaviour and analyse the effect of human bias on decision-making. Extending this principle to any real-world application, descriptive models seek to conceptualise the underlying structure of a system and delve into its behaviour. Ragsdale (2008) regards simulation as one of the main management science modelling techniques associated with descriptive modelling.



### 4.1.2 Simulation suitability and modelling process

Simulation starts with the process of abstraction, that is, the creation of a valid model which provides a good approximation of the real-world system under study. The next step in simulation involves subjecting the model to repetitive testing and experimentation in order to study its behaviour (Sokolowski and Banks, 2009). Providing further insight into this process, Borshchev and Filippov (2004) explain that simulation seeks to analyse this behaviour by executing the model and observing its discrete or continuous state changes over iterative runs.

Having associated simulation with descriptive decision theory, a field concerned with how real-world systems actually behave, it is necessary to take a fundamental look at the type of systems that computer simulation is best suited for. Pidd (2009) asserts that systems to which simulation has been successfully applied tend to share the following set of common attributes:

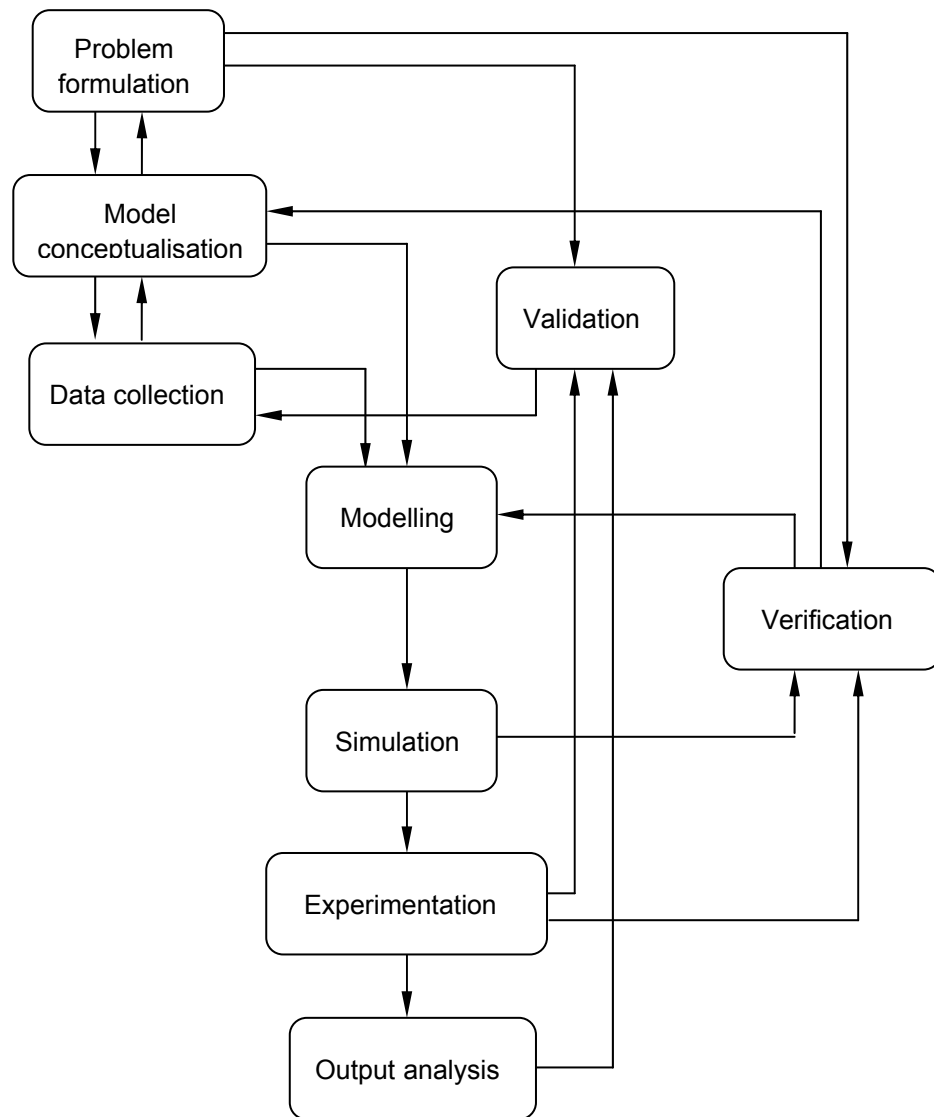
- They exhibit dynamic behaviour, which varies through time. Inherent variations may be attributed to known factors that can be described using mathematical equations or may be due to random and unpredictable sources of variability.
- They are interactive. By definition, systems comprise a number of interdependent components which interact in order to achieve common goals. System components may also interact with objects in the system's environment.
- Reality is complex and multifaceted. Therefore, most real-world systems modelled using simulation are complicated. The intrinsic interactions and dynamics discussed above produce the distinctive and complex nature of such systems.

Sterman (1991) points out that every simulation comprises two major components, namely the infrastructure component which reproduces the physical structure of the system and the behavioural component which encapsulates the set of decision rules that determine the system's behaviour and performance.

Wainer (2009) discusses the phases in the modelling process which will help develop these two components into a full-scale simulation model. These are illustrated in Figure 4.1. It is evident that these modelling phases are not entirely sequential as the overall process includes iterative loops.

An outline of the phases is presented below:

1. Problem formulation. This phase aims to outline the scope, boundaries and objectives of the investigation. It can also determine the feasibility of the study.



Source: Wainer (2009)

Figure 4.1 Modelling steps

2. Model conceptualisation. This is a high-level description of the two fundamental components of the simulation, that is, the system's structure and behaviour. The conceptual model comprises the system's objects, their interfaces and attributes.
3. Data collection. This phase is concerned with the sourcing of input data that will be used to run the simulation model and collect output statistics during the experimentation phase. Decisions made in this phase concern the use of deterministic and/or stochastic data.
4. Modelling. Using the conceptual model developed in phase 2, a detailed representation of the real-world system under investigation is created. The

model designed in this phase is fully specified. Care is also taken to outline the model's underlying assumptions and limitations.

5. Simulation. This phase involves the implementation of the specification model using simulation software. The developed simulation model is an executable program which will provide the experimental framework for the testing carried out in phase 7.
6. Verification and validation. Verification is the process of determining if the behaviour of the simulation model is consistent with what is outlined in the model's specification. On the other hand, validation ensures the behaviour of the simulation model corresponds with that of the real-world system. If both of these processes reveal deviations, the specification and simulation models, as appropriate, may need to be reviewed and refined.
7. Experimentation. This phase includes executing the simulation model and recording the output of the simulation runs.
8. Output analysis. Simulation statistics are analysed to understand the behaviour of the system. Visualisation can shed more light into the observed behaviour.

The credibility of the output generated at the end of the simulation runs, depends heavily on the correctness of the model. Verification and validation are the two processes that help researchers gain confidence in the simulation model. Kelton et al. (2007) use manual simulation to illustrate the mechanics of simulation. Manual simulation involves producing a chronological list of the model's events and diagrams which graphically track changes in data collected from statistical accumulators. Wainer (2009) argues informal verification and validation techniques are founded solely on human reasoning. Therefore, manual simulation can be viewed as such an informal technique.

Verification and validation are further facilitated by visualisation. The use of animated graphics is accepted to enhance our understanding of the model and ability to interpret the output data (Sokolowski and Banks, 2009). Another important technique which can be used for verification and validation is sensitivity analysis. This involves changing the values of input data to observe the effect these changes have on the model's output (Saltelli et al., 2008).

This section initially outlined the surge of interest in modelling and the crucial role of models in facilitating research in the field of OR. This thesis posits that it is possible to apply pull control policies introduced in the context of lean production in non-repetitive job-shop systems. One possible way to provide answers to the research questions framed in this thesis would be to introduce and test these policies in a real-world

setting. However, it is argued that such an approach would be cumbersome and prohibitive due to the cost, time and risk implications.

Having determined the compelling reasons for resorting to a modelling methodology, the section distinguished between prescriptive and descriptive modelling approaches. It was suggested that prescriptive analysis is primarily concerned with optimisation which is not within the scope of this research. By contrast, descriptive modelling lends itself to the study of actual yet suboptimal behaviour and is therefore better suited for achieving the research objectives of this thesis. The section introduced simulation as the dominant approach in descriptive modelling.

## **4.2 Conventional simulation**

The focus of a given study on optimisation or merely the analysis of a system's behaviour is the sole criterion for the selection between a prescriptive and descriptive modelling approach. The characteristics of the modelled system are those that determine the most suitable simulation technique to use.

Models are classified into static and dynamic, depending on whether their state changes as a function of time. Shapiro (2007) explains that how these changes occur over time further differentiates dynamic models into discrete, continuous or mixed continuous/discrete. By definition, discrete models display state (status) changes in distinct and unconnected points in time whereas changes are observed in inseparable points in time in continuous models. Mixed models show change patterns observed in both continuous and discrete systems. A final distinction can be made between stochastic (probabilistic) and deterministic models depending on the randomness of their input variables.

Jahangirian et al. (2010) present an extensive review of applications of simulation in the manufacturing and business sectors. Their research focuses on the attention that specific simulation techniques received in theoretical and empirical studies. They identify Discrete-Event Simulation (DES) as the most popular and widely applied technique, followed by System Dynamics (SD), hybrid simulation, Agent-Based Simulation (ABS), Monte Carlo and intelligent simulation. Five more simulation techniques occupy the lowest ranking positions. In the context of their research, hybrid is a term used to describe the integration of two or more simulation techniques whereas intelligent simulation results from the combination of simulation and AI techniques used to solve complex, real-life problems.

Monte Carlo is the forerunner of modern simulation. It is mainly found in mathematical models which estimate the values of stochastic variables using random sampling (Kalos and Whitlock, 2009). Its applications range from synthetic data generation used for testing, to computer games and further spreadsheet-based studies in valuations and risk analysis. Çayirci and Marinčić (2009) argue that Monte Carlo is also suitable for static problems which are not analytically tractable.

Different simulation techniques are appropriate depending on the nature of dynamic systems and whether the latter are continuous or discrete. SD originally termed industrial dynamics is founded on the principle that every physical control system behaves similarly to an organisational system. The observed system's behaviour is assumed to be the result of variables which exist both within the system and its environment (Jackson, 2003). The interactions of these variables lead to complex causal relationships which in SD are described using sets of interconnected feedback loops showing positive (reinforcing) and negative (balancing) impact. These dynamic interactions are the primary determinants of the continuously changing system status.

Other fundamental concepts in SD are those of levels and rates (Pidd, 2004). Quantities of different elements of the system, e.g. information and cash flow through the systems in changing rates. Their accumulation creates levels or stock. Initially, such continuous system changes were mathematically described using differential equations that were read and executed by SD software. However, modern SD packages allow the user to build feedback loops as well as stock and flow diagrams which get converted into sophisticated simulations. SD is particularly suited for the study of continuously evolving socio-economic systems (Ford, 2009) with numerous applications in business strategy (Meyers, 2010) and environmental studies (Fishwick, 2007) among other areas.

By contrast, DES is mostly suited for the modelling of discrete systems. DES models the operation of a system through a series of chronological events, that is, discrete points in time that mark changes of the system's state (García-Hernando et al., 2008). DES is named after the major concepts of the discrete modelling approach. Systems which are modelled using DES comprise entities. In their abstract form, entities represent objects of the observed system which interact to achieve set common goals (Taha, 2011). A job in a manufacturing system is a case in point. The properties of a given entity are termed attributes. For the case of the job used in the previous example, attributes may be related to the priority of the job, due-date and routing. Other important features of DES models include activities and states (Banks et al., 2010). Activities are periods of a specified length which result in changes of state, for instance the completion of processing of a given job. The system's state can be determined by

collecting all those variables that can provide a full description of the status of the system at any point in time.

Pidd (2009) highlights manufacturing as one of the main application domains for DES. He suggests DES is used in the design of new production facilities as well as in existing plants to either evaluate the effects of new control policies or periodically check the operation and performance of the system. Wang et al. (2011) subscribe to this view and promote DES as a decision support tool suitable for the design and improvement of production systems. DES is recognised as a powerful tool for modelling manufacturing systems by Pichappan et al. (2011). They maintain it can handle the complex and dynamic nature of production systems with high credibility and flexibility.

This section identified the most popular and widely used non ABS techniques. It proceeded to outline their appropriateness for modelling either static or dynamic systems. In the case of dynamic systems, a distinction was drawn between continuous simulation and DES depending on how the system's state changes as a function of time. Emphasis was placed particularly on DES and the discussion substantiated its suitability in modelling discrete-event driven production systems. This analysis provided the necessary justification for the selection of a simulation technique that will be founded on the principles of DES. The fundamental notions and building blocks of DES were reviewed.

### **4.3 Agent-Based Simulation (ABS)**

Analysing trends in discrete modelling, Banks et al. (2010) claim discrete agent-based systems are becoming the vehicle for remarkable developments in the field of computer simulation. DES remains a fundamental tool for the design and implementation of distributed agent-based modelling systems (Flynn et al., 2011). Uhrmacher and Weyns (2009) explain how sequential and parallel discrete-event simulators can be combined to synchronise the behaviour of agents with physical time. Schuldt (2011) argues that event-driven time progression is preferred to time-stepped progression in ABS as the former avoids unnecessary mapping of time steps which do not correspond to any events.

#### **4.3.1 Intelligent Software Agents**

Since their inception, software agents were regarded a subfield of AI. Wooldridge (2002) contents the nature of this relationship is a matter of ongoing debate and highlights some important points of differentiation. More specifically, it is suggested that

whilst AI is concerned with components of intelligence, the field of agents seeks to encapsulate these components into computational software entities. AI is also criticised for failing to capture the social aspects of agency until very recently. Nevertheless, the origins of agents can be traced in AI which is itself an interdisciplinary field with influences from diverse areas including neuroscience, cognitive psychology, linguistics, mathematics and computer engineering.

The lack of a universally accepted definition of what a software agent constitutes can be attributed to the cross-fertilisation in the research conducted in the above areas. Definitions of agents abound (Wooldridge and Jennings, 1995; Sycara, 1998; Ferber, 1999; Jennings et al., 1998; Wooldridge, 2002; Wooldridge and Dunne, 2006). It is noteworthy that a plethora of terms is used to describe agents, for example, as control units, software components, computer programs, problem-solvers, decision-making entities etc. Furthermore, there appears to be no consensus in terms of the capabilities that software agents need to possess.

The term agent points to the notion of agency, that is, appointing someone to complete an assigned task on one's behalf. In order to achieve their goals human agents would typically be expected to interact with others. In line with this analogy Turban et al. (2011, p. 613) define an intelligent agent as "*an autonomous computer program that observes and acts upon an environment and directs its activity toward achieving specific goals*". They note agents are also able to learn in order to advance knowledge already built into them.

Wooldridge and Jennings (1995) introduced the weak notion of agency. This refers to a set of basic properties which are generally accepted to characterise all software agents. The first property is autonomy, which is noted in several definitions including the one above. Autonomy refers to the ability of agents to operate independently without direct interventions by humans or software counterparts and have at least partial control of their internal state and actions. Reactivity and proactiveness are two properties which describe the manner in which agents act within their environment. Reactive agents respond to changes and stimuli they perceive. Proactiveness denotes the ability to exhibit goal-directed behaviour by taking initiative. As agents have partial view of their environment, they need to use social interaction as a means of expanding their sphere of influence. Social ability is the fourth property in the weak notion of agency. It presumes communication with other agents. Social agents are also equipped with coordination and negotiation skills which ensure agents act collectively in a well-defined manner avoiding conflicts.

North and Macal (2007) assert each agent possesses a unique set of attributes and behavioural characteristics which determine their diversity and heterogeneity. Attributes define what an agent represents, for instance a machine in a production system. Behavioural features include perceptual tools used to sense the environment, decision-making protocols, plan projection mechanisms to assess the likely outcomes of their decisions and finally adaptation and learning capabilities. Russell and Norvig (2010) associate adaptation with agent intelligence and rationality. They explain that autonomous agents use their ability to learn as a process of building on prior knowledge and thus improving experience of their environment and therefore their own performance. Adaptation also enables agents to cope with dynamic environments and the volatility of changes taking place therein.

The stronger notion of agency encapsulates mental attitudes more closely associated with human practical reasoning. As a result it is perceived as more contentious (Wooldridge and Jennings, 1995). Luck et al. (2004) defend the importance of developing appropriate control architectures to model different behavioural mechanisms in agent systems. They differentiate between reactive, deliberative and hybrid agent systems. Reactive agents are merely equipped with stimulus-response decision rules. Deliberative agents are arguably more sophisticated. The majority of such deliberative architectures are founded on the Beliefs-Desires-Intentions (BDI) model of rational agency which they endorse as the most successful form of agent architecture.

The BDI model is associated with cognitive agents which use goal-oriented inference mechanisms to achieve their goals (Dunin-Keplicz and Verbrugge, 2010). In this architecture, beliefs represent knowledge agents have accumulated over time about their environment, whilst desires correspond to the agent's goals. Intentions are a subset of the agent's desires that agents select to commit themselves to until changes arise that force them to abandon their intentions. The process of deliberation is concluded by deciding on actions and developing these into complex plans that will be used to fulfil the agent's goals.

#### **4.3.2 Multi-Agent Systems (MAS) and applicability**

An agent working in isolation would fail to deliver most of its purported benefits as it would be unable to cooperate with other agents and thus compensate for the impartial knowledge provided by its designer. The most successful implementations of agent technology can be found in models based on communities of intelligent agents. The



stream of MAS research is concerned with the design and implementation of interacting social agents in distributed environments (Bussmann et al., 2004).

According to Di Marzo Serugendo and Karageorgos (2011, p. 111), a MAS is a set of interacting agents which are situated in a common environment and collaborate to complete a common and coherent task. In doing so, each agent seeks to achieve its own sets of objectives which may be conflicting. Paolucci and Sacile (2005) compare Multi Agent-Based Simulation (MABS) with conventional non agent-based simulation. They argue MABS is characterised by the following distinguishing features:

- The components of the modelled system are represented by interacting agents.
- The autonomy of agents results in an inherently distributed modelled system.
- Agents can be added or removed from the system dynamically during run time, thus enabling modelling flexibility.
- Modelling efforts focus on simulating agent behaviour at the micro level. The system-wide behaviour at the macro level emerges naturally from the interoperability of agents.

Schneeweiss (2003) points out that MAS have been successfully applied to complex Distributed Decision-Making (DDM) problems and are further synonymous with Distributed Artificial Intelligence (DAI). The distributed nature of MAS is not merely a natural consequence of their combined diversity and autonomy. In this context, the term distributed also points to problem decomposition and allocation of smaller more manageable tasks to a population of heterogeneous yet cooperative agents. Luck et al. (2004) concur and suggest that owing to the fact that agents have partial knowledge and limited viewpoint of their shared environment, they are able to concentrate on solving their own sub-problems more efficiently at a local level. Luger (2009) is emphatic in explaining that DAI is not concerned with low level technical issues of distributing processing power and designing parallelisation algorithms. He asserts that instead, the main supposition in DAI is that distributed and cooperative intelligence is a radical move away from centrally storing knowledge which is solely manipulated by a single general-purpose control unit.

The attractiveness of agent-based modelling has been associated with the agent perspective it takes (North and Macal, 2007). People can relate to the mental attitudes and goal-oriented reasoning of agent-based architectures far better than how they respond to other types of models. In terms of applicability, the diffusion of MAS has been mostly documented in systems sharing the following characteristics:

1. **Complexity** which specifically arises from the interaction and interoperability of multiple components (Ioannou and Pitsillides, 2008). Unlike conventional simulation, which models complexity at the whole system level, MAS adopt a bottom-up approach. Behaviour is modelled at the agent level and the computational framework runs through numerous agent interactions to replicate the system-wide complexity (North and Macal, 2007).
2. **Heterogeneity** stemming from components of the modelled system that have different attributes, perform different actions and follow diverse decision protocols (Sage and Rouse, 2011). Using heterogeneous agents which have different knowledge and capabilities constitutes the representation of such systems at aggregate level possible.
3. **Volatility** that results from unpredictable and often constant changes. Such dynamic environments require the autonomy and flexibility exhibited by agents (Wooldridge, 2002). Agents are built with learning abilities that enable them to adapt their plans in response to such unexpected changes.
4. **Decentralisation** (distribution) of information, activities and decision-making across the multiple heterogeneous components of the modelled system (Giachetti, 2010). Representing such systems requires a similar distributed approach in which the system-wide behaviour emerges from the interactions of agents that have fractional (distributed) knowledge and data. Control in these systems is also decentralised as there is no global supervisor (Di Marzo Serugendo and Karageorgos, 2011).
5. **Openness** (scalability) which means that the system size and resulting complexity can change by adding/removing components. Tanenbaum and Van Steen (2007) present scalability as one of the main goals in developing distributed systems. MAS allow agents to be created/removed during the simulation (Paolucci and Sacile, 2005), whilst conventional simulation models typically represent closed systems.
6. **Asynchronism** (parallelism) based on individual threads of execution (i.e. tasks) which are not concurrent but instead indicative of the complex and dynamic nature of the system. Autonomous agents have the ability to work independently and decide when to act upon their environment, update their knowledge and interact with other agents (Nguyen and Jain, 2009).

Production control systems exemplify all the above characteristics. Their intrinsic complexity arises from the heterogeneity of the entities that need to be modelled in such environments, that is, jobs, machines, tooling, operators and so forth. Complexity is also the result of conflicting objectives that need to be achieved in order to maximise

production control efficiency. Production systems are dynamic as they are subject to changes occurring both in their own setting and external environment, for instance machine blockages and order cancellations respectively. Production planning and control functions are inherently decentralised with strategic planning performed at higher levels of the organisation whilst scheduling and control are carried out at the shop-floor level. Production systems are subject to dynamic conditions that impact on system size, for example machine breakdowns reduce the number of operational machines. Finally, production control functions are asynchronous as they are performed as and when required. A typical example is dispatching which needs to be carried out every time a new job enters the input buffer of a work centre.

Being inherently distributed and dynamic, the field of production control has been specifically targeted for applications of agent-based modelling (Panurak, 2000). The current challenges facing manufacturers are outlined by Bussmann et al. (2004). These force manufacturers to outlive competitors on the basis of cost minimisation, innovation, customisation, shorter lead times and better customer service. Agility, responsiveness, flexibility, re-configurability and leanness can lead to significant competitive advantage. However, these are accepted to have serious implications for production control. Production systems must be able to typify flexible and adaptable behaviour and agent-based modelling is sufficiently robust to cope with the complexity of these systems. Pham et al. (2006) concur and strenuously argue that multi-agent approaches will continue to advance the state-of-the-art in applications of operations planning and control and supply chain systems.

Leitão (2009) argues that contrary to the inherent distributed nature of production control, practical applications traditionally rely on centralised hierarchical frameworks which fail to exhibit the capabilities outlined above. He insists that MAS provide functionalities able to support the parallel execution of scheduling and control activities in complex distributed, dynamic and expandable production systems. Despite these remarks, he reports weak adoption of these systems by the industry. This observation is consistent with the findings of Jahangirian et al. (2010) who list ABS as the fourth most popular simulation technique with only 5% adoption rate in their research sample. Attempting to shed more light on this intriguing finding, Bussmann et al. (2004) attribute the narrow adoption of agent architectures to the limited training of production control engineers in designing and implementing such methodologies and architectures.

This section presented how DES has driven developments in the field of MABS. By exploring the origins of agents in DAI, the section proceeded to explain how the field of MAS seeks to encapsulate human practical reasoning skills and mental behaviour in intelligent software components.

Attention was drawn to features of MAS which collectively add to their superiority in supporting decision-making in perplexing highly dynamic decentralised systems. The section analysed the intricacies of systems where MABS were reported to be successfully applied and attempted to delve into how MAS deal with this increased complexity.

The focus of the discussion shifted to manufacturing and it was demonstrated that production control systems share all these characteristics. Drawing evidence from the relevant literature, the section presented MAS as a robust but relatively novel decision support tool. An interesting finding of this review was that despite the proclaimed superiority of MAS, their applications in production control to date are scarce. As a result a void was discovered in the literature that needs to be filled in by conducting further research into how production control systems can be modelled and studied using MABS. This section contributes to the thesis by critically appraising ABS and selecting it as the primary research method adopted in this thesis.

#### **4.4 Applications of simulation modelling in production planning and control**

With competitive advantage shifting towards mass-customisation, responsiveness and agility in the 1990's, it was clear that traditional centralised production control systems were too rigid to cope with these new challenges. Agent-oriented software was seen as the vehicle for distributing the control of intra-company processes and managing the interoperability of complex supply chain networks (Leitão, 2009). This opened new avenues for research into applications of MAS in manufacturing. This section initially reviews the recent literature concerned with the diffusion of agent-based modelling in the wider production planning and control context. The focus of the discussion gradually narrows onto job-shops. The assertion of central importance to this thesis is that pull control policies can be extended into job-shops. For this reason, the section seeks to explore whether agent-based modelling is among the enabling technologies for this extension and compare its suitability and performance against other forms of conventional simulation.

##### **4.4.1 Production planning and control using MAS**

In the wake of the upsurge of interest in software agents, initial research efforts focused on applications of agent-based modelling in small-scale production systems. Owing to their size, compactness and inherent flexibility, manufacturing cells proved to

be a suitable test-bed. Heragu et al. (2002) propose a conceptual MAS framework which models the machines and material handling system of a cellular production system. Their framework combines features of hierarchical and heterarchical control allowing both vertical and horizontal agent interactions. Agents perform their control functions using a combination of optimisation, learning and knowledge-based algorithms.

A MAS for the control of a robotic packing cell is presented by Fletcher et al. (2003). Their system integrates automatic identification technology allowing the cell to handle rush orders and machine breakdowns. The modelled cell is both responsive and reconfigurable as resources can be easily added/removed using the designed plug-and-play agent system. Their framework is implemented in JACK™, a JAVA-based development platform. Tang and Wong (2005) design an agent-based architecture to provide a prototype for production control in a flexible cell comprising two robots, a conveyor line and an Automated Storage and Retrieval System (AS/RS). Their MAS includes reactive agents implemented in the Java Agent Development (JADE) platform.

RFID and agent technology are combined by Wang and Lin (2009) who develop a model to simulate the planning and control functions of an automated manufacturing cell. The proposed architecture includes a performance analysis agent which reviews operations schedules it receives from the planning agent and monitors their execution using real-time data transmitted by RFID tagged WIP items. The platform used for the implementation of the proposed MAS is JAVA-based Aglets.

Another distinct research stream is concerned with the design of agent frameworks and their validation using case-studies. Lim and Zhang (2003) develop a MAS framework capable of integrating the planning and control functions of agile manufacturing systems. An optimisation agent is designed to assess generated schedules using multiple criteria. Their framework is implemented in Microsoft Visual Basic and tested on a simple case-study comprising six machines. An extension of their work is presented in Lim and Zhang (2004), where a similar framework implemented in JAVA is employed to evaluate alternative reconfigurations before attempting any physical shop-floor changes.

Object-oriented simulation is used by Van der Zee (2006) for the development of a framework which aims to capture the interdependencies of production control decisions. This includes three classes, namely jobs, flow items and agents with the latter performing the control functions. The framework is tested on a fictitious repair shop comprising one inspection and one repair stations. Even though the impact of

control policies such as MRP-push, JIT-pull and CONWIP on the scope and timing of control decisions is recognised, there is no clear reference to a specific control practice modelled in the proposed framework. A real-time distributed scheduling system is developed by Wang et al. (2008). They design a scheduling agent which collaborates with one real-time and several resource agents to update the generated schedules dynamically. The framework is implemented in JADE and tested on a system consisting of two work-cells.

Despite dealing primarily with applications of MAS in manufacturing planning and control, some studies tend to elaborate extensively on design issues concerning agent control architectures, coordination and communication protocols. Frayret et al. (2004) argue that advances in manufacturing have resulted in forms of interdependencies between control activities which cannot be captured by existing typologies. They propose a new classification for the coordination/control of agent-based manufacturing systems. This differentiates between programmed/non-programmed coordination depending on whether agent collaboration is pre-planned or adjusted during model execution. The classification further distinguishes between direct/indirect coordination based on agent mutual adjustment or the presence of a mediator agent. An agent-based architecture is developed by Zhang et al. (2007) to support the evaluation of different shop-floor configurations based on machine utilisation and on-time completion of jobs. Their study deals extensively with agent control and coordination issues. A GA is developed to handle agent interactions and achieve production control optimality. They also design a new implementation platform and internet interface for the proposed networked manufacturing system.

Production planning and control is considered in the context of research dealing with applications of agent-based modelling in supply chain systems. Labarthe (2007) develop a modelling framework for supply chain management. They investigate the optional integration of existing scheduling infrastructure such as the Advanced Planning System (APS). Their findings suggest that a full integration increases the volume and complexity of agent interactions. As an alternative they propose building some APS functions into the agents. The framework is implemented in the MAJORCA platform developed in the context of their research. A MAS designed to simulate the operation of a distributed APS within a supply chain is presented by Santa-Eulalia (2011). The framework is implemented in the experimental platform FEPP and used to test the effect of tactical planning and control policies in the case of a softwood supply chain.

The survey of the literature which deals with applications of MAS in production planning and control identified a number of review papers. Apart from examining the diffusion of MAS in manufacturing planning and control, these studies discuss barriers to the adoption of agent technology. According to Caridi and Cavalieri (2004), job-shop scheduling is the domain which attracts the greatest research interest. Despite this observation, they claim there is no significant progress in transferring research breakthroughs in agent technology to the industry. They justify this important finding by pointing to the high dependency of the specifications of agent-based architectures on the characteristics of scheduling systems. They argue in favour of the development of a systematic framework which will help classify MAS architectures according to their suitability for scheduling systems with similar shop-floor configurations and control policies.

Monostori et al. (2006) identify manifold reasons for the slow adoption of MAS by the manufacturing industry. They argue that despite recent advances in the field of Agent-Oriented Software Engineering (AOSE), development platforms are not robust for industrial applications. They appear confident however, that scheduling and control are areas where the industrial take-up of MAS is expected to increase in the short-term. Their views regarding the industrial strength of development tools and platforms are shared by Shen et al. (2006) who claim that despite the existence of standards for generic agent-based systems, there are no available standards driving the development of agent-based manufacturing systems. Lee and Kim (2008) point out that the potential of agent-based scheduling systems should be explored with the context of the broader supply chain.

The review presented in this section identified that apart from a few papers which consider small-scale applications in cellular systems, all remaining papers do not explicitly identify the type of application environment. This is despite the pervasive impact of the configuration and physical layout of manufacturing systems on the scheduling and control practices applicable in their environments. A possible explanation for such a simplification could be that modelling a specific type of production setting, e.g. a job-shop would add significant overhead to the MAS in terms of modelling and development complexity. Such a major limitation however, prohibits industrial applications of the proposed agent-based production planning and control systems. The diffusion of MAS in manufacturing cells is a noteworthy exception. Cellular systems can occupy significant shop-floor areas in production settings.

They are far more flexible and reconfigurable than rigid production lines. However, they are not able to cope with the high variety and low volume production accommodated by job-shops.

#### **4.4.2 Agent-based job-shop scheduling**

The novelty and robustness of agent-based modelling sparked great research interest in applications of software agents in the area of complex job-shop scheduling. With a new market-driven agenda demanding manufacturing adaptability and responsiveness, proposed agent-based models sought to instil these characteristics into job-shop scheduling systems.

An adaptable agent-based scheduling system is proposed by Cheeseman et al. (2005). The architecture is implemented in JADE and tested on a small operational cell. However, the system's re-configurability is not experimentally tested. Lou et al. (2010) design a MAS platform for adaptable virtual job-shops which can be re-configured by adding manufacturing resources in response to variable demand. This is achieved by an auction mechanism which allows task agents to select and engage machines to perform their tasks. A novel feature of this architecture is the task agent which coordinates the negotiation process and allows faster generation of schedules. The platform is implemented in Java and experimental testing produces feasible schedules.

Scheduling responsiveness, that is, the ability of the system to update its schedules in response to unexpected events is another aspect which features heavily in agent-based models developed for job-shops. Leitão and Restivo (2006) design a MAS architecture which combines features of heterarchical and hierarchical agent control. The former is used to enable good reaction to disturbances and the latter employed to optimise the quality of production control. The architecture is implemented in JADE and tested on a flexible manufacturing system. Even though responsiveness is highlighted as one of the strongest features of this model, the unexpected events included in the experimental conditions are not clearly outlined. There is further no mechanism for assessing the performance of the system.

Hybrid heterarchical/hierarchical control is also implemented in the agent-based job-shop scheduling system developed by Wang and Liu (2006). A negotiation mechanism allows agents to collaborate in order to handle disturbances such as the arrival of new jobs and machine breakdowns. The agent system is implemented in Java. A case-study involving a small job-shop is used to test the performance of the schedules in terms of flow-time and makespan. A MAS architecture for the integration of planning



and scheduling of job-shops is presented by Wong et al. (2006). The system employs a rescheduling method which evaluates alternative process routings using multiple criteria, for instance, the time to generate a new schedule. The architecture is implemented in JADE and experiments show it slightly outperforms other rescheduling systems reported in the literature.

The state-of-the-art in the area of responsive job-shops is concerned with the design of systems that allow scheduling disturbances to be handled in real-time. Such a MAS architecture is proposed by Zattar et al. (2010). Their MAS combines heterarchical control and an operation-based time-extended negotiation protocol which restricts the agent interaction time. It further allows machine operations to be grouped to minimise set-up times. Lou et al. (2012) develop a MAS which integrates proactive and reactive job-shop scheduling. During the execution of the proactive schedule, machine agents announce their state thus allowing their counterparts to dynamically repair their schedules. The reactive schedules are improved from a global viewpoint by the scheduling management agent and the task management agent. The proposed MAS is tested on a small-scale case-study and shown to produce feasible schedules.

Agent-based modelling has been adopted in research attempts aiming at solving the job-shop scheduling optimisation problem. Liu et al. (2007) develop a MAS with an auction mechanism based on Lagrangian relaxation implemented across a rolling time horizon to globally optimise schedules. The architecture is designed for the scheduling of job-shops under various patterns of dynamic job arrivals. The architecture is implemented in Microsoft Visual C++ and a set of experiments shows that the model is both effective and stable. Guo and Zhang (2009) develop an agent-based architecture for intelligent manufacturing systems. A key component of this architecture is the job-shop scheduling optimisation algorithm designed to generate allocations of machine tools, workers and robots and minimise makespan. Limited information is presented regarding the experiments designed to test the feasibility and convergence speed of the algorithm.

Agent technology has been integrated with artificial intelligence optimisation techniques. Ennigrou and Ghédira (2008) create a MAS architecture to solve the deterministic FJSSP. Their architecture uses resource agents which are equipped with TS algorithms and collaborate to find a global optimum. A diversification technique is executed when the number of iterations by each agent exceeds a certain threshold. The architecture is implemented in ACTALK and testing on benchmark problems shows that the diversification technique produces comparable makespan results to other approaches reported in the literature. Renna (2010) proposes a pheromone-

based approach which is founded on the theory of Ant Colony Optimisation (ACO). Following the analogy of ant colonies which lay pheromone trails between their nests and food sources, the approach assumes parts flowing through manufacturing systems are ants that deposit pheromone to mark their throughput time. A MAS conceptual framework using this pheromone-based coordination approach is designed to model the scheduling of dynamic job-shops.

Given the central importance of dispatching in job-shop scheduling, it is not surprising that agent-based architectures are designed to either assess the efficacy of existing dispatching rules or introduce and test new ones. Walker et al. (2005) design a MAS for a general job-shop. A distinctive feature of their architecture is the evolutionary algorithm used to create new heuristics by combining six core scheduling rules. The algorithm is tested on a benchmark problem and found to result in good performance with regards to several scheduling measures. Reinforcement Learning (RL) is proposed as a better alternative to other agent negotiation mechanisms by Wang and Usher (2007). They propose a MAS where job agents use a RL algorithm to make routing decisions in a job-shop. Their model is built on Visual C++ and simulation experiments are carried out to test the performance of the algorithm.

The MAS designed by Rajabinasab and Mansour (2011) uses a pheromone coordination mechanism which ensures that whenever sequencing needs to be performed in a job-shop, the respective job agents submit proposals to the machine agent which evaluates them according to their calculated job pheromones. The architecture is implemented in JADE and experimental testing shows that the MAS sequencing approach outperforms five common dispatching rules across a range of operational efficiency metrics. A MAS for the JSSP is designed by Kouider and Bouzouia (2012). The system uses a supervisor agent which decomposes the JSSP into a set of interrelated sub-problems. These are subsequently assigned to resources agents which use behavioural plans to solve them and minimise schedule idle time. Their MAS is implemented in C++. Tests show that it produces better scheduling performance compared to eight different dispatching rules in scenarios involving static and dynamic job-shop conditions.

In line with recent trends in manufacturing, the majority of studies reviewed in this section deal with the design of robust scheduling systems for responsive and reconfigurable job-shops. Several studies seek to exploit the computational capabilities of intelligent agents to solve the NP-hard JSSP to optimality. Most of the studies also tend to focus on core scheduling functions e.g. job assignment and prioritisation whereas limited emphasis is placed on production control functions. In some of these

works, an agent-based approach was followed to develop new dispatching heuristics and compare their performance with widely used dispatching rules.

#### **4.4.3 Simulation modelling in applications of pull control in job-shops**

Whilst the literature is replete with successful implementations of pull control policies in flow-shop systems, product diversity and the complex logistics of diverse routings in job-shops avert their transferability into non-repetitive manufacturing. However, the recent literature heralds the emergence of small number of studies that seek to overturn this dogma. This section reviews studies which use conventional and agent-based simulation to carry out research in the core of this thesis, that is, the extension of pull control to job-shops. As these studies share some of the research objectives of this thesis, gaps in their approach, limitations and demerits will be discussed from a highly critical viewpoint.

##### **4.4.3.1 Use of conventional simulation**

Li (2005) argues that pull control is applicable to job-shops provided the latter are reformed into cellular systems. The creation of cells presumes the identification of products with similar process routings using GT. It is suggested that three more JIT practices need to be adopted in order to facilitate the transformation of job-shops into manufacturing cells. These practices include operations overlapping (one-piece flow), reduction of set-up/processing time variability and set-up time reduction. A model of a job-shop is developed in SIMSCRIPT. Job routing data is appropriately selected to maximise the number of created cells. However, such a simplification may not necessarily be possible in full-scale industrial applications.

An extension of this work is presented by Li (2010) who investigates the impact of cellular reconfiguration, CONWIP control, set-up time reduction and quality improvement on job-shop performance. Simulation models built in SIMSCRIPT are used to compare the performance of a job-shop under push control with that of the reconfigured system operating a CONWIP policy. In a similar vein to the original work presented in Li (2005), the shop-floor layout adaptation through the formation of cells is treated as an important precondition for the introduction of CONWIP control. This contention ignores previous studies (Luh et al., 2000; Ryan et al., 2000; Ryan and Choobineh, 2003) exemplifying the direct introduction of CONWIP control to jobs-shops without attempting any form of shop-floor reconfiguration.

Kesen and Baykoç (2007) seek to introduce a dual-card Kanban system in a job-shop setting and design a new dispatching mechanism to maximise the utilisation of the Automated Guided Vehicles (AGV) used for material transportation. The mechanism prioritises AGV visits at starving stations. Despite acknowledging the limited number of reported applications of kanban control in job-shops, their study does not delve on the complexities of introducing pull control in functional layouts. Moreover, there is no evidence of how the pull logic of the dual-card kanban system was adapted to job-shops and conceptualised for implementation in the simulation developed in ARENA. Experiments involve a hypothetical jobs-shop processing five jobs on nine machines, but the ability of the model to cope with more realistic experimental parameters was not assessed.

A Hybrid Push/Pull (HPP) control mechanism for Dual Resource Constrained (DRC) systems is developed by Salum and Araz (2009). Capacity constraints in DRC systems are imposed by the limited number of machines and workers. Their approach integrates CONWIP control for the release of raw materials into the system with the proposed HPP which controls WIP within the system. Part transfers are handled by machine operators. As a result, the HPP does not employ cards thus avoiding some of the complexities of applying the kanban system to job-shops. The proposed system uses a when/where rule determining when to process or transport parts and the destination station. A simulation model is developed using Arena. The HPP is tested on a manufacturing cell and results show good performance when processing and transportation times are similar. Therefore, its hypothesised suitability for job-shops is not validated. The basic when/where rule is supplemented by several other rules determining the cyclic allocation of workers to machines and the handling of multiple transportation signals by workers. The application of all these rules in a complex job-shop would practically neutralise the alleged simplified logistics of implementing the HPP.

Despite previous studies in the areas discussed above, Diaz and Ardalan (2010) claim their research presents the first implementation of a dual-card kanban system in intermittent production. A key feature of their model built in ARENA is a priority system which allows station operators to use real-time information about customer waiting lines and assign the highest priority to the product with the greatest customer demand. However, their main assumptions regarding the operation of the kanban system are fundamentally erroneous. In line with the basic operating principles of kanban control (Liberopoulos and Dallery, 2000), in a dual-card system customer demand is initially communicated to the last workstation of a job's process sequence and from there it

propagates backwards only if there are processed parts available in the output buffer of the upstream station. The simultaneous transmission of demand to all workstations points more to a Base Stock control policy. Furthermore, there is no discussion of the implementation of the conceptualised model in Arena. Simulation tests were conducted on an experimental job-shop processing six jobs with diverse process routings on four machines. Both the demand and processing times for all products were assumed to be uniform. This oversimplification is contradictory to the unpredictable nature of demand and extreme product diversification characteristics of non-repetitive production systems.

Müller et al. (2012) introduce push-kanban control for job-shops. The system integrates centralised MRP-type planning functions with decentralised control. Decentralised WIP control establishes maximum WIP levels at each machine. When this maximum level is exceeded a bottleneck-oriented capacity control mechanism is activated. The kanbans used in the system are not visual signals but rather time kanbans representing slots in each machine's available capacity plan. A forward-linked kanban loop component ensures that before a job is released for processing to a successive machine, the latter is checked for available free kanbans. A model is developed using the Techomatix Plant Simulation software package. The review of their work identifies several issues. The study provides limited insight into the algorithm used to set the local WIP limits. It is further not clearly explained how the system deals with capacity violations. Müller et al. admit that the bottleneck-oriented capacity control component is not accurately modelled in their simulation. Even though their model is highly detailed, it fails to demonstrate how it handles the large volume of interactions and negotiations resulting from the decentralisation of control and operation of time kanbans. Finally, it is argued that compared to the use of the POLCA system in job-shops, push-kanban copes better with disturbances yet it is not clear how the latter were considered in experiments.

A decentralised and isoarchic control architecture for pull-controlled job-shops with small series is proposed by Ounnar and Pujo (2012). Job-shops producing in small series are a special case of the general non-repetitive job-shop since they allow batch processing of products which are released in the system in a repetitive fashion. The proposed architecture called PROSIS is based on the notion of holons initially conceived by Koestler (1967). Holons have close similarities to agents as they have decisional capabilities and intelligence. PROSIS decentralises production control into three types of holons, namely, product, resource and order holons. The architecture is isoarchic as all holons are equipped with the same decisional capacity. It is accepted

that pull control used in repetitive systems is not directly applicable to job-shops. As a result, PROSIS seeks to create an artificial pull flow effect. Multi-criteria analysis is used to identify and prioritise products close to completion. Virtual production lines are then created to pull these products through the system and free-up manufacturing resources. In addition to creating an artificial type of pull control which has limited commonality with the pull control mechanisms proposed in the context of JIT, the proposed system is designed for a specific type of job-shop which is a simplified version of the general job-shop. Furthermore, instead of using a distributed agent-based simulation environment to implement and test this architecture, the model is built and tested in ARENA. A precursor of this study, which is subject to similar limitations, is presented in Pujo and Ounnar (2008).

Portioli-Staudacher and Tantardini (2012) present a study entitled “*a lean-based Order Review and Release (ORR) system for non-repetitive manufacturing*”. They admit the majority of non-repetitive systems are organised as job-shops. However, they proceed to suggest that the complexity of job-shops is forcing some non-repetitive manufacturing firms to streamline their processes and identify dominant flows or operate virtual cells. As a result, the focus of their work deviates from the conventional perception that regards non-repetitive production systems as job-shops. Their work seeks to extend the ORR approach which is primarily suited for job-shops into flow-shops. They develop a variation of the ORR policy which is tested using simulation in an experimental non-repetitive flow-shop that seeks to achieve the lean goals of WIP minimisation, throughput time reduction and workload levelling through balanced flows.

Harrod and Kanet (2012) compare the performance of a job-shop under four production control policies, namely no control, Kanban, CONWIP and POLCA. An intriguing point in their study concerns the analysis of how Kanban control is implemented in an experimental job-shop consisting of five machines processing jobs with diverse routings. This fails to clearly explain how a job is pulled into the input buffer of a succeeding station. Jobs arriving at the system are assumed to be routed into the “ready buffer” at the system entry point instead of pulling a fully processed job from the output buffer of the last station in the corresponding job sequence and sequentially triggering production in upstream stages. This is a point of major differentiation from the operating principles of the kanban system. Their study is the first to report the so-called lockup phenomenon which causes machines to stop cold when other stations do not release the required kanban cards. It can be justifiably conjectured that the phenomenon results from the incorrect adaptation of kanban control to the job-shop considered.

The review of studies employing conventional simulation to extend pull control in job-shop settings suggests that these investigations are fraught with serious limitations and weaknesses. Those studies proposing the transformation of job-shops into cellular systems as a prerequisite for the adoption of pull control disregard reported applications of pull mechanisms, primarily COWNIP, in job-shops involving no prior layout adaptation. However, some of the purported successful applications of pull control in job-shops are based on flawed assumptions regarding the operation of kanbans and overall display an alarming unfamiliarity with kanban control. Several of these studies also fail to adequately demonstrate how the adaptation of pull control to job-shops was conceptualised for implementation in simulation models. In their attempt to overcome the complexities of the kanban system, a few studies design modified kanban systems relying on detailed sets of rules and protocols that effectively negate the alleged simplicity of the proposed alternatives. These would be extremely cumbersome to implement in any realistic job-shop setting. Finally, certain studies fail to adequately demonstrate how conventional simulation models were able to handle the complexities of decentralised control.

#### **4.4.3.2 Use of agent-based simulation**

Distributed agent-based simulation is recognised as a superior modelling approach compared to conventional discrete event simulation (Mönch, 2007). This is argued on the basis of its computational efficiency and overall ability to cope with the complexities of decentralising the production control of job-shops. A small research stream involving studies that employ agent-based simulation to introduce pull control in job-shops has emerged in the recent literature.

Wu and Weng (2005) propose a MAS for the scheduling of flexible job-shops. The proposed job-shop scheduling system adheres to the principles of JIT by seeking to achieve the objective of job earliness/tardiness minimisation. Their MAS differentiates between jobs with only one or more operations remaining. Dispatching heuristics corresponding to these two kinds of jobs are designed and the appropriate heuristic is employed whenever a job falling into these two categories enters the queue of a workstation. The proposed MAS is developed in C++ and tested on a five-machine job-shop. Apart from employing the JIT objective of weighted earliness/tardiness minimisation, the proposed scheduling system bears no further resemblance to a JIT pull-controlled system as it effectively operates a push control policy.

The sequencing method developed by Wu and Weng (2005) is employed in a MAS designed by Weng et al. (2008) for the control of flexible job-shops in MTO firms. The MAS developed in C++ integrates new rules for dynamically setting due-dates and a job release mechanism. The latter keeps new jobs in a pool and only releases them into the system when the workload, defined as the total remaining processing time for all the jobs in the shop-floor, drops below a preset limit. The proposed system adopts some of the principles of JIT production. Firstly, the dominant scheduling objective is the minimisation of weighted earliness/tardiness. Secondly, the objective of the job release mechanism is to a certain extent similar to that of CONWIP. More specifically, whilst CONWIP seeks to maintain a constant workload by releasing jobs into the system each time customer demand pulls completed jobs from the FGI, the proposed release mechanism pushes new jobs into the system as long as the total workload allowed in the shop is not exceeded.

A MAS for the introduction of CONWIP control in job-shops with stochastic job arrivals and processing times is designed by Papadopoulou and Mousavi (2007a). Several agent types are defined to emulate the logic of the CONWIP mechanism. When the simulation clock signals the due-date of a given job, its job manager agent attempts to pull a fully processed job from a central buffer controlled by the FGI management agent. If a fully processed job of this type exists in the FGI, this is pulled out of the system and a new job is pushed into the system for processing. The routing of this job through the system is handled by the respective job manager agent. The proposed MAS architecture is implemented in JACK™ and tested on a good approximation of a realistic job-shop setting. A similar architecture designed to simulate the operation of a lean job-shop under Base Stock control is presented in Papadopoulou and Mousavi (2007b). The model uses a workstation output buffer agent to represent the output buffer of each workstation in the system. A newly arrived job is held until its due-dates before it pulls a fully processed job from the FGI and simultaneously trigger the processing of replenishment inventory at all stations. The model is empirically validated.

Miyashita and Rajesh (2010) design a MAS for the production control of the wafer fabrication stage of a semiconductor manufacturing system. The wafer fabrication is a special case of a flexible job-shop where each job has its own process routing involving hundreds of process steps and is allowed to revisit the same machine. The proposed system, called Coordination for Avoiding Machine Starvation (CABS), aims to dynamically identify shifting bottlenecks and regulate flow to prevent machine starvation. Agents coordinate through the exchange of messages to perform the



dispatching of jobs. It is argued that CABS emulates the operation of JIT pull mechanisms as the agents utilise information from succeeding stations to prioritise tasks which contribute to the completion of jobs with the heaviest demand. CABS is built in the SPADES middleware and tested on a wafer fabrication testbed problem involving only two jobs. A source-agent releases jobs to the system by assigning them to the machine agent processing the first step in its sequence. In this manner, jobs are initially pushed into the system before they are subsequently pulled from downstream processes. Therefore, CABS emulates a hybrid push/pull control mechanism, not a pure pull mechanism like kanban.

This is also the case in the conceptual model for an agent-based kanban scheduling system proposed by Turner et al. (2012). The system is designed for rapid response production environments sharing common characteristics with job-shops. They claim the proposed system follows the principles of kanban-based pull control, yet they explain that work starts at a workstation when there is available capacity. The graphical representation of the conceptual model also suggests that customer demand pushes new jobs into the system, instead of pulling completed jobs from the last workstation thus triggering production sequentially in upstream stages. In addition to these serious violations of the kanban principles, the proposed conceptual model is not empirically validated.

A successful application of pure pull control in HVLV job-shops is reported in Papadopoulou, et al. (2007). Intelligent agent decision support is used to implement the kanban system originally designed for repetitive production lines (Liberopoulos and Dallery, 2000). Several agent types are defined to handle the complex mechanics of the kanban system in a job-shop retaining its original configuration. The notion of zero due-date WIP held at the output buffers of workstations and FGI is also instrumental for the introduction of pure pull control. Demand for new jobs is met using fully processed jobs from the FGI. This, in turn, pulls replenishment WIP sequentially from upstream stations. Their model built in JACK™ is extended in Papadopoulou and Mousavi (2008) which seeks to study the performance of kanban control in a dynamic job-shop subject to disturbances caused by rush and cancelled orders as well as machine breakdowns. Simulation tests performed in a realistic job-shop setting show that pull control in job-shops improves flow-time and tardiness performance at the expense of higher WIP levels.

The review presented in this section counts only a small number of studies. This is indicative of the fact that research in the use of agent-based simulation in pull-controlled job-shops is still in its infancy. Certain studies adopt a rather myopic view in

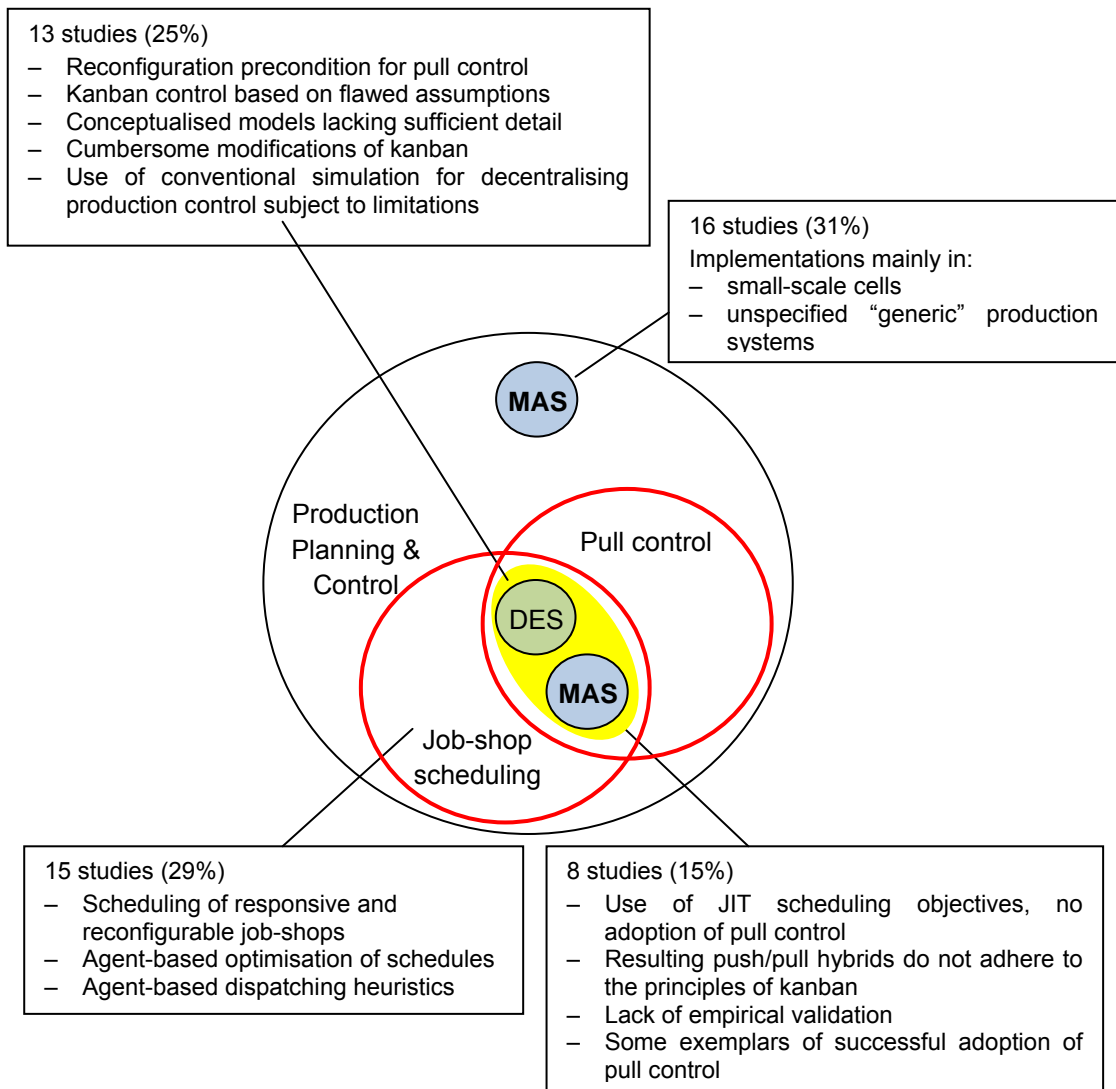
that they solely associate the lean transformation with the selected adoption of JIT scheduling objectives, primarily that of earliness/tardiness minimisation. These studies do not attempt to replace push control with JIT-inspired pull policies. Other studies seeking to allegedly apply kanban-based pull control result in the creation of hybrid push/pull systems which deviate from the principles of pure kanban control. From the two studies in this category, one proposes a conceptual model which is not empirically validated. The most salient developments in extending pull control in job-shops are presented in research carried out by Papadopoulou and Mousavi (2007a,b; 2008). Their attempts to directly apply the mechanics of pull control policies designed for repetitive systems in job-shops are both fruitful and productive justifying further research and empirical work in this area.

Figure 4.2 summarises the similarities shared by studies using intelligent agents to overcome the difficulties of extending pull control in non-repetitive production settings. The figure provides an overview of the surveyed sub-areas, the distribution of the 52 reviewed papers and main trends in each of these. It shows where the research area of main interest to this thesis (yellow-shaded oval) is placed within the wider context of applications of simulation modelling in production planning and control.

#### **4.5 Chapter summary**

This chapter uses a deductive approach to arrive at the selection of the most appropriate primary research method that can be employed to answer the questions posed by this thesis. The chapter begins by exploring the purpose and role of modelling in OR. The adopted deductive approach proceeds through the following steps:

1. Descriptive and prescriptive modelling approaches are reviewed. As optimisation is not within the scope of this thesis, it is concluded that a prescriptive modelling approach is not relevant. The study of the operation of job-shops under pull production control and the assessment of a possible performance differential requires a descriptive modelling approach.
2. With simulation being the most prevalent descriptive modelling approach, the focus of the discussion shifts towards the available computer simulation techniques. A brief overview of static (Monte Carlo), continuous (SD) and discrete-event techniques is presented. Since DES is founded on the notion of event-triggered changes of system status, it is deemed most suitable for modelling the intermittent production accommodated in job-shops.



**Figure 4.2 Schematic of papers reporting applications of simulation modelling in the wider context of production planning and control**

3. The review delves into the two main exemplars of DES, namely conventional and agent-based simulation. The superiority of agent-based simulation is argued on the basis of its unique ability to handle the intricate characteristics of job-shops as well as the volatile and decentralised nature of production control in their context. This thesis posits that the centralised MRP-push control logic traditionally applied in job-shops can be substituted by the decentralised JIT-pull control designed for production lines. The complex logistics of job-shop production caused by job and process routing diversity amplify the complexity of this extension. Agent-based is the only type of simulation able to cope with the resulting volume of interactions and high level of coordination.

Therefore, the chapter asserts that the adoption of a relatively novel, yet powerful approach, in particular one relying on the use of MAS, can represent a real breakthrough in the extension of pull control in non-repetitive production systems.

The chapter further sets to explore the state-of-the-art in applications of simulation in the field of production planning and control. The initial review of MAS in general production control systems serves as a precursor for the more specialised implementations of MAS in pull controlled job-shops. It exemplifies trends concerning all production systems, for instance, the integration of MAS with real-time data collection systems. It did however produce two unexpected findings. It revealed a tendency in relevant studies to model “generic” production environments, despite the close association that exists between the type of production setting and the nature of control functions performed in its context. It further drew attention to the limited industrial applications of agent-based production planning and control owing to the lack of robust MAS development platforms. It must be noted that most reported academic applications rely on the use of free development environments (with any limitations these may have), whilst only a few use commercial off-the-shelf software.

The review of studies using MAS for job-shop scheduling identified that major emphasis is placed on designing adaptive and flexible scheduling systems. These objectives have central importance in JIT production. Despite the integrated nature of scheduling and control, these studies put production control in the sidelines of their research. The survey of the literature continued with applications of conventional and agent-based simulation in pull controlled job-shops. Research in this area was found to be limited, with the number of studies employing conventional simulation exceeding those using MAS. As this is the area of the utmost relevance to the thesis, the survey of the related literature adopted a critical stance. Overall, both streams of research are subject to significant limitations concerning both their rationale and primary research approach. Their most serious demerits are outlined below:

1. Research attempts to introduce pull control in job-shops, involve either new push/pull hybrids or modified versions of the original pure pull kanban-based system. However, these newly proposed systems are thwarted due to their fundamentally flawed assumptions regarding the basic operating principles of pull production control.
2. The transition to a JIT scheduling system is not always complete. This is particularly true in the case of studies claiming to have designed a JIT scheduling system simply because they have embraced some elements of JIT, e.g. the JIT-inspired scheduling objective of earliness/tardiness minimisation.

This observation is consistent with the findings of Chapter 2 which broadly reviewed lean and JIT implementation issues.

3. Shop-floor adaptation is emphatically argued as a prerequisite for the introduction of pull control in job-shops. Studies supporting this view do not assess the cost implications of such reconfigurations nor consider the possible loss of exclusive advantages of functional layouts e.g. the ability to accommodate high variety low volume production.
4. Decentralised pull production control is not always modelled robustly in the adopted approaches. This holds true particularly in studies seeking to model the interactions involved in decentralised control using conventional simulation.
5. Vague conceptual models, lack of empirical validation, small-scale or unrealistic experimental settings are common occurrences in several studies investigating the introduction of pull control in job-shops.

This thesis seeks to address these limitations by proposing a direct yet complete implementation of JIT scheduling and control in job-shops. The job-shop scheduling framework for this transition is developed in chapter 3. This will be integrated with pull control mechanisms discussed in the same chapter. Most importantly the pull control mechanisms originally designed for repetitive production lines will not be modified in any way. The extension of pull control will be investigated in job-shops which retain their original functional layouts. The role of a MAS-based approach in overcoming the associated complexities and addressing the limitations of existing research in this area is instrumental in this research.

Section 4.1.2 of this chapter outlines the main stages of the simulation process. The problem formulation and model conceptualisation stages have been completed using input from chapters 1, 2 and 3. The remaining stages of the simulation, namely modelling of the proposed MAS for pull-controlled job-shops, simulation and verification are presented in chapter 5. Further to the experiments carried out in chapter 6, the proposed MAS is applied and tested on an industrial case-study.

Concluding, this chapter appraises the suitability of agent-based simulation for producing answers to the research questions framed in this thesis. By reviewing the existing, yet limited research carried out in the application of pull control in job-shops and pointing out the gaps and weaknesses of existing studies, it reinforces the justification for this research.

## **5 Lean scheduling and control of non-repetitive production systems using intelligent agent decision support**

This chapter presents the primary research undertaken in the context of this thesis. The key tool employed to carry out this primary research is agent-based simulation. The superiority of MAS in dealing with complex, distributed and dynamic problem scenarios was discussed in Chapter 4.

Chapter 5 initially seeks to develop the conceptual job-shop scheduling framework presented in Chapter 3 into a fully specified model that can be subsequently implemented as a MAS. Designing a job-shop system which can operate under various pull control mechanisms including Kanban, Base Stock and CONWIP presents a major challenge. Due to the different operating principles of push and pull control, the job-shop requires some special enabling infrastructure to operate under pull control. However, this thesis argues that a major shop-floor layout reconfiguration or adaptation of the original pull control logic is not a necessary prerequisite for the application of pull control in job-shops.

This chapter explains the implementation of the designed job-shop scheduling model as a MAS. Section 5.1 provides a brief overview of the characteristics of the modelled non-repetitive production system. The infrastructure and operation of the job-shop under consideration when push and pull control is applied is discussed in section 5.2. This further identifies the practical implications resulting from the adoption of pull control and suggests ways in which these can be overcome using agent-based simulation. Section 5.3 presents the full configuration of the designed system, including the simulation's key input parameters and output data. The fully specified job-shop scheduling model is implemented as a MAS in section 5.4 which details the key functions and interactions of its various agent types. An overview of the development platform used to implement the model is presented in section 5.5. Section 5.6 presents small-scale test problem scenarios used to verify the agent-based job-shop scheduling system. Finally, the main findings of this chapter are reviewed in section 5.7.

### **5.1 Scheduling system overview**

The main assertion of this thesis is that pull mechanisms designed to control the flow of products through mass production lines are transferable to non-serial job-shops. This section presents an overview of the infrastructure and operation of the job-shop scheduling system which is used as a test-bed for the introduction of JIT pull control. Particular emphasis is placed on how the pull control logic is applied to the non-

repetitive production system under consideration. The job-shop uses a fictitious functional layout and is a typical representation of a manufacturing system able to accommodate:

- Non-repetitive production of a diverse range of products (jobs) each of which has its own process sequence (routing).
- HVLV fabrication of a large variety of products in small batches.
- MTO production where the mix of products to be manufactured in each production period is determined by confirmed orders placed by customers. Therefore, the system does not produce to stock finished products.

## **5.2 Job-shop infrastructure and operation**

The job-shop under consideration, is assumed to be flexible, allowing recirculation, that is, jobs flowing through the system can visit a workstation more than once until they are fully processed. Workstations comprise groups of parallel machines able to perform identical (or similar) processes. Each workstation has its own input storage buffer that feeds the machine(s). It stores jobs that queue in front of the machines to receive processing. Before the system is initialised, all input storage buffers are empty. The job-shop is designed to operate under three different control modes, namely push, pull and hybrid push/pull depending on how it responds to demand and the mechanisms applied to control the flow of parts (WIP) through the system.

### **5.2.1 Job-shop operation under push production control**

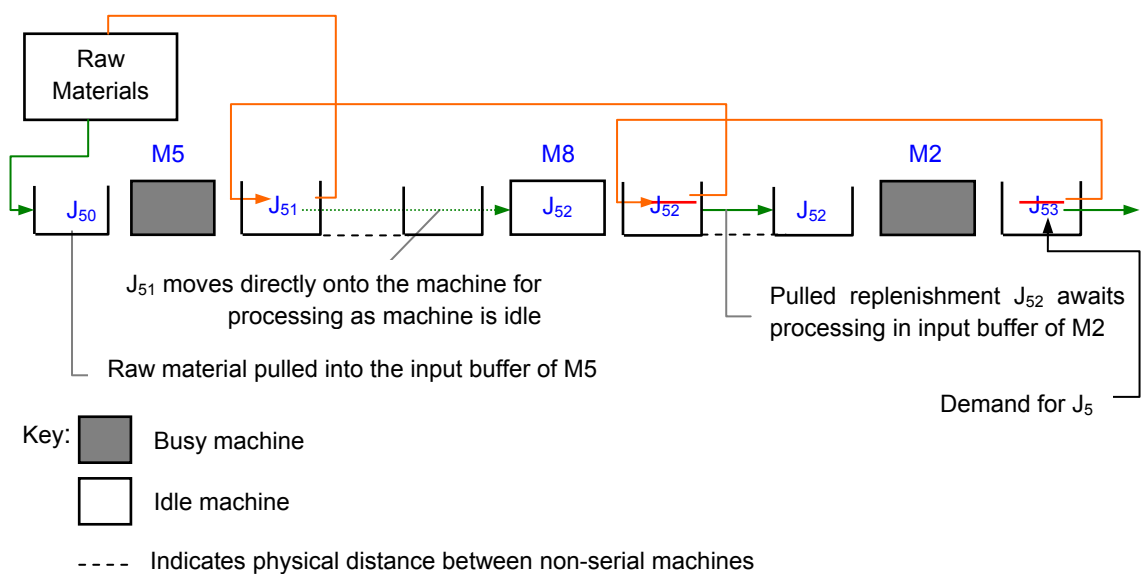
When the designed production system operates in push control mode, a job-list generated by an MRP-type planning system “pushes” jobs through the job-shop. The job-list specifies the type and quantity of orders that need to be manufactured. Jobs are released into the system immediately and production is triggered in the first workstation in line with their respective process sequence. If there is no available capacity in that workstation (machines are busy processing other jobs) the job joins the queue of waiting operations in the workstation’s input buffer. Jobs completing their processing at one workstation are “pushed” into the input buffer of the next workstation as specified in their process sequence without any consideration of that workstation’s workload.

### **5.2.2 Job-shop operation using pull production control mechanisms**

When the job-shop operates under pull control, in addition to an input buffer, each workstation has an output buffer. Before the system begins its operation, all output

buffers are assumed to be filled with a certain level of processed parts (intermediate WIP or FGI depending on the operation performed by the workstation in question). Instead of releasing an order into the system for processing as soon as this has been received (as it is the case under push control), the system attempts to fulfil it right at its due-date by “pulling” finished parts from the output buffer of the last workstation in the job’s process sequence. As soon as FGI is pulled out of the system, internal demand is generated for its replenishment. In order to meet this demand, the workstations upstream need to release intermediate WIP stored at their output buffers into the input buffers of the stations downstream.

Figure 5.1 illustrates this pull control logic in a simple example based on one job, for instance  $J_5$ . Its process routing specifies that in order to produce a unit of  $J_5$ , three processing steps need to be completed. The first processing step (operation  $J_{51}$ ) is performed on machine M5, the second processing step  $J_{52}$  on machine M8 and the third processing step  $J_{53}$  on machine M2. The three machines belong to non-serial workstations. Processed parts are stored in the output buffers of each workstation. Demand for  $J_5$  pulls a fully processed  $J_5$  job (a completed  $J_{53}$  task) from the output buffer of M2. In order to replenish this, the same quantity of  $J_{52}$  needs to be released from the output buffer of M8 into the input buffer of M2 for processing. This will then also need to be replenished and therefore, the same quantity of  $J_{51}$  is released from the output buffer of M5 into the input buffer of M8 where its processing begins immediately as there are no queuing jobs. The WIP held for  $J_{51}$  in the output buffer of M5 is replenished by having raw materials (denoted as  $J_{50}$ ) released from the raw materials storage into the input buffer of M5.



**Figure 5.1 Operation of the Kanban-controlled job-shop**



The main difference between the Kanban and Base Stock pull control mechanisms is the manner in which demand for parts is communicated to upstream workstations. In the case of Kanban control, demand is communicated to upstream workstations sequentially. Referring to the example presented in Figure 5.1,  $J_{51}$  parts would be requested for transfer into the input buffer of M8 only if  $J_{52}$  parts were available in the output buffer of the latter for release into the input buffer of M2. By contrast, the Base Stock control mechanism ensures demand for parts is communicated simultaneously to all upstream stations regardless of the availability of inventory in the output buffers of stations upstream. Practically, this means that the Base Stock mechanism replenishes intermediate levels of WIP more rapidly than is the case with Kanban.

The designed job-shop is effectively an inventory replenishment system. It fulfils orders using its available inventory and instigates production only to replenish this to pre-set levels. The job-shop is designed to operate in line with the overarching principles of Kanban and Base Stock control presented by Liberopoulos and Dallery (2000) for serial flow-shop systems. However, the underlying pull logic is not straightforward for the following reasons:

- Production, in fact in this context inventory replenishment, relies on continuous information exchange between the output buffers in the system. More specifically, each output buffer needs to handle requests for inventory sent by the output buffers of downstream stations. It further needs to issue its own requests for WIP to the output buffers of upstream stations.
- The level of inventory in the output buffers needs to be dynamically updated each time WIP is released into the input buffer of a requesting station or WIP produced by any of the machines in the same workstation is processed and ready to be temporarily stored.
- Parts flow through the system in the same direction as in the case of push control, but demand information for replenishment parts propagates backwards, from the last production stage to all preceding stages.

In the case of a job-shop processing a range of diverse jobs on a large number of machines, the volume of workstation interactions and distinct part flows through the system increase exponentially. Another major challenge is presented by the inventory held in the system. Krishnamurthy and Suri (2009) stress that for pull control to function in any non-repetitive production system, the latter must maintain a minimum level of inventory for each product at every station. They argue that whilst this is infeasible in one-of-kind production systems, it is possible in job-shops manufacturing a high variety of products.

**This thesis is founded on this premise.** It posits that pull control can be applied to job-shops provided there is an appropriate scheduling infrastructure to handle the interactions between workstations, manage the inventory maintained at various points in the system and coordinate the flow of replenishment parts. This essential infrastructure is provided in the form of the multi-agent scheduling system presented in this chapter.

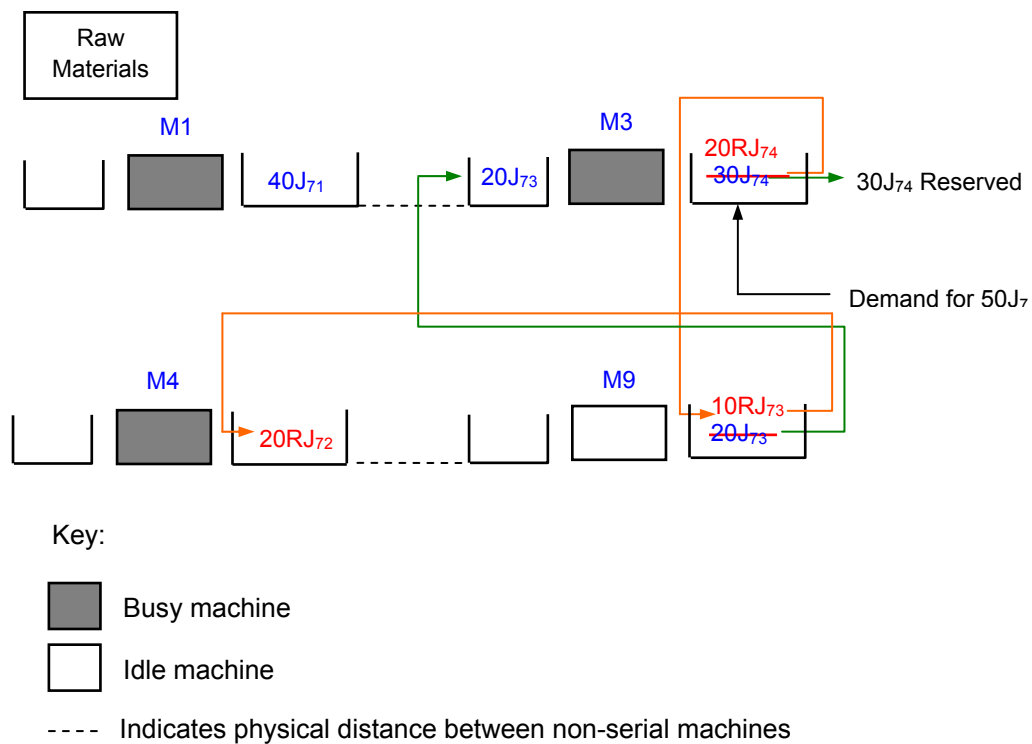
There are however, implications resulting from the application of pull control to job-shops which have not been considered in theoretical models of pull control proposed for repetitive production lines. These concern the system's response to situations whereby demand for the release of FGI from the last workstation in a job's process sequence or intermediate WIP from any upstream station can only be met in part or not met at all. In line with the operating principles of Kanban and Base stock, if demand for FGI cannot be met at all, FGI is not pulled out of the system and consequently, the replenishment process cannot be instigated in any of the upstream stations. However, considering the system starts its operation with workstation output storage points filled with a certain level of FGI, this situation is unlikely to arise at system initialisation. The situation can occur during the operation of the system whilst inventory replenishment is underway, but this means that FGI will be replenished at some point. The system may then need to respond to demand for FGI and WIP which exceeds its available stock levels. Restricting the system to only fulfil orders in full would have a detrimental impact on its performance and could eventually result in deadlocks.

This thesis seeks to address practical implications resulting from the application of pull control in job-shops by introducing a "batch-splitting" function into the designed scheduling system. The batch-splitting function allows any output buffer in the system to fulfil requests for inventory (relating to finished products or intermediate WIP) in part and log a request for the unfilled portion to be replenished when inventory becomes available by subsequently releasing it into the input buffer of the requesting station. In order to facilitate this behaviour, the system is designed as follows:

1. When the system operates under pull control, all intermediate and finished inventories residing in the system (queuing in input buffers, being processed on machines or temporarily stored in output buffers) are flagged as replenishment tasks and as such have no associated due-dates. They are not associated with orders placed by customers. This is in contrast to the system's operation under push control, whereby all jobs that flow through the system correspond to customer orders and have specified due-dates.

2. Each output buffer keeps track of its WIP (or FGI) level and updates this dynamically.
3. In addition, every output buffer maintains a list of unfulfilled requests for inventory which it attempts to fulfil as soon as inventory becomes available.
4. Fulfilled portions of orders are simply reserved by the system. Only complete orders are released from the system when all split portions have been fulfilled.

The above batch-splitting function is illustrated in the example presented in Figure 5.2. The example assumes a Kanban-controlled job-shop comprising four workstations. At time  $t$ , the system needs to fulfil an order for 50 units of  $J_7$ . Machine  $M3$  performs the final operation ( $J_{74}$ ) for  $J_7$ . Its output buffer holds 30 units of finished  $J_7$  products (completed  $J_{74}$  operations).

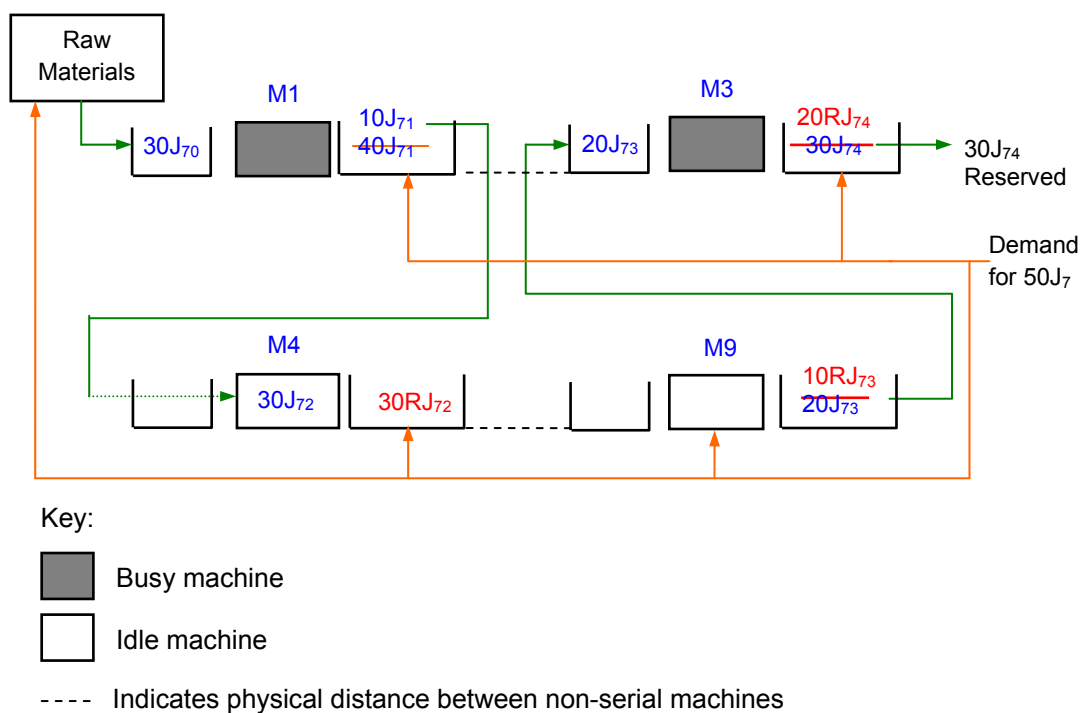


**Figure 5.2 Batch-splitting in the Kanban-controlled job-shop**

As the full order cannot be fulfilled, the requested batch of 50 units is split into two parts. The available 30 units of  $J_7$  are reserved and the output buffer logs a request for 20 more units. The 30 units of inventory pulled from the output buffer of  $M3$  need to be replenished. Therefore, 30 units of  $J_{73}$  need to be released from the output buffer of the preceding station, in this case  $M9$ , into the input buffer of  $M3$ . The 20 units of  $J_{73}$  available in the output buffer of  $M9$  are released and a request for 10 units is logged. The inventory consumed needs to be replenished and therefore, 20 units of  $J_{72}$  need to

be released from the output buffer of M4. Since this output buffer currently holds no inventory the demand for replenishment tasks propagates no further.

By contrast, assuming the job-shop operates under Base Stock control as illustrated in Figure 5.3, the consumption of 30 units of  $J_{74}$  from the output buffer of M3 instigates the replenishment process simultaneously at all preceding stations. For M4 this means that a request is logged for 30 units of  $J_{72}$  as there is no inventory available. The output buffer of M1 is able to release 30 units of  $J_{71}$  into the input buffer of M4. Furthermore, raw materials are released from raw material storage to replenish the 30 units of  $J_{71}$  released from the output buffer of M1 into the input buffer of M4.



**Figure 5.3 Batch-splitting in the job-shop controlled by the Base Stock mechanism**

The job-shop can operate under hybrid push/pull control as dictated by the CONWIP mechanism. The system's infrastructure is similar to that of the push system. There are no output buffers in workstations. However, there is central storage point for all FGI. Orders received by the system are fulfilled using the available FGI. In case of insufficient inventory, orders can be split into fulfilled and unfulfilled portions as appropriate. The fulfilled portion of the order will reserve FGI from the available stock. A new job of the same type and batch size will be released into the system for processing. The job will be pushed through the system as it would be in the case of pure push control. The fully processed job will replenish the consumed FGI. The FGI buffer will log a request for inventory corresponding to the unfulfilled portion of the

order. Once all portions of the original order have been fulfilled, this is released from the system. The above practically means that whenever demand pulls FGI out of the system, jobs of the same type and quantity are released and pushed through the system to replenish the FGI to its pre-set level.

### **5.2.3 Job-shop's reaction to unexpected changes**

Regardless of the selected operation mode (push, Kanban, Base Stock or CONWIP) the system is designed to respond robustly to disturbances. Internal disturbances arise in the form of machine breakdowns which reduce the system's machining capacity in the short-term. Machines can develop faults during their operation or during set-up time. When a fault develops whilst the machine is being set-up, the system assumes the setting-up process needs to be re-started once the fault has cleared.

Breakdowns affecting machines which are busy processing parts can cause either some or no impact on WIP. If there is no impact on the batch being processed, the machine will simply resume its operation at the end of the fault. If the fault damages a portion of the batch, the system is designed to react as follows. The unaffected portion of the batch (this includes both processed and unprocessed parts) will remain on the machine and its processing will resume at end of the fault. The scrapped portion will be replaced as set out below:

- If the system operates under push control, a new “replacement” job with the same batch size and due-date as the scrapped portion will be released into the system for processing.
- Under Kanban and Base Stock control, scrapped portions of replenishment jobs will be replaced by simultaneously placing requests for the same type and quantity of inventory at the output buffers of all workstations upstream of the machine where the fault developed. The same type and quantity of raw materials will also be released from the raw materials storage.
- When the system is controlled by the CONWIP mechanism, a new replacement job with the same batch size as the scrapped portion will be released into the system for processing. Unlike jobs flowing through the push system, under CONWIP control the replacement job does not correspond to an order. The replacement job replenishes the FGI maintained by the system and therefore has no due-date.

External unplanned events affecting the system concern the arrival of high-priority (rush) orders or cancellations of received orders. Cancellations of orders that have not

yet been released into the system for processing (system operation under push control) or orders which have not yet been fulfilled using available inventory (system operation under pull control) will simply be removed from the system's job-list. Cancellations will mainly affect jobs whose processing or replenishment (depending on the system's operation mode) is in progress when the cancellation occurs. In these cases, the system is designed to handle order cancellations as follows:

- Under push control, the cancellation of a job already released into the job-shop for processing will result in the job being removed from the system. The system will collect only certain performance information related to this job, e.g. impact on machine utilisation but overall the job will not be included in the set of completed jobs.
- Under Kanban, Base Stock and CONWIP control, the cancellation of an order which is either fulfilled in part or in full using the system's available FGI will result in the following two actions. Firstly, the reserved FGI will be returned to the respective storage point so that it will be available for future orders. Secondly, the replenishment process that was instigated by the system's attempt to restore its inventory to pre-set levels will be allowed to continue.

Rush orders are flagged so that they are always prioritised by the system. Under push control the system will release high-priority jobs into the system as soon as the respective orders are received. Similarly, when the system operates under pull control, it will attempt to fulfil rush orders using the available FGI immediately. All tasks associated with rush orders, including intermediate WIP, replenishment and replacement tasks of either full or split high-priority batches will carry the high-priority flag as they travel through the system.

The system is configured to use a range of dispatching rules (as discussed in section 5.3 below) to prioritise jobs queuing in the input buffers of workstations. A selected dispatching rule is applied globally, that is the same rule is used to assign jobs to all the machines in the system. However, before the dispatching rule is applied, the scheduling system checks for queuing high-priority tasks. If the queue includes tasks associated with high-priority orders, then these are selected and assigned to machines first. The dispatching rule is then applied to the remaining normal-priority jobs. This ensures that replenishment tasks associated with high-priority orders "travel" through the system faster and therefore, FGI replenishment and complete fulfilment of rush orders are accelerated.

### 5.3 Job-shop scheduling system configuration

The operation of the designed job-shop scheduling system depends heavily on its configuration. The system's configuration determines the following:

1. **The job-shop's infrastructure and machining capability.** The parameters that need to be defined here concern the number of machines in each workstation.
2. **Scheduling input data concerning the specifics of products fabricated in the system.** In this context, the terms "product", "job" and "order" are used interchangeably. Each customer order relates to a given product that needs to be manufactured in the job-shop. This is treated by the job-shop's scheduling system as a new job. Therefore, the first scheduling input that needs to be defined is a job-list, i.e. the set of products that needs to be manufactured. This is equivalent to the planned order releases typically generated by a PPC system for a given production period.

Each job involves a set of operations (tasks or processing steps) that need to be performed in a predefined sequence on certain machines. In addition to their routings, the following details need to be specified so that all jobs flowing through the system are fully defined:

- Arrival time (or ready time). This is the time the system receives the respective order.
- Set-up times. The time required to configure each machine that the job visits. No set-up is required when the operation previously performed by a machine corresponds to the same processing step of the same job type.
- Processing times of the entire job's processing steps on the respective machines in line with its process sequence.
- Batch (lot) size. Orders typically concern batches of identical products. Batch processing presumes no pre-emption, i.e. the complete batch must be fully processed before the machine can change over to another order.
- Due-date ( $d_i$ ). This is the promised delivery date of a fully processed job (finished product) to the customer. Due-dates can be specified explicitly or computed according to the Total Work Content (TWC) method (Jayamohan et al., 2000) as follows:

$$d_i = a_i + c \times TWC \quad (5.1)$$

Where:

$a_i$  =Arrival time of job  $i$

$c$ = Due-date tightness coefficient

TWC= the total set-up and processing time for the complete job (all operations)

- Release time ( $r_i$ ). This is the time at which jobs are released into the system for processing. Depending on the system's production control mode, the release time of a given job is set equal to its:
  - Arrival time under push control. A new job is pushed (released) into the system, as soon as the corresponding order is received.
  - Due-date under pull control. This follows the underlying principles of pull control. Customer demand pulls finished goods out of the system and simultaneously triggers production of new parts within the system.

Irrespective of the set production control mode, the release time of every high-priority job is set equal to its arrival time.

**3. The scheduling system's sequencing settings.** The system is able to select from a predefined list of dispatching rules. The selected dispatching rule is implemented throughout the system, i.e. the same prioritisation criterion is used to assign jobs to all the available machines. The system can be set to operate under any of the 17 dispatching rules below:

- CR, EDD, FCFS, LIPT (alternatively referred to as LPT), Longest Total Processing Time (LTPT, which is equivalent to the MWKR), MOPNR, MS, SST, SIPT (alternatively referred to as SPT) and WINQ. All these rules are defined in Appendix A.

In order to capture a variety of shop-floor conditions, the aforementioned are supplemented with the rules below.

- Modified Due-Date ( $MDD_i$ ). This rule is a variation of the EDD. It assigns priority to the operation belonging to the job which has the earliest MDD (Kanet and Li, 2004). The MDD is computed as follows:

$$MDD_i = \max[(t + TWCR_i), d_i] \quad (5.2)$$



where:

$t$  = Current time

$TWCR_i$  = Total work content remaining for job  $i$

$d_i$  = Due-date of job  $i$

- Least Number of Operations Remaining (LOPNR). The only difference to the MOPNR rule is that in this case, priority is assigned to the operation associated with the job that has the least number of operations remaining.
- Shortest Total Processing Time (STPT). The only difference to the LTPT is that, the STPT prioritises the operation with the shortest total processing time remaining.
- Highest Pull Frequency (HPF). This rule prioritises the job with the highest number of identical operations queuing in the input buffer of a given machine (Hum and Lee, 1998).
- Repetitive Lots (RL). The rule gives priority to the operation which is the same as the last operation assigned to a given machine (Flynn, 1987).
- FCFS/LATE. A composite rule which is only applicable when the system operates under pull control. It assigns priority to the operation for which there is a request for inventory in the output buffer of the same machine. If there is no such queuing operation in the input buffer of the machine, the rule defaults to FCFS (Framinan et al., 2000).
- SPT/LATE. This rule is similar to the FCFS/LATE however; the fallback rule is SPT and not FCFS (ibid).

**4. The scheduling system's production control mechanisms.** The job-shop can be set to operate under push and pull control. Three pull control mechanisms are applied, namely, Kanban, Base Stock and CONWIP.

**5. The job-shop's inventory levels.** Generally there are three types of inventory in the system. Raw materials, WIP in the form of operations queuing in the input buffers of machines and FGI inventory. A fourth type of inventory is maintained in the system when this operates under Kanban and Base Stock control. This relates to processed WIP temporarily stored in the output buffers of machines waiting to be "pulled" into the input buffers of succeeding machines. The system operates under the assumption of ample raw materials. Therefore, depending on the selected production control mode, inventory settings concern the following:

- FGI (operation under CONWIP control only). Pre-specified FGI levels for the range of products typically manufactured in the system are available at system initialisation.
- Inventory stored in the output buffers of machines (operation under Kanban or Base Stock control). Practically, this means that the system is initialised “filled” with intermediate WIP. As discussed in Section 5.2.2, this is imperative for the operation of the system under pull control.

**6. Dynamic conditions and unexpected disturbances affecting the normal operation of the job-shop.** The job-shop considered here can be set to operate under both static and dynamic conditions. Under static conditions, most scheduling input data is deterministic and explicitly pre-specified. Dynamic conditions concern job arrivals but also routings, batch sizes, set-up and processing times all of which can be randomly generated using the following range of stochastic distributions Erlang, Exponential, Gamma, Lognormal, Normal, Poisson, Secure Random, Triangular, Uniform, and Weibull (Simard, 2012). The system’s configuration sets the following machine breakdown parameters:

- The machine(s) developing faults. These can be either explicitly pre-set or randomly generated using any of the above stochastic distributions.
- The timing and duration of faults. These can also be pre-specified or randomly generated.
- The impact on parts processed by the machine(s) developing faults. A probability damage determinant is used to specify this. More specifically, by setting the value of the probability damage determinant to 0% the system assumes no impact on WIP.

The system’s configuration for unexpected events related to order cancellations and high-priority jobs includes the following settings:

- Rush orders. The job type, arrival time and all other relevant job data can be either pre-specified or randomly generated. However, all rush orders are assigned a zero due-date.
- Order cancellations. The job type and time of cancellation can be either pre-set or randomly generated.

**7. The termination conditions for the job-shop’s operation.** Unless a time is explicitly specified, the system’s operation terminates by default once all jobs have been completed.

**8. The system's scheduling output.** The system is set to collect output statistics used to assess its performance in terms of the following metrics:

- Number of tardy (late jobs). This is the count of all fully processed jobs which are released from the system for delivery to the customer later than their due-date.
- Total absolute deviation of earliness/tardiness (TADE/T). This measures the JIT performance of the generated schedule. It is determined using the following formula:

$$\text{TADE/T} = \sum_{i=1}^n |(d_i - c_i)|, \quad \forall \text{ job } i \text{ in the system} \quad (5.3)$$

Where:

$d_i$  = Due-date of job  $i$

$c_i$  = Completion time of job  $i$

- Average flow time (AFT). This measures the average time jobs spend in the system after they have been released into the system. It is computed using the formula below:

$$\text{AFT} = \frac{\sum_{i=1}^n (c_i - a_i)}{n}, \quad \forall \text{ job } i \text{ in the system} \quad (5.4)$$

Where:

$c_i$  = Completion time of job  $i$

$a_i$  = Arrival time of job  $i$

- Throughput (TP). This determines the total time jobs spend in the system after their processing starts. It is calculated using the formula below:

$$\text{TP} = \sum_{i=1}^n (c_i - s_i), \quad \forall \text{ job } i \text{ in the system} \quad (5.5)$$

Where:

$c_i$  = Completion time of job  $i$

$s_i$  = Start time of the first processing step (operation) of job  $i$ . When the system operates under pull control (including CONWIP), this is equivalent to the job's due-date.

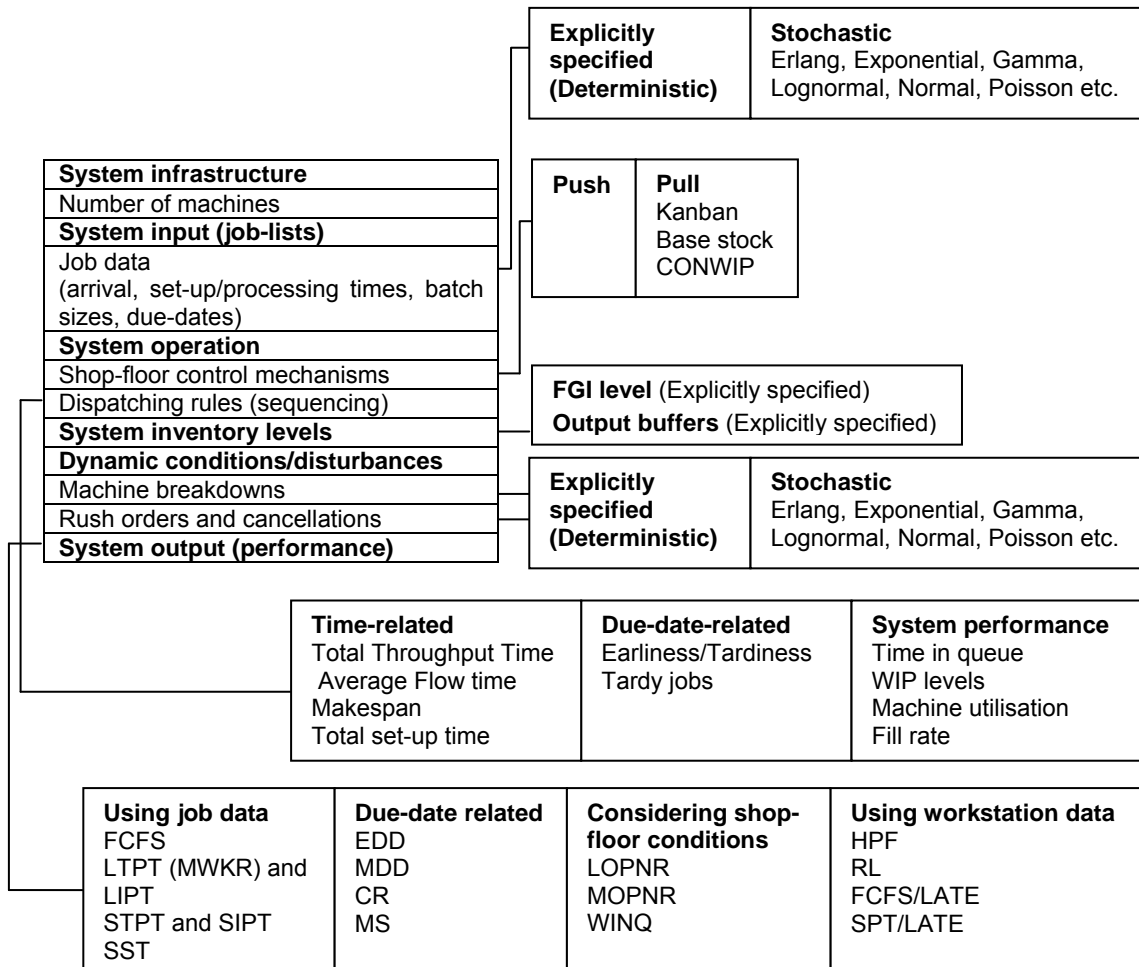
- Makespan. This measures the total time required to complete the full set of jobs. It is determined by subtracting the start time of the first job from the completion time of the last job processed in the system.
- Total time in queue. The total time all jobs spend queuing in the input buffers of machines they visit.
- WIP level. This is computed as the total time jobs spend either queuing in input buffers or being processed over the total time system operation time.
- Machine utilisation. This is computed by dividing the total time during which all the machines in the system are busy (processing) over the total available machining time (number of machines times the system operation time).
- Set-up time. This is computed as the total time required to set-up the machines until the system terminates.
- Fill rate (or service level). This measure computes the percentage of orders which are fulfilled immediately using FGI. The fill rate is always 0% when the system operates under push control.

Under pull control, the system's full operation is broken down into two distinct cycles. The first cycle covers the period during which the system fulfils orders using its available FGI. During the second phase the system produces to replenish its intermediate WIP and FGI to pre-set levels. Consequently, performance measures related to time in queue, machine utilisation, WIP are determined for both cycles. All the job-related performance measures outlined above are calculated for customer orders not replenishment jobs. Figure 5.4 provides an overview of the complete range of system configuration settings.

#### **5.4 Agent-based simulation model**

This section discusses the architecture of the multi-agent system developed to model the operation of the job-shop scheduling system described in the previous sections. In order to differentiate between the job-shop scheduling system and the agent-based simulation model, the latter is henceforth referred to as Brunel Agent Scheduling System (BASS). BASS is designed to achieve the following objectives:

1. Simulate the operation of the job-shop scheduling system.
2. Investigate the applicability of pull control mechanisms in job-shops which are typical representations of non-repetitive production systems.



**Figure 5.4 Overview of the push/pull-controlled job-shop's configuration**

3. Test and evaluate the job-shop's performance under push and pull control policies. Using the set of predefined performance criteria, identify and analyse any resulting performance differential.

11 different agent types are used in BASS. Their names and number of instances are identified below:

- System Manager Agent (SMA). One instance of this agent is spawned in BASS.
- Customer Agent (CA). One instance of this agent is spawned in BASS.
- Failure Manager Agent (FMA). One instance of this agent is spawned in BASS.
- Dispatcher Agent (DA). One instance of this agent is spawned in BASS.
- Machine Agent (MA). An instance is spawned for every machine in the job-shop. Machines are added/removed using the system's configuration interface.
- Job Manager Agent (JMA). An instance is spawned for every order the system receives.

- Workstation Supervisor Agent (WSA). An instance is spawned for every workstation in the job-shop. Workstations are created using the system's configuration interface.
- Workstation Input Buffer Agent (WIBA). An instance is spawned for each workstation in the system.
- Workstation Output Buffer Agent (WOBA). An instance spawned for each workstation in the system when the latter operates under Kanban and Base Stock control.
- FGI Manager Agent (FGIMA). One instance of this agent is spawned in BASS.
- Performance Monitor Agent (PMA). One instance of this agent is spawned in BASS.

#### **5.4.1 Overview of agents in BASS**

Agents in BASS represent entities within the job-shop e.g. orders and WIP or its infrastructural components, for instance, the available machines. The agents further emulate the behaviour of the component or entity they represent. In certain agents, these behavioural aspects are particularly intense. One such example is the JMA which handles the progression of jobs and flow of WIP through the system. In doing so, the JMA delivers most of the push and pull control functions of the scheduling system. One common feature shared by most agent types in BASS concerns their ability to identify other agents that exist in BASS and communicate with these. The detailed actions and interfaces of each agent type in BASS are discussed below.

##### **5.4.1.1 System Manager Agent (SMA)**

The SMA is responsible for managing the overall system, constructing all other agents in the system but not necessarily interacting with most of them directly. It performs the following functions:

1. It manages the configuration of the system. It imports agent names from a database which stores the names of agents with particular roles in BASS. It then reads and processes the configuration database to construct these agents.
2. It starts new jobs in the system by generating instances of the respective JMAs. New jobs are created from:
  - A pool of normal priority jobs (specified in the system's configuration file) which is stored in the SMA's "known\_jobs" database. Internal

notifications are posted within the SMA so that it can handle the initiation of a known set of jobs at their appropriate times.

- Requests by the CA for the initiation of high-priority jobs (rush orders). The SMA responds to such requests by placing rush orders into its “known\_jobs” database.

When the simulation clock signals the arrival time of a known job, the SMA constructs its JMA, stores the JMA’s details and instructs it to start the processing of the job.

If a job that needs to be initiated concerns a job marked for cancellation (following a notification by the CA), the SMA responds by logging its cancellation in the performance tracing log and removing it from its “known\_jobs” database.

3. It deals with job cancellations. Its response depends on the status of the job when the cancellation occurs. More specifically, if the cancellation concerns:
  - An imminent job which has not yet started its processing, practically a job which does not yet have a JMA constructed, the SMA responds by including this in its “completed\_jobs” database.
  - A completed, i.e. fully processed job. The SMA simply records in the performance tracing log that the job cannot be cancelled as it is already completed.
  - A job currently being processed in the system. The SMA places an appropriate timestamp in the performance tracing log to record the cancellation. It further informs all WSAs and the respective JMA that the job has been cancelled.
4. It is responsible for monitoring the state of the system and determining when the system can be terminated. The system terminates (finishes) under the following conditions:
  - All known jobs are fully processed.
  - A terminating condition specified in the system’s configuration arises e.g. certain system operation (run) time has elapsed. The SMA records the termination condition by logging a tracing message in the system’s performance log.

Irrespectively of the condition that forces the system to finish, the SMA informs all JMAs, WSAs, MAs and the FGIMA that the system is finished so that all remaining performance information can be collected. The SMA waits for 1 millisecond to elapse to allow other agents to respond to the system finished

notification. It then informs the PMA that the system is finished and waits for the latter to respond.

#### **5.4.1.2 Customer Agent (CA)**

BASS is able to cope with demand fluctuations taking the form of high-priority jobs (orders) arriving at the system unexpectedly or jobs that are no longer required by the customers and therefore need to be cancelled. In order to model rush orders and order cancellations the CA is introduced. The CA has a limited role which involves the following functions:

1. High-priority job initiation. Rush orders are initiated at the time specified in the system's configuration file. The CA marks them as "high-priority" specifying that they need to be immediately released into the system (by setting their due-date equal to their arrival time). It then sends an appropriate notification to the SMA requesting the immediate initiation of these jobs.
2. Job cancellation. The CA waits until the specified (in the system's configuration file) job cancellation time and then sends the cancellation notification to the SMA.

#### **5.4.1.3 Failure Manager Agent (FMA)**

Modelling machine breakdowns requires the introduction of a new agent type which informs machines in the system that they have developed faults. The FMA is designed to provide this capability. The FMA accesses its "fault\_set" database and obtains information about the next fault (fault specifics are determined in the system's configuration file). It then waits until the scheduled time of the fault and informs the appropriate MA that a fault has developed, simultaneously providing the latter with the following information:

- Machine downtime, i.e. length of time during which the machine will be non-operational due to the fault.
- Effect of fault on the job in-progress, that is, whether or not damage has been caused.

#### **5.4.1.4 Dispatcher Agent (DA)**

The selection of the next job to be processed on any machine in the system from the list of all the jobs queuing in the machine's input buffer is performed by the DA. The DA



records the arrival of new tasks in the input buffers of machines by processing notifications sent by the WIBAs. It then waits until it is allowed to perform a task assignment<sup>1</sup>. It records the name of the workstation which is eligible (contains at least one idle machine) for task assignment. It also requests task information from the respective WIBA which it uses to implement the specified dispatching rule.

As soon as information about the queuing tasks is received, the DA performs its task selection routine as outlined below:

1. Initially, it attempts to assign tasks associated with high-priority jobs. If there are more than one high-priority tasks available these are prioritised on a FCFS basis. Once all high-priority tasks have been assigned it proceeds to prioritise normal-priority tasks.
2. It selects the specified dispatching rule (as set in the system's configuration file) from its "selectors" database. It then performs the rule's logical and mathematical functions using the task data provided by the WIBA. For instance, when the SIPT rule is selected, the DA compares the processing time for each task and selects the one with the shortest next operation time. However, if the selected rule is:
  - The WINQ, in addition to the obtained task data, the DA needs to collect information about the workstation's input buffer queues to make a task selection.
  - One of the combined LATE rules, namely FCFS/LATE or SPT/LATE, then provided the system is set to operate in pull mode (either Kanban or Base Stock)<sup>2</sup>, the DA needs to obtain information from the workstation's output buffer to make an appropriate task selection. The information concerns outstanding requests for inventory.

When the dispatching rule cannot distinguish between two or more tasks which all meet the rule's selection criteria e.g. case of three tasks having the same shortest set-up time when the SST rule is implemented, then two "fallback" mechanisms are implemented. Initially, the dispatching rule attempts to select tasks on a FCFS basis. If this first fallback criterion fails, the rule prioritises tasks in alphabetical order of their BASS string representation.

---

<sup>1</sup> This condition is relevant only when the system operates under pull control. It ensures the DA will only begin prioritising tasks after inventory has been released (pulled) from the workstation output buffers or the FGI storage in the case of Kanban/Base Stock and CONWIP control respectively.

<sup>2</sup> The FCFS/LATE and SPT/LATE are reduced to FCFS and SIPT when the system operates under push and CONWIP control (no output buffers exist in the system in these cases).

3. It stores the selected task and marks it as “ready for assignment”.
4. It records a note in the performance tracing log that a task was selected successfully.

The DA is able to handle requests for tasks to be assigned to specific machines. This scenario is particularly relevant when more than one machine from the same workstation have requested the assignment of a new task. If any of these machines have performed the same operation (processed the same step of the same job type) before requesting the assignment of a new task, the new task will be assigned to this machine. The machine prioritisation rule ensures that unnecessary set-up times are eliminated. The DA receives requests for the selection of specific machines from the WSAs and stores these into its pending assignment list for processing. The requests contain information about the type of set-up performed by each machine.

#### **5.4.1.5 Machine Agent (MA)**

The MA represents a machine in a workstation. Several instances of this agent are created to model all the machines in the system. As MAs are also responsible for the processing operations (tasks) of the jobs assigned to them, they emulate both an infrastructural and behavioural component of BASS.

Each MA is able to monitor and update its status. Under normal operating conditions and whilst in processing mode, the MA’s status is set to “busy”. The MA maintains its “busy” status for a time interval equivalent to the processing time of the particular task at hand (the overall processing time is appropriately adjusted in case of a job batch size exceeding one unit). The MA changes its status to “idle” and notifies the WSA about its updated status as soon as it has completed its processing.

The MA updates its status in response to unexpected system disturbances such as job cancellations and machine breakdowns. When a notification is sent by the WSA regarding a job (order) cancellation, the MA reacts as follows:

- If the task in-progress belongs to the cancelled job and the latter represents an actual order, the MA changes its status to “cancelled” and then to “idle”. This practically means that the processing of the task is terminated. The MA then informs the WSA that it has reacted to the job cancellation.
- If the task at hand is an inventory replenishment task, its processing continues and the MA informs the WSA that it has reacted to the job cancellation appropriately.

In the event of a breakdown signalled by the FMA, the MA changes its status (either busy or idle) to “out-of-order” for the duration of the fault. At the end of this interval, the MA resets the machine to its initial status, i.e. the status prior to the development of the fault.

The MA responds to messages it receives from the WSA containing notifications that the system is finishing due to terminating conditions set in its configuration. The MA updates its status to “finished”, completes any processing in-progress and sends data regarding its total operation time to the PMA. It then terminates its operation.

The MA executes its job processing function by responding to messages sent by the DA instructing it to process assigned tasks. The MA initially waits until its status changes to “idle”. It then changes its status to “setting-up” for a time interval equivalent to the set-up time of the respective assigned task. At the end of this interval, the MA sets its status to “busy” for a time period equivalent to the task’s processing time. Once the processing of the task is completed, the MA notifies the respective JMA and changes its status to “idle”. However, in the event of interruptions whilst the machine is processing or being configured, the MA reacts as follows:

- In case of job cancellations or the system finishing (for the reasons discussed above), the MA collects the task’s performance information and informs the respective JMA that the task has been terminated.
- In case of a fault, the MA attempts to deal with the fault and its actions are detailed below.

The MA’s reaction to a fault depends on its status at the time when the breakdown occurs. If:

- The machine is idle i.e. the machine is not processing tasks when the fault occurs. The MA simply changes its status to “out-of-order” for the pre-determined fault duration and back to “idle” following the conclusion of the downtime period.
- The machine is being configured when the fault develops. The set-up operation needs to be performed again at the end of the fault.
- The machine is “busy” processing a batch when the fault occurs. In this case there are two possible scenarios depending on the effect of the fault:
  1. The fault does not cause any damage to the batch being processed. The MA changes its status to “out-of-order” for the fault duration. When the downtime period concludes, the MA changes its status to busy and resumes the processing of the batch.

2. The fault damages a portion of the batch. The batch is “split” and the scrapped portion (only) is replaced. The MA informs the respective JMA that a portion of the batch is scrapped, allowing the JMA to replace that component of the task. This is achieved by creating a new (replacement) job identical to the initial job (process sequence and due-date) and batch size equal to the scrapped portion. The MA resumes processing for the remaining items once the fault has cleared.

#### **5.4.1.6 Job Manager Agent (JMA)**

A JMA instance is created for every job processed by the system. The JMA’s role is to manage the job as it flows through the system by handling its initiation, progression and completion. The JMA is instrumental in delivering the system’s production control function. In order to achieve this, the JMA is designed to execute different job management protocols depending on the system’s production control mode.

When the system operates under push control, a JMA receives an instruction sent by the SMA regarding the initiation of a new job. The JMA responds to this instruction by storing the provided job details, that is, the job type, batch size, ready time, due-date and priority in its “current\_job” database. The JMA initiates the processing of the job, which under push control progresses through the system forwards (from the raw materials buffer to the first processing workstation, then the second, third and so forth). The JMA sends a task for the first step of the job to the appropriate WIBA. The JMA obtains the name of the buffer from its “next\_workstation” database and sends an appropriate notification to it so that the task can be added to its input buffer. The JMA will record the job’s start time, that is, the time when the processing of its first task begins. This will be used for the computation of performance metrics such as the throughput time once the system has completed its operation.

Once the processing of the task is completed by the machine, the respective MA will confirm this to the JMA which will record the completion of this job’s step in the system’s performance tracing log. It will then start the next step of the job by sending a new task to the input buffer of the workstation responsible for processing the second step of the job. The JMA will repeat the process outlined above until it receives confirmation from the machine processing the final step of the job that the latter is completed. The JMA will then record the completion of the job in the tracing log. It will store the job’s finishing performance information and finally inform the PMA and SMA that the job is finished.

The JMA's role is different when the system is set to operate under Kanban control. The JMA will respond to a request sent by the SMA to initiate a new job. It will store the job's details and treat this as its current job. The job's details comprise its type, ready time, due-date, batch size and priority. The JMA's actions so far are similar to those it would perform under push control. However, the JMA's job initiation routine is modified after this point as below:

1. The JMA will note that the job is to be progressed through the system backwards, i.e. that it needs to treat the job's last operation as its first step, the penultimate as the second processing step and so forth.
2. The JMA will not release the job into the system immediately but rather wait until its due-date and record this as the job's start time.
3. The JMA will begin processing the job's final step.

The JMA will proceed with the progression of the job through the system as follows. It will communicate with the output buffer of the relevant workstation (i.e. initially that processing the final step of the job) and request inventory to fulfil the order associated with this job. It will specify the details of the order (job type, batch size and priority) in the message it will send to the relevant WOBA. It will then wait for the WOBA's response. When it finally receives a response confirming the full or partial availability of inventory the JMA will perform these actions:

- It will process the received inventory. It will record that inventory was obtained for the final step of the job and that a portion of the order is fulfilled. It will then store the details of this portion in its "completed\_portions" database. The details of the fulfilled portion will be reset (the due-date of the original job is no longer relevant) and the portion will be marked for replenishment.
- It will begin the replenishment of the obtained inventory at the previous step of the job (workstation processing the penultimate operation of this job). It will communicate with the appropriate WOBA and request inventory, providing the latter with the job type, step, batch size (now equivalent to the portion used to partially fulfil the order) and priority. As soon as the respective WOBA sends its response, the JMA will begin processing the inventory supplied. Assuming that only part of the inventory requested can be supplied, the JMA will split this into the equivalent fulfilled and unfulfilled components. It will record that the job is being split. It will then send a message to the WIBA of the workstation downstream and request that the latter adds a new task for the component of the job that received inventory to its buffer. For the component of the job that

did not receive any inventory, the JMA will asynchronously repeat the process of requesting inventory from the relevant WOBA.

The above two steps are carried out each time the WOBA corresponding to a certain step of the job fulfils a request for replenishment inventory. In this manner, the overall replenishment process ripples backwards but sequentially. This means that only when inventory stored at the output buffer of a (supplying) station is consumed to replenish inventory consumed by a succeeding (requesting) workstation, the JMA is allowed to go to the output buffer of the workstation upstream and request the release of WIP into the supplying station's input buffer. When the JMA can use all its stored completed portions to fulfil the full order, it provides the final performance information for this job to the PMA and records that the order is fulfilled in the performance tracing file. If all the portions which flow through the system (including replenishment and replacement) and are associated with this job have been finished, the JMA informs the SMA that the job itself is finished.

Under Base Stock control, the JMA handles job initiation in a similar way to that used with Kanban control. The actions it takes to progress jobs flowing through the system are also consistent with those applying under Kanban control. Initially, the JMA attempts to fulfil an order by requesting inventory from the output buffer of the workstation processing the final step of the job. As soon as the availability of some inventory is confirmed, the JMA processes this by storing the details of the received portion in its "completed\_portions" database. The portion is marked for replenishment. From this point onwards, there are subtle differences to the way the JMA manages the progression of replenishment tasks through the system.

The JMA communicates with the output buffer agents of all the workstations which process step(s) of the job (except the last one which has already supplied inventory to fulfil the order) and submits requests for inventory matching the details (job type and batch size) of the fulfilled portion of the order. Replenishment of the first step of the fulfilled portion can proceed immediately by feeding the input buffer of the respective workstation with the appropriate raw materials. Whenever a WOBA responds to this request by making the requested inventory available either in full or in part, the JMA splits the inventory received into a fulfilled and an unfulfilled component. The fulfilled component is directed to the input buffer of the workstation downstream. The JMA continues to request inventory for the unfulfilled component. The JMA loops through the same process until there are no unfulfilled requests for inventory in any of the output buffers of workstations processing steps of the job. Similarly to Kanban, orders are fulfilled when completed portions can be combined to fulfil the full order. The JMA's

functions and interactions with the PMA and SMA following the fulfilment of the full order and completion of job respectively are the same as those used under Kanban.

Under both Kanban and Base Stock control, the JMA responds to messages it receives from MAs regarding the completion of processing steps. More specifically, every time a machine completes the processing of WIP assigned to it, the respective MA communicates with the JMA which then progresses the respective job by adding the WIP to the workstation's output buffer. The JMA further records the completion of the task and its temporary storage in the workstation's output buffer in the performance tracing log.

The JMA's behaviour under CONWIP control has similarities with aspects of its behaviour under both push and pull control. Job initiation is quite consistent with what is presented above for Kanban and Base Stock control. The JMA will handle a request sent by the SMA for the initiation of a new job, by treating this as its current job and storing the job's details (type, ready time, due-date, batch size and priority). The JMA will wait until the job's due-date (unless the job is cancelled in the meantime). The job's due-date will be stored as its start time. The JMA will then communicate with the FGIMA and request inventory to fulfil the order.

Assuming only partial availability of inventory, the JMA will split the order into a fulfilled and unfilled component. It will record that the order was split and note the quantity of the received FGI in the performance tracing log. The fulfilled portion will be stored in its "completed\_portions" database. An inventory request will be submitted to the FGIMA for the incomplete portion. The completed portion's performance information (mainly its due-date) will be reset and the portion itself will be flagged as a replenishment job. The JMA will then initiate the replenishment job by sending a task for its first step to the appropriate workstation input buffer.

Under CONWIP control, replenishment jobs flow through the system in a similar fashion to how jobs complete their processing under push control. Once the final step of a replenishment job is completed, this will be added to the FGI buffer. Assuming there are outstanding requests for FGI corresponding to this job, the FGIMA will return the requested inventory to the JMA. This will then attempt to match this with other stored completed portions for the same order, initiating the replenishment of the received inventory as detailed above. If the full order can be fulfilled the JMA will store the job's performance information and inform the PMA and SMA that the job is finished.

The JMA's job managing role further involves reacting to notifications it receives from other agents regarding job cancellations. The JA handles these events as follows:

- If the system operates under push control, the JA will simply terminate the job in question and record this in the performance tracing log. It will then inform the SMA that the job is finished.
- If the system is controlled by a Kanban, Base Stock or CONWIP mechanism, the JA will react by returning already processed replenishment inventory to the appropriate storage buffer. The inventory will be relinquished as follows. If there is at least one unit of processed inventory, under Kanban or Base Stock control this will be returned to the output buffer of the workstation which performed the processing. Under CONWIP control, inventory relinquishment affects portions of FGI which have been reserved to partially fulfil an order. Portions associated with the specific cancelled order will be returned to the FGI buffer.

Another important capability that the JA has concerns the way it responds to machine faults. As soon as an MA notifies the JA that a fault caused damage to WIP (associated with the job managed by the JA), the JA will arrange for the scrapped WIP to be replaced. If only a portion of the WIP is scrapped, the JA will merely replace this. The JA will log the creation of the replacement job and store its start time. If the system operates under push control, the JA will initiate the job by sending a task for its first processing step into the input buffer of the workstation responsible for carrying this out. The JA will initiate the job's second processing step as soon as the machine which performed the first processing step notifies it that the task is completed and so forth. Any damage caused to WIP when the system operates under CONWIP control concerns replenishment WIP which will be replaced following the same procedure.

The JA follows a different protocol to replace scrapped WIP portions under Kanban and Base Stock control. Assuming that the machine which developed the fault performs the 5<sup>th</sup> processing step of a job, its JA will request inventory matching the job type, processing step and batch size of the damaged portion from the output buffer of the workstation which performs the job's 4<sup>th</sup> step. If there is inventory available, the JA will ask the workstation where the fault developed to add this to its input buffer for processing. Once processed, this WIP will replace the scrapped portion. The JA will replace the inventory removed from the workstation which performs the 4<sup>th</sup> processing step, by requesting inventory from the output buffer of the workstation performing the 3<sup>rd</sup> processing step of the job and so forth. Asynchronously with the aforementioned actions, the JA will send a new task into the input buffer of the workstation performing the 1<sup>st</sup> processing step of the job.



#### **5.4.1.7 Workstation Supervisor Agent (WSA)**

The WSA is responsible for managing the machine(s) that belong to a given workstation. Its primary function is to respond to machine status changes by taking appropriate action. Following a notification sent by any of the machines in the workstation, the WSA will proceed as follows:

- The status of a machine has changed to idle. The WSA will communicate with the DA and ask it to assign a new task to the machine in question. The WSA will further provide the DA with details of the set-up operation last performed by this specific machine. This information is relevant to the DA as it will initially attempt to assign a new task to the machine already configured to perform the same type of processing. In other words the DA will check if there are any idle machines within the workstation which have just finished processing the same job type and step so that the new task can be assigned to one of them.
- The status of a machine has changed to out-of-order. The WSA will instruct the DA to cancel a previous request to assign a new task to the machine that has developed the fault.
- The status of a machine has changed to busy. In this case, the WSA does not need to take any action.

The WSA handles notifications it receives from the SMA regarding job cancellations and the system's termination and informs the MAs it has control of that they need to finish their operation. The detailed actions of the MAs in response to these notifications were discussed in Section 5.4.1.5. The WSA will wait for the MAs to confirm receipt of its instructions. In case of job cancellations, the WSA will reply to the original notification it received from the SMA and confirm that it finished processing the cancellation. If the original notification concerned the system's termination, the WSA will respond to confirm it has finished its operation.

#### **5.4.1.8 Workstation Input Buffer Agent (WIBA)**

WIBA instances are created when the system operates under push, pull and CONWIP control. Although there may be several machines in a workstation, these all share the same input buffer. The WIBA's role is to manage its "input\_buffer" database. In doing so, the WIBA takes the following actions:

1. Adds new tasks to its buffer. The WIBA responds to requests it receives from the JA to add WIP to its buffer. Initially, it records the task's information

(including its arrival time) in its database. It then informs the DA that a new task has arrived and needs to be assigned to one of the machines of the specific workstation.

2. Provides information about the tasks in its buffer to the DA. Whenever the DA is ready to perform its job prioritisation and assignment routine, it requests up-to-date information from the WIBA about the tasks stored in its buffer. The WIBA responds to these requests by providing all the relevant task information (processing times, set-up times, due-dates etc.) that will enable the DA to select the machine to which it can release the next task.
3. Removes from its buffer tasks assigned to machines for processing. This action simply involves updating its database by deleting the task in question.

The WIBA is also designed to respond appropriately to notifications sent by the SMA concerning job cancellations and the system's termination. If a job is cancelled, the WIBA will terminate and remove from its buffer all tasks (if any) associated with this specific job. It will then send appropriate notifications to confirm that it reacted to the job cancellation. Its actions will be similar in case the original notification concerned the system's termination. However, in that case, the WIBA will also send a notification to confirm that it is finishing its operation.

#### **5.4.1.9 Workstation Output Buffer Agent (WOBA)**

WOBA instances are only created when the system is set to operate under Kanban and Base Stock control. The main role of the WOBA is to manage its inventory. Each WOBA maintains two databases which are dynamically updated. These are the "inventory" and "request log" databases.

The request log stores requests for the release of inventory to succeeding stations. The requested inventory is used to replenish WIP consumed by other stations further downstream. Whenever a JA forwards a new request for inventory to the WOBA the latter generates a unique identification number (ID) for it and logs its details (request id, job type, processing step, and priority) in its database.

The WOBA handles requests to add WIP processed by the workstation's machines to its buffer. These requests are forwarded to the WOBA by the respective JAs. The WOBA modifies its inventory database by recording information about the received inventory (job type, processing step and quantity). It then checks its inventory requests log for pending requests for this type of inventory (specific job type and step). Initially, the WOBA will attempt to only fulfil requests for inventory associated with high-priority

jobs. Only when there are no competing requests for high-priority jobs that can be fulfilled (using the available inventory) the WOBA will proceed to process requests for normal-priority jobs. Assuming there are more than one unfulfilled requests for the same type of inventory, the WOBA will process the request that it received first. In line with these rules the system is able to provide inventory on a FCFS basis whilst ensuring high-priority jobs are always prioritised over normal-priority jobs. The WOBA then completes the following steps:

1. It determines the quantity of inventory needed to fulfil the request.
2. It confirms it has some inventory available to fulfil the request (in full or partially). The confirmation is recorded in the system's performance tracing log.
3. It removes as much inventory as possible from its buffer and updates its inventory database accordingly.
4. It allocates the inventory to the request. To complete this action, it modifies its inventory request log by either removing or updating the original request (depending on whether the latter is fulfilled in full or partly) and records this in the system's performance tracing log.
5. It responds to the JA which submitted the original request and supplies it with details of the WIP items being provided.
6. It attempts to process any other requests in its log in case it has more inventory stored for the specific job type (and processing step).

#### **5.4.1.10 FGI Manager Agent (FGIMA)**

The FGIMA is only relevant when the system operates under CONWIP control. The job-shop's overall configuration in that case resembles that of the push system i.e. workstations comprise several machines and a common input buffer. There is further a central point where all finished products are stored and this is controlled by the FGIMA.

The FGIMA has the same basic design as the WOBA used in Kanban and Base Stock control. It interacts mainly with JA instances and its main function is to manage its inventory. This involves receiving requests for FGI and logging them into its request log database. The FGIMA also updates its inventory database whenever a request for FGI can be completely or partially fulfilled. When the FGIMA receives a notification from the SMA that the system is terminating it clears its inventory and sends a response to indicate that it has finished its operation.

#### **5.4.1.11 Performance Monitor Agent (PMA)**

The PMA is responsible for compiling the system's performance report. The report comprises the scheduling performance metrics presented in Section 5.3. In order to carry out the mathematical computations involved in determining the values of the performance measures, the PMA needs to obtain specific data from the MAs and JAs.

All MA instances in the system are designed to send notifications to the PMA before terminating their operation. In addition, they supply the PMA with data which allows it to determine the total time the machines were in operation. This information is required for the computation of the machine utilisation performance metrics.

However, the majority of the performance metrics are computed using job data provided to the PMA by the JA instances created in BASS. The data comprises the job's initial parameters such as the type, ready time, due-date, batch size and priority as well as data concerning the job's processing start time, completion time and progression through the system, e.g. time the job spent queuing for processing. Each JA in the system supplies this data to the PMA as soon as the job it manages is completed i.e. the associated order is fulfilled. The PMA produces the system performance report after receiving a notification from the SMA that the system has terminated its operation.

### **5.5 BASS Development**

Intelligent software agents can be developed using general programming languages such as C++ and Java™. However, most large-scale applications of multi-agent systems require specialised agent development platforms (Macal and North, 2010). These platforms or toolkits provide the host environment for developing and deploying software agents. They also provide the enabling infrastructure for agent registration, communication, coordination, security etc. (Mařík and McFarlane, 2005). By providing the basic building blocks for agent realisation, agent toolkits allow developers to focus on designing and implementing distributed communities of agents for particular applications without having to develop the infrastructure required to support these from scratch (Mascardi et al., 2008).

Shen et al. (2006) point out that agent toolkits must follow standards stipulating what infrastructural components they need to provide and how the latter should be designed. The most widely recognised agent standards are developed by the Foundation of Intelligent Physical Agents (FIPA). FIPA's standards are presented in the form of

specifications which prescribe amongst others, agent abstraction models and architectures, message exchanges mechanisms, coordination protocols and agent management (FIPA, 2012).

With widely known development environments ranging from open-source software to commercial platforms, significant research efforts were undertaken in recent years to evaluate and compare these (Nguyen, et al., 2002; Vrba, 2003; Hao et al., 2005; Unland et al., 2005; Monostori et al., 2006; Bordini et al., 2009).

One such comprehensive review is presented by Luck et al. (2004) and considers JACK™, the platform used in this thesis for the realisation of BASS. Their study compares the features of FIPA-compliant JACK™ to those of other toolkits including JADE, Zeus, RETSINA, IMPACT and Living Markets. The analysis suggests that JACK™ provides the most faithful interpretation of the BDI architecture. It is further found to provide one of the most lightweight solutions. The Graphical User Interface (GUI) of JACK™, which is fully integrated into its development environment, is another emphasised feature. The comparative analysis recognises JACK™ as the most refined toolkit arguing this is not unexpected given its commercial background. The robustness of JACK™ and suitability for large-scale industrial applications is also strenuously argued by Winikoff (2005).

### **5.5.1 JACK™ overview**

JACK™ is a development environment for building and running autonomous agents and MAS. It is produced by Agent Oriented Software (AOS) Group and it is currently at version 5.6. JACK™ is a commercial platform but an academic license is also available to support research applications (AOS, 2012a). All JACK™ agents are programmed in JACK™ Agent Language (JAL) which extends JAVA with the necessary constructs to support agent-oriented behaviour. JACK's Agent Compiler converts JAL source files into JAVA code. As a result, JACK™ agents can run on any JAVA platform (AOS, 2012b).

The main reasons for the selection of JACK™ for the realisation of BASS are the following:

1. The reasoning behaviour of JACK™ agents follows the BDI model of abstraction which provides the best representation of human cognitive processes (AOS, 2012c). Not only does this facilitate the encoding of JACK™

software agents but it also results in the most accurate and realistic emulation of human reasoning.

2. JACK™ extends the BDI to provide the necessary support for the developing socially capable agents that communicate and interact with other agents within the environment they are embedded in (Evertsz et al., 2004).
3. JACK™ agents can be programmed to exhibit both reactive and proactive behaviour. They are able to react to event-driven stimuli arising within their environment and proactively seek to achieve their predetermined goals. Consequently, JACK™ autonomous agents can operate in extremely complex and dynamic environments (Wallis et al., 2002).
4. As commercial software, JACK™ is not bound by the limitations and inefficiencies of other available platforms (Fletcher et al., 2003). It is a stable environment with low resource requirements where hundreds of agents can run on any low specification computer.

A JACK™ agent includes the following main programming constructs (AOS, 2012d):

- **Beliefsets.** These are datasets used to store facts and knowledge about the agent's world. Whenever the agent's beliefset is updated, an event is automatically posted to trigger appropriate action by the agent.
- **Events.** These provide the stimulus that initiates agent action. Without events, JACK™ agents would be inactive indefinitely. Events can be internal notifications that the agent posts to itself to start a new task or messages that other agents within its environment send to it. Events are differentiated into normal events and BDI events. A normal event is typically related to an ephemeral phenomenon which causes spontaneous agent reaction. Upon receipt of a normal event, the agent will handle it by executing the first plan which is both relevant and applicable to this event. By contrast, BDI events modify the agent's knowledge and instigate proactive behaviour. In response to BDI events, agents use advanced heuristics to select the most appropriate plan of action from a selection of available plans.
- **Plans.** These contain prescriptive instructions concerning the procedural steps that agents need to follow to achieve their goals. Each plan is designed to handle a specific event. Plans include JAL statements which allow the agent to identify a relevant plan (to handle an event) and further determine the context i.e. the conditions under which plans are applicable. Plans include reasoning methods which are executed so that agents can achieve their goals.

- **Capabilities.** Capabilities are collections of beliefsets, events and plans relevant to each agent. They represent the functionalities encapsulated in each autonomous software agent.

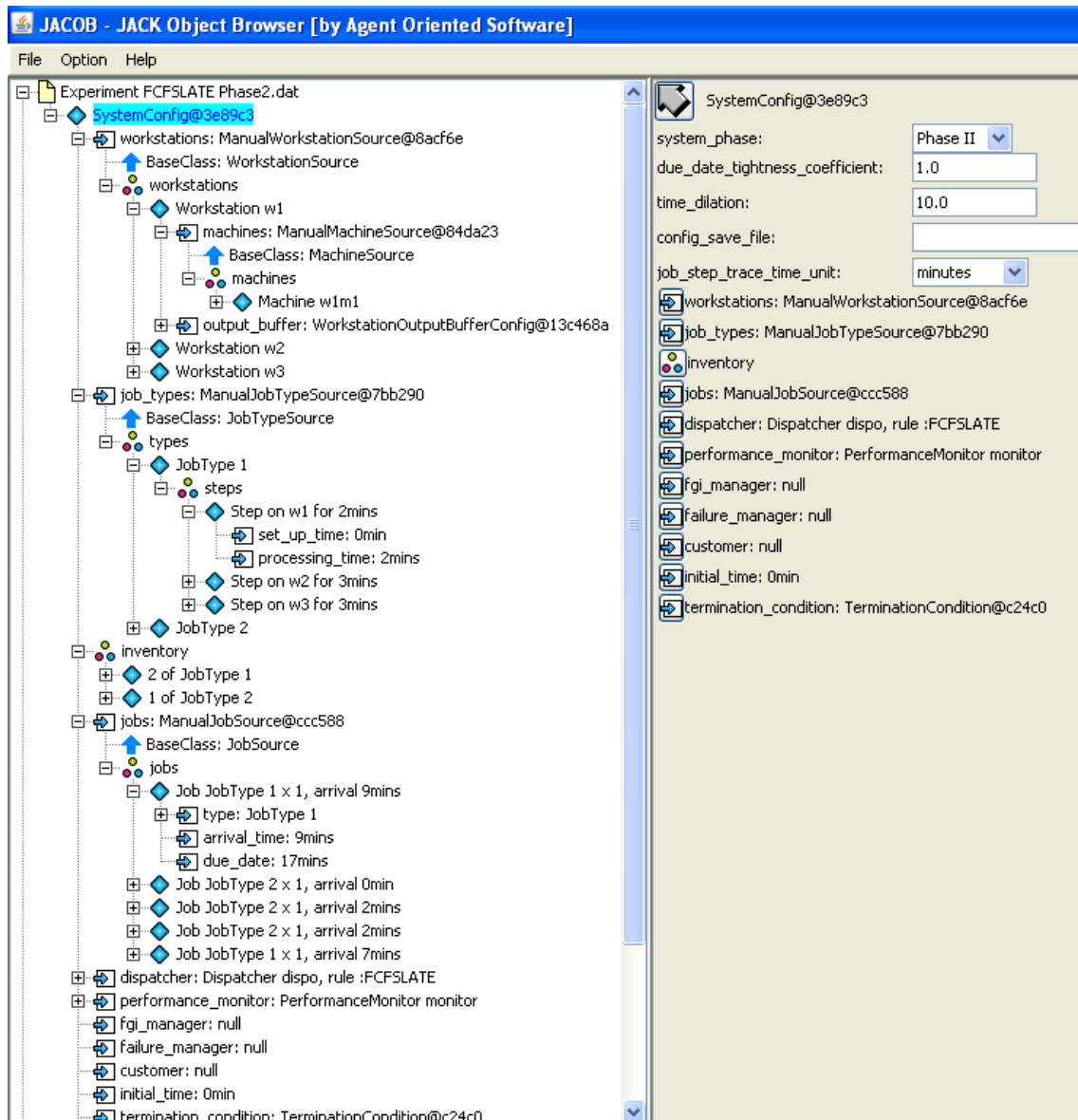
The analogy between the above JACK™ agent constructs and the BDI model is the following. Beliefsets represent the agent's **beliefs** and perceptions about their world. All events (internal notifications, messages exchanged with other agents and beliefsets updates) result in agents developing **desires** to achieve certain goals. Plans that agents have committed to and are about to execute, represent their **intentions** (Shajari and Ghorbani, 2004).

The JACK™ Development Environment (JDE) is a flexible environment for designing, implementing and tracing agent applications. JDE provides a powerful graphical editor interface which allows agents and their constituents to be defined within JACK™ projects. A MAS will typically comprise several projects and the JDE offers the support for their integration into one distributed application. It also allows the reusability of agent components, e.g. plans used by several agents can be easily shared (AOS, 2012e). The JDE's graphical editor can be used to produce diagrammatic representations of the agent's interface, i.e. messages an agent posts (to itself), sends/receives to/from other agents and beliefsets it accesses. These diagrams further show the agent's overall structure (plans and enclosing capabilities). An overview of the JDE and examples of design diagrams are presented in Figures 5.6-9.

The most important feature of the JDE is its advanced plan editor which produces plan-related statecharts. The creation of statecharts (or statechart diagrams) similar to those used in the Unified Modelling Language (UML) are accepted as standard modelling practice in the design of agent-oriented behaviour (Borshchev and Filippov, 2004). As their name suggests, statecharts show the transition of agents through different states. They define the timing of these transitions and further capture messages exchanged, decisions made and subtasks carried out to complete plans. Statecharts are also important visual tracing tools allowing the application's behaviour to be examined at runtime (Shendarkar and Vasudevan, 2006; North and Macal, 2007). Examples of JDE graphical plans are shown in Appendix C.

Another important component of JACK™ is the JACOB™ Object Modeller. JACOB™ is a system that allows object data structures to be stored and transmitted. JACOB™ uses its own language to define objects and their fields in data files (AOS, 2012f). These files can be viewed or edited using JACOB's object browser. Various

configurations of BASS can be created, stored and edited as JACOB™ data files. An overview of the JACOB™ graphical environment is presented in Figure 5.5.



**Figure 5.5 Overview of the JACOB™ interface**

Due to space limitations, this shows the configuration of a small-scale 2x3 problem. In this configuration, BASS is set to operate in Kanban control mode (referred to in the settings as Phase II). Due-dates are defined using the TWC method with a tightness coefficient set to 1. Each of the three workstations in the system comprises a single machine. Overall five instances of the two jobs processed in the system are created. These have a batch size of 1 unit and different arrival times. At initialisation, the output buffers in the system hold 2 units of WIP associated with job 1 and one unit of WIP



associated with job 2. Unit processing times are expressed in minutes and set-up times are ignored. The selected dispatching rule in this configuration is FCFS/LATE.

### **5.5.2 Implementation of BASS using JACK™**

The implementation concerns the realisation of the design presented in Section 5.4. The 11 agent types of BASS were encoded as autonomous agents using JACK™. Having established the overall architectural design of BASS, the implementation involved the following steps:

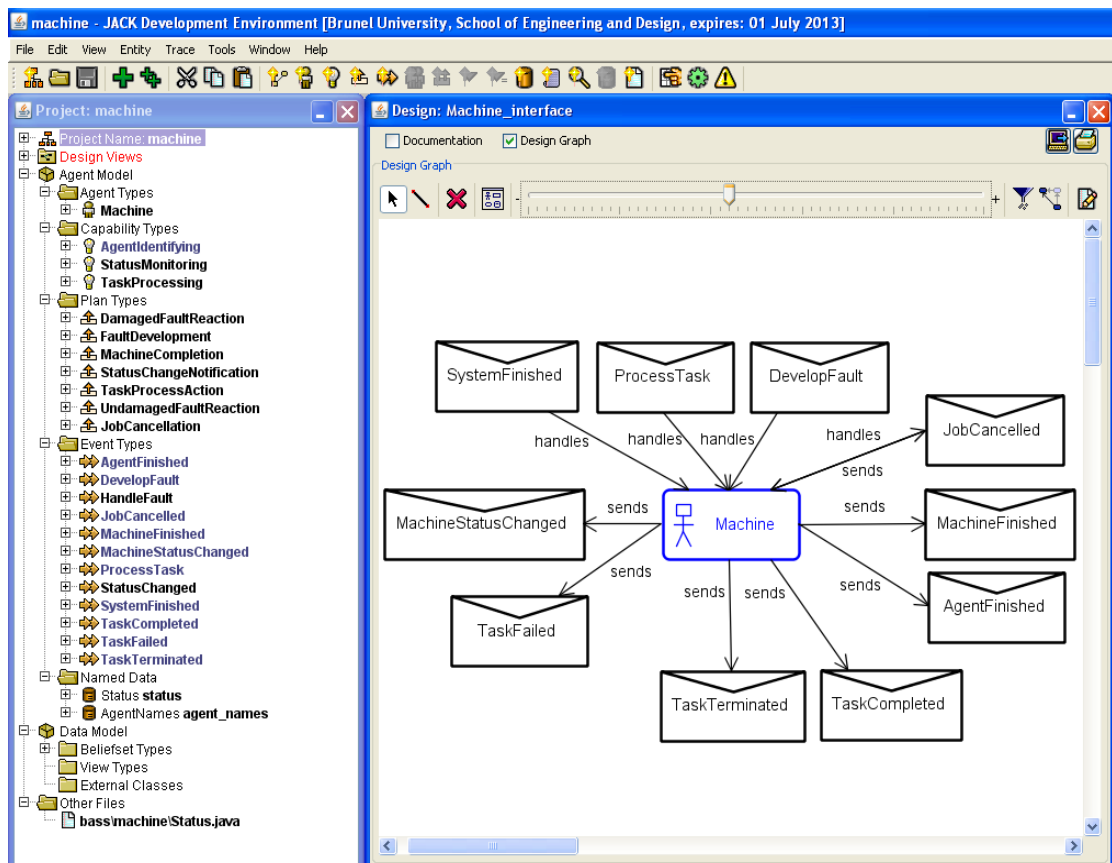
1. Initially, the interactions between the various agent types were determined. At this point it was necessary to define the events exchanged by agents in the Application Programming Interface (API) JDE project. The complete listing of external events in BASS is presented in Appendix B.
2. A JDE project was created for each agent type in BASS. Generally, there was a one-to-one mapping between the agents and projects created with the exception of the workstation project which comprises three agent types, namely the Workstation Supervisor, Workstation Input Buffer and Workstation Output buffer.
3. The appropriate events were imported from the API project into the respective agent projects and design diagrams were created to define the external interface of each agent.
4. By referring to the specifications of BASS, the enclosing capability structure of each agent was built. This included the internal interactions within each agent in the form of messages posted and access to beliefsets.
5. For each event received by other agents an appropriate plan was designed to handle it. Plans were subsequently grouped under the already designed capabilities.
6. The reasoning methods used by plans were implemented using the JDE's graphical plan editor, so that their steps are visually traced at runtime. Other sections of the plans were coded using textual reasoning methods as appropriate.

This section discusses the implementation of the MA which is a moderately complex agent type in BASS. The MA's main constructs are illustrated in Figure 5.6, which also shows the MA's interface. In response to a message sent to the MA, an appropriate plan is invoked to handle the received instruction. The MA sends its own messages to

other agents in the system and these are handled in a similar fashion. The MA's interactions typify those that agents in BASS have with their external world.

The MA supports the following functional objectives:

- Processing the respective operations (tasks) of jobs assigned to it.
- Monitoring and updating its status to show that the machine is question is busy, idle, out-of-order etc.



**Figure 5.6** The MA's main constructs in JDE and design view diagram of its external interface

As shown in Figure 5.7, each of the aforementioned functional objectives is modelled as a JACK™ capability. A third capability, that is AgentIdentifying, is common across most agent types which interact with other agents in BASS. It allows the MA to identify other agents designed to perform specific roles within the system. The TaskProcessing capability encapsulates the functionality required to process tasks. The capability is delivered by the three plans shown in Figure 5.8. The StatusMonitoring capability encapsulates the functionality required to keep track and modify appropriately the machine's status. The four plans associated with this capability are shown in Figure 5.9.

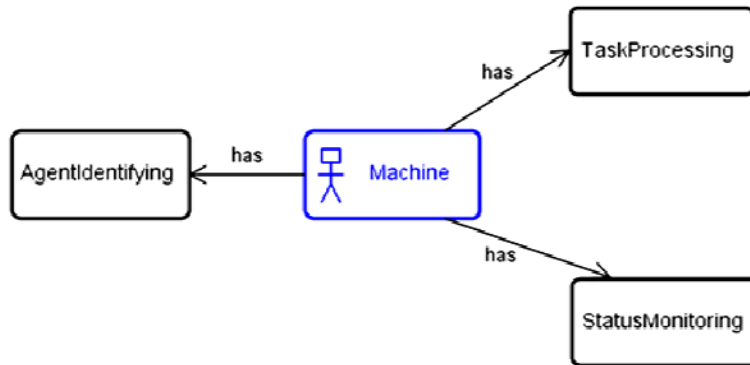


Figure 5.7 Design view diagram of the MA's capabilities

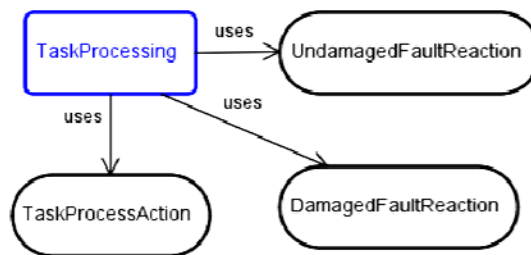


Figure 5.8 Design view diagram of the MA's TaskProcessing structure

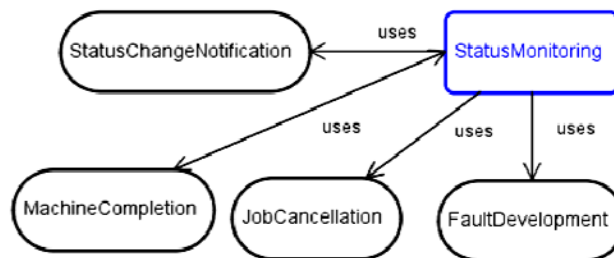


Figure 5.9 Design view diagram of the MA's StatusMonitoring structure

The complete structure of the aforementioned plans in terms of the event handled by each specific plan; messages posted internally within the agent; messages sent to other agents; access to beliefsets and reasoning methods executed within the plan are discussed in detail in Appendix C. The explanation of the architecture, interface and structure of all other agent types within BASS has been omitted for economy.

## 5.6 BASS verification

In order to confirm that the agent-based simulation model, namely BASS is an accurate representation of the modelled job-shop scheduling system, preliminary experimentation was performed using JACK™.

In parallel with the tests run in JACK™, manual simulation was carried out using the same set of experiments. The main objective of the verification process was to compare the scheduling performance output generated by JACK™, with the equivalent expected output produced by manual simulation. In case of discrepancies, the performance tracing logs provided by JACK™ were utilised to determine the points of differentiation between BASS and the manual simulation tests.

A range of experiments were designed to test key operational features of BASS under all four production control modes, namely push, Kanban, Base Stock and CONWIP. The examined features included the following:

- Batch splitting due to insufficient WIP in the output buffers or central FGI depending on the set pull control mode.
- Machine breakdowns.
- Batch splitting following damage to components of the batch in-progress.
- Arrival of rush orders and cancellations of jobs released into the system.
- Dispatching rules.
- Machine prioritisation in case of workstations with parallel machines.

The designed experiments involved small-scale problems, so that they would not be too cumbersome to carry out by manual simulation. The manual simulation of one such experiment referred to as BVE1 is presented in Appendix D. This models the operation of the modelled job-shop under push control. It involves two different jobs which arrive at the system at different times. Both jobs are processed in batches of five units. Each job visits the three available machines by following its own routing. Set-up times are ignored in this experiment. The total processing times are used to compute the due-dates in line with the TWC method. The implemented dispatching rule is SIPT. Two machine faults occur at times  $t=17$  and  $t=28$  minutes affecting machines 2 and 3 respectively. Both faults damage certain components of the batches being processed. The manual simulation experiment demonstrates how the scrapped components are being removed and replaced by new raw materials which are released into the system for processing.

The manual simulation provides snapshots of the operation of the system at various timestamps identifying the end of processing of a job on a certain machine and the assignment of a new task. Timestamps also relate to unexpected disturbances such as the two machine breakdowns. Performance information related to the fulfilment of orders as well as the time jobs spent in queue, level of WIP and machine utilisation between consecutive timestamps is recorded. The final table collates all the performance information and computes the performance metrics generated by JACK™. These are shown at the end of the manual simulation experiment for comparison. The performance tracing log created by JACK™ for this specific experiment is provided in Appendix E.

A modified version of BVE1, referred to as BVE2 is created to test the job-shop's operation this under CONWIP control. This is presented in Appendix F. BVE2 has a similar configuration to BVE1, however, in this case, faults are developing at  $t=17$  and  $t=23$  minutes to ensure that the affected machines are busy processing parts and allow the system's reaction to damaged WIP to be observed.

Appendix G presents a manual simulation experiment referred to as VBE3 which is carried out to test the job-shop's operation under Kanban control and the scheduling system's reaction to the arrival of a rush order at  $t=9$  minutes and its subsequent cancellation at  $t=10$  minutes following its partial fulfilment. Finally, Appendix H shows a manual simulation experiment testing the operation of the job-shop which is configured to use parallel machines. The job-shop is controlled by the Base Stock mechanism and set-up times are considered in this experiment. In all cases the direct comparison of the manual simulation results and output generated by JACK™ reveals no discrepancies.

## **5.7 Chapter summary**

This chapter introduced the production system used for the application of push and pull control. An analysis of how pull control exercised by the Kanban, Base Stock and CONWIP mechanisms can be applied to non-repetitive production systems employing job-shops was provided. This thesis proposes a direct implementation of pull control principles as they were designed for repetitive production lines without any attempt to modify the original pull control logic or adapt the job-shop's functional layout. However, the chapter draws attention to the following practical challenges that need to be addressed to ensure a successful application of pull control in job-shops:

- In order to implement the pull control logic in job-shops, the production system needs to be filled with some initial inventory. Under Kanban and Base Stock

control this concerns WIP placed in all workstation output buffers whereas in the case of CONWIP this is FGI kept at the system's final storage point. The purpose of filling the system with some initial inventory is two-fold. This inventory is used to meet some of the demand. The job-shop is further designed to trigger production in all upstream stages and replenish the inventory to pre-set levels. The replenished inventory is used to fulfil outstanding orders and so forth.

- An issue can potentially arise when the initial inventory is not enough to meet demand. The system would not be able to meet customer orders, no inventory would be consumed to satisfy the demand and therefore the replenishment process would never be instigated. Situations like these would bring the whole system to a standstill. In order to mitigate the risk of deadlocks, a batch-splitting function is built into the designed job-shop.
- Batch-splitting ensures that orders can be split into two sub-batches, one that can be fulfilled using the already available inventory and a second for which a request for inventory is placed at the respective storage point.

In order to allow the job-shop's scheduling system to perform the above functions, its distributed and heterogeneous entities, e.g. jobs, inventory storage buffers etc. have to engage in continuous interactions and coordinated decision-making. Due to the sheer volume and diversity of orders typically processed by non-repetitive production systems, the volume and complexity of such interactions is significantly high and can be further exacerbated under unexpected disturbances affecting dynamic job-shops. Developing further the key arguments presented in Chapter 4 where the potential of MAS was explored in great depth, this chapter endorses the suitability of MAS in handling these technical challenges.

Having discussed the operation of the designed job-shop scheduling system under both push and pull control, the analysis focuses on the job-shop's configuration outlining the input data that needs to be supplied and defining suitable metrics that will be used to assess its performance. Attention is further drawn to important initialisation parameters such as the system's initial inventory level and dispatching rules implemented to prioritise jobs.

Following on from the presentation of the complete specification model, the chapter examines its implementation into a MAS referred to as BASS. The types of agents in BASS and associated number of instances are reviewed in detail with the discussion focusing on their functions and external interfaces. Major emphasis is placed on certain agents for instance the JA and WOBA which deliver most of the functionality required

to implement the pull control logic. An overview of JACK™, the development platform used to implement BASS is also provided, highlighting the robustness of JACK™ in coping with complex, large-scale industrial applications. The main constructs of JACK™ agents are briefly discussed followed by a sample presentation of how the MA in BASS is implemented in JACK™.

Simple experiments are also devised and used to verify BASS by performing a simulation. The aim of the verification process is to confirm that the model does what it was designed to do. For this purpose, the testing carried out at this stage involved extensive event and entity tracing. It further sought to determine the system's performance and compare it with the performance output generated by JACK™. The problem scenarios used for the verification of BASS were designed to test most of its features; however, key emphasis was placed on its ability to simulate the operation of the job-shop scheduling system under push and pull production control. The verification process has served to provide full confidence in BASS.

BASS is extensively tested in Chapter 6, which presents a wide range of static and dynamic problem instances used to validate the model but most importantly answer the two research questions formulated in this thesis. BASS is further utilised to examine the application of pull production control to an industrial case study.

## **6 Experimental and empirical validation of the application of pull production control to non-repetitive production systems**

This thesis posits that pull control can be extended to job-shops and improve their scheduling performance. Agent-based simulation is proposed as the most suitable tool for testing this assertion. The design and implementation of the agent-based scheduling system, namely BASS, developed to test the transferability of pull production control to job-shops were discussed in detail in Chapter 5. This chapter presents the experimental and empirical testing undertaken to answer the research questions posed in this thesis.

The remainder of the chapter is organised as follows. Job-shop scheduling problem instances sourced from the literature are identified and discussed in section 6.1. As these employ deterministic data, a second set of problem instances is developed and enriched with stochastic data to allow the performance of pull production control to be evaluated under dynamic conditions. Certain problem parameters are modified in order to perform sensitivity analysis and examine the impact of these changes on the performance of push and pull production control. The application of pull control to a real-life non-repetitive manufacturing system is presented in section 6.2. The simulation output generated using the case data is found to be at odds with the results obtained from the static and dynamic experiments. These results are analysed in light of the significant case data limitations in section 6.3 which presents the main conclusions of this chapter.

### **6.1 Experimental testing using BASS**

In order to test the application of pull control in job-shops and compare its performance with that of push control, a series of simulation experiments are formulated comprising both deterministic and stochastic scheduling data. The computational experiments are based on the benchmark job-shop scheduling problems proposed by Beasley (2012). The specific benchmark problems were considered in previous simulation studies concerned with the optimisation of job-shop scheduling performance (Mönch, 2007; Kouider and Bouzouia, 2012) and found to be suitable test-beds.

#### **6.1.1 Design of experiments**

The benchmark problems determine the job-shop's configuration by specifying the number of machines, job types as well as job routings and processing times (in seconds). They involve various assumptions, for instance, they consider a non re-



entrant job-shop where all jobs are available at time zero and set-up times are negligible. New orders concern single units of products. The original benchmark problems do not specify due-dates. However, as several metrics in the simulation output concern performance metrics involving due-dates, the latter are computed using the TWC method. Due-dates are considered to be tight in all the adapted experiments and for this reason the due-date tightness coefficient is set to 1.

The data sourced from the original benchmark experiments is supplemented with additional parameters set to regulate the operation of the scheduling system under pull control. The level of inventory in the job-shop's output buffers and final storage points in the case of Kanban/Base Stock and CONWIP respectively is a key system initialisation parameter. It facilitates the operation of the job-shop when pull control is exercised allowing the system to meet initial demand (at least in part) using the available inventory and trigger production to replenish this to preset levels to fulfil outstanding orders. Initial inventory levels directly influence the system's performance in respect to service level, i.e. the percentage of orders fulfilled using existing stock. Given that job-shops operate as MTO systems, inventory levels should be kept as low as possible. In the case of the static scheduling problems, the initial inventory allows the system to meet 20% of the anticipated demand for the given production period thus providing an equivalent service level. In order to achieve this, the inventory level at system initialisation is set to one unit per job type.

An overview of the full set of the static scheduling parameters is provided in Table 6.1. This shows the total number of multiple job instances (orders) in each adapted benchmark problem. The rationale for the generation of multiple orders is to facilitate the testing of pull control mechanisms. Five different orders are received for each job type; however, at system initialisation the job-shop can only fulfil one of these using its available inventory. All remaining orders are fulfilled progressively each time the system replenishes the consumed FGI.

**Table 6.1 Experimentation parameters for static problems**

<b>Benchmark Problem</b>	<b>Jobs x Machines</b>	<b>Number of job instances</b>	<b>Number of operations</b>	<b>Input Data</b>
LA16	10x10	50	5,000	– Job processing times and routings (as per benchmark problem) – Arrival times, t=0 – Set-up times=0 – Batch sizes: single units – Due-dates based on the TWC method, tightness coefficient=1
LA20	10x10	50	5,000	
LA21	15x10	75	11,250	
LA25	15x10	75	11,250	
LA26	20x10	100	20,000	
LA30	20x10	100	20,000	
LA31	30x10	150	45,000	
LA35	30x10	150	45,000	
SW15	50x10	250	125,000	
SW20	50x10	250	125,000	

Each adapted benchmark problem from those presented in Table 6.1 is utilised to test the system's performance under the four production control policies considered, i.e. push, Kanban, Base Stock and CONWIP. Since dispatching rules influence the job-shop's performance, experiments were carried out using a selection of dispatching rules. SIPT, EDD, WINQ and FCFS/LATE are considered in order to have a fair representation of each of the four dispatching rule groups identified in Figure 5.4. As a result, every benchmark problem is associated with four sets of experiments corresponding to the four production modes, with each comprising four different tests involving the selected dispatching rules. Therefore, each of the 10 benchmark problem has 16 variations and the overall number of designed static experiments is 160.

Five larger scale benchmark problems adopted from Beasley (2012) are appropriately modified using stochastic data to model the job-shop's performance under dynamic conditions. In addition to considering a larger number of job types, machines and total number of operations, the dynamic experiments avoid the unrealistic assumptions of static experiments which concern simultaneous job arrivals at system initialisation, negligible set-up times and single unit batches. The dynamic conditions considered in this set of dynamic experiments concern:

- the order mix,
- job arrival times,
- set-up times,
- batch sizes and
- unexpected disturbances i.e. machine breakdowns and high-priority (rush) orders.

In order to generate a random mix of jobs that visit the job-shop for processing whilst ensuring an equal probability of arrival for the various job types, job orders are generated using the uniform distribution. The min value of the uniform distribution is set to 1 in all experiments whereas the max value is set equal to the number of job types in each benchmark problem. Batch sizes are assumed to be following the triangular distribution with min, mode and max values set to 50, 150 and 250 respectively in all benchmark problems. All batch sizes are rounded to the closest multiple of 10. The machine set-up time to changeover from processing operation  $p_{ij}$  to  $p_{kl}$  is accepted to be uniformly distributed with a min and max values set to 0 and  $p_{kl}$  respectively (Ovacik and Uzsoy, 1994). The inter-arrival time of dynamic orders is known to be best represented by the values of the exponential distribution with a mean determined using equation 5.5 (Vinoda and Sridharan, 2008):

$$\alpha = \frac{\mu_p \times \mu_g}{U \times m} \quad (5.5)$$

Where:

$\alpha$  = mean job inter-arrival time following the exponential distribution

$\mu_p$  = mean total processing time (considering all job types)

$\mu_g$  = mean number of operations per job

U = Machine utilisation level (assumed in these experiments to be 80%)

M = Number of machines in the job-shop

In this set of dynamic experiments, machine breakdowns are assumed to have no impact on batches being processed on the machines which develop faults. The machines which develop faults are generated using a uniform distribution with a min value equal to zero (the machine index in the original benchmark problems and BASS experiment configuration files starts from zero) and a max value equal to the number of machines in each benchmark problem. The time between failures and machine downtime (time to repair) are accepted to be best represented by the exponential distribution with the Mean Time Between Failures (MTBF) assumed to be 1,000 (sec) and Mean Time To Repair (MTTR) calculated using the equation below (Renna, 2010):

$$MTTR = 1.5 \times ePT \quad (5.6)$$

Where:

MTTR = Mean time to repair following an exponential distribution

ePT = Expected processing time for all jobs considered, set equal to  $\mu_p$  in Equation 5.5

Rush orders can represent 10-20% of the overall orders (Thürer et al., 2010) so their proportion in the order mix is set to 20% in all the dynamic experiments. In terms of their mix, arrival times and batch sizes rush orders are generated in the same fashion as normal-priority orders. The only difference is that the mean time of the exponential distribution used to generate their arrivals is assumed to be five times higher than the respective mean inter-arrival time for normal priority orders. The utilised benchmark problems and associated experimentation parameters are presented in Table 6.2.

Similarly to the static experiments, each of the five dynamic experiments includes a sub-set of 16 variations which account for the four production control modes and different dispatching rules namely, LTPT, MS, LOPNR and RL. The overall number of dynamic experiments included in the complete set is therefore 80. Given that the

volume and order mix in these experiments is rather unpredictable the initial level of inventory at system initialisation is set to 250 units for each job type. Assuming the system receives five orders of the same job type which have the maximum batch size considered, the initial inventory available can provide an estimated service level of 20%. However, as demand is variable the actual service level can only be ascertained using the performance statistics collected at the end of the simulation.

**Table 6.2 Experimentation parameters for dynamic problems**

Benchmark Problem	Jobs x Machines (nxm)	Total job instances	Number of operations	Input Data
LA40	15x15	31	6,975	<ul style="list-style-type: none"> <li>– Job inter-arrival times ~ Exponential (62)</li> <li>– MTTR ~ Exponential (74)</li> <li>– Rush order inter-arrival time ~ Exponential (310)</li> </ul>
ABZ7	20x15	38	11,400	<ul style="list-style-type: none"> <li>– Job inter-arrival times ~ Exponential (30)</li> <li>– MTTR ~ Exponential (36)</li> <li>– Rush order inter-arrival time ~ Exponential (150)</li> </ul>
ABZ9	20x15	38	11,400	<ul style="list-style-type: none"> <li>– Job inter-arrival times ~ Exponential (30)</li> <li>– MTTR ~ Exponential (36)</li> <li>– Rush order inter-arrival time ~ Exponential (150)</li> </ul>
YN1	20x20	38	15,200	<ul style="list-style-type: none"> <li>– Job inter-arrival times ~ Exponential (36)</li> <li>– MTTR ~ Exponential (44)</li> <li>– Rush order inter-arrival time ~ Exponential (180)</li> </ul>
YN4	20x20	38	15,200	<ul style="list-style-type: none"> <li>– Job inter-arrival times ~ Exponential (36)</li> <li>– MTTR ~ Exponential (44)</li> <li>– Rush order inter-arrival time ~ Exponential (180)</li> </ul>
All experiments Order mix ~ Uniform [1,n] Batch sizes ~ Triangular [50,150,250] Set-up times ~ Uniform [0, p <sub>ki</sub> ] Due-dates set using the TWC method – affected by dynamic job arrival times Machines developing faults ~ Uniform [0,m] Pull inventory levels set to 250 units for each job type				

As it is important to ensure that the same pool of stochastic data generated for each of the benchmark problems considered is used in all of the 16 variations, the data was generated externally using Arena's Factory Analyser version 9.0. The stochastic data used in the five sets of dynamic experiments is presented in Appendices I-M.

### 6.1.2 Computational results

The experimentation results for the 10 static problems are presented in Tables 6.3-12. These also show the range of criteria used to evaluate the job-shop's performance under push and pull control. The tables are colour-coded to enable the immediate identification of the production control modes which result in the most superior and inferior performance (blue and red shaded cells respectively). Certain cells are shaded with a light green colour to indicate that the best performance is achieved by implementing push control. Each table is supplemented with a column which records the percentage performance differential resulting from the adoption of pull control. No percentage values are presented when pull is outperformed by push control.

Given the assumed negligible set-up times, no performance statistics are collected in that respect for the static experiments. The analysis will focus on all performance metrics except for the fill rate. By default, a fill rate exceeding zero can be observed only when production is controlled using any of the pull control mechanisms. Therefore, there is no basis for comparison between push and pull control as far as the fill rate is concerned. The fill rate is computed in order to determine the level of demand met using the system's initial stock. Demand in the static experiments is stable resulting in a predictable service level of 20% across all experiments in which pull control is adopted. Consequently, the collection of performance statistics with regards to the fill rate is mostly relevant in the dynamic experiments. The analysis of the results reveals a number of significant findings:

1. In all the ten static experiments, Kanban control consistently outperforms push and other pull control mechanisms in terms of six out of the nine performance criteria. These criteria are presented below. The percentage shown in brackets represents the respective average performance differential achieved across all ten experiments by implementing pull control exercised by the Kanban mechanism:
  - Total throughput time (81%)
  - Average flow time (72%)
  - Makespan (38%)
  - Tardiness (80%)
  - Combined earliness/tardiness (80%)
  - Number of tardy jobs (20%)

**Table 6.3 Performance output for LA16 (10x10) 50 job instances**

Performance Metric	Push				Kanban				Base stock				CONWIP				Pull adoption performance differential
	SIPT	EDD	WINQ	FCFS/LATE	SIPT	EDD	WINQ	FCFS/LATE	SIPT	EDD	WINQ	FCFS/LATE	SIPT	EDD	WINQ	FCFS/LATE	
Total throughput time (sec)	91,842	94,893	101,472	127,758	38,039	34,922	26,011	26,425	42,477	38,366	30,327	34,420	82,297	89,134	87,356	89,134	80%
Average flow time (sec)	2,272	2,384	2,342	2,812	1,296	1,234	1,055	1,064	1,385	1,302	1,142	1,224	2,181	2,318	2,282	2,318	62%
Makespan (sec)	3,893	4,251	3,735	4,134	2,937	2,329	2,658	1,927	2,842	2,529	2,685	2,395	4,814	4,576	4,292	4,576	48%
Tardiness (sec)	86,847	92,468	90,362	113,833	38,039	34,922	26,011	26,425	42,477	38,366	30,327	34,420	82,297	89,134	87,356	89,134	77%
Earliness/Tardiness (sec)	86,847	92,468	90,362	113,833	38,039	34,922	26,011	26,425	42,477	38,366	30,327	34,420	82,297	89,134	87,356	89,134	77%
Number of tardy jobs (jobs)	50	50	50	50	40	40	40	40	40	40	40	40	40	40	40	40	20%
Fill rate (%)	0	0	0	0	20	20	20	20	20	20	20	20	20	20	20	20	-
Average machine utilisation (%)	69	63	72	65	76	81	75	80	78	81	81	82	50	50	52	50	21%
WIP level (jobs)	29	28	31	34	19	49	28	51	34	62	59	71	8	9	9	9	73%
Total time in queue (sec)	86,847	92,468	90,362	113,833	39,707	112,250	61,723	99,993	84,645	159,610	157,079	175,982	13,949	17,509	15,761	17,509	84%

**Table 6.4 Performance output for LA20 (10x10) 50 job instances**

Performance Metric	Push				Kanban				Base stock				CONWIP				Pull adoption performance differential
	SIPT	EDD	WINQ	FCFS/LATE	SIPT	EDD	WINQ	FCFS/LATE	SIPT	EDD	WINQ	FCFS/LATE	SIPT	EDD	WINQ	FCFS/LATE	
Total throughput time (sec)	102,181	79,171	117,668	144,016	31,314	30,095	21,541	19,373	36,458	34,481	26,339	30,314	84,551	86,213	87,185	86,213	87%
Average flow time (sec)	2,504	2,361	2,640	3,081	1,171	1,146	975	932	1,274	1,234	1,071	1,151	2,236	2,269	2,288	2,269	70%
Makespan (sec)	3,837	4,624	3,802	4,109	3,540	2,087	3,092	1,749	3,540	2,359	3,540	2,225	4,664	4,514	4,323	4,514	54%
Tardiness (sec)	97,951	90,807	104,782	126,841	31,314	30,095	21,541	19,373	36,458	34,481	26,339	30,314	84,551	86,213	87,185	86,213	85%
Earliness/Tardiness (sec)	97,951	90,807	104,782	126,841	31,314	30,095	21,541	19,373	36,458	34,481	26,339	30,314	84,551	86,213	87,185	86,213	85%
Number of tardy jobs (jobs)	50	50	50	50	40	40	40	40	40	40	40	40	40	40	40	40	20%
Fill rate (%)	0	0	0	0	20	20	20	20	20	20	20	20	20	20	20	20	-
Average machine utilisation (%)	71	59	72	66	67	82	73	80	67	83	67	84	52	53	53	53	12%
WIP level (jobs)	33	26	35	37	17	48	26	51	32	65	49	73	8	8	9	8	70%
Total time in queue (sec)	97,951	90,807	104,782	126,841	40,901	100,140	65,947	92,453	100,730	155,540	166,105	169,890	13,168	15,461	15,527	15,461	85%

Performance Metric	Best Performance			
	LA16		LA20	
	Rule	Mode	Rule	Mode
Total throughput time (sec)	WINQ	Kanban	FCFS/LATE	Kanban
Average flow time (sec)	WINQ	Kanban	FCFS/LATE	Kanban
Makespan (sec)	FCFS/LATE	Kanban	FCFS/LATE	Kanban
Tardiness (sec)	WINQ	Kanban	FCFS/LATE	Kanban
Earliness/Tardiness (sec)	WINQ	Kanban	FCFS/LATE	Kanban
Number of tardy jobs (jobs)	All rules	All pull	All rules	All pull
Fill rate (%)	All rules	Push	All rules	Push
Average machine utilisation (%)	SIPT	CONWIP	SIPT	CONWIP
WIP level (jobs)	SIPT	CONWIP	SIPT	CONWIP
Total time in queue (sec)	SIPT	CONWIP	SIPT	CONWIP

**Key:**

- Most inferior performance
- Most superior performance
- Best push performance used to assess performance differential with best performing pull mode

**Table 6.5 Performance output for LA21 (15x10) 75 job instances**

Performance Metric	Push				Kanban				Base stock				CONWIP				Pull adoption performance differential
	SIPT	EDD	WINQ	FCFS/LATE	SIPT	EDD	WINQ	FCFS/LATE	SIPT	EDD	WINQ	FCFS/LATE	SIPT	EDD	WINQ	FCFS/LATE	
Total throughput time (sec)	221,676	179,037	207,766	276,926	78,223	80,439	54,318	56,541	92,160	87,590	66,380	78,419	146,333	154,986	151,446	154,986	80%
Average flow time (sec)	3,408	3,078	3,255	3,962	1,576	1,605	1,257	1,287	1,762	1,701	1,418	1,579	2,484	2,599	2,552	2,599	68%
Makespan (sec)	4,892	6,002	4,834	5,016	4,557	3,798	3,736	3,518	4,561	4,438	3,693	3,740	5,660	5,157	5,184	5,157	41%
Tardiness (sec)	215,642	190,900	204,168	257,146	78,223	80,439	54,318	56,541	92,160	87,590	66,380	78,419	146,333	154,986	151,446	154,986	79%
Earliness/Tardiness (sec)	215,642	190,900	204,168	257,146	78,223	80,439	54,318	56,541	92,160	87,590	66,380	78,419	146,333	154,986	151,446	154,986	79%
Number of tardy jobs (jobs)	75	75	75	75	60	60	60	60	60	60	60	60	60	60	60	60	20%
Fill rate (%)	0	0	0	0	20	20	20	20	20	20	20	20	20	20	20	20	-
Average machine utilisation (%)	82	67	83	80	80	86	84	87	79	81	87	87	64	67	68	67	3%
WIP level (jobs)	52	38	51	59	27	85	45	88	48	89	98	111	11	13	13	13	70%
Total time in queue (sec)	215,642	190,900	204,168	257,146	95,391	322,455	151,045	313,364	197,037	391,656	366,270	422,584	30,629	35,147	35,026	35,147	84%

**Table 6.6 Performance output for LA25 (15x10) 75 job instances**

Performance Metric	Push				Kanban				Base stock				CONWIP				Pull adoption performance differential
	SIPT	EDD	WINQ	FCFS/LATE	SIPT	EDD	WINQ	FCFS/LATE	SIPT	EDD	WINQ	FCFS/LATE	SIPT	EDD	WINQ	FCFS/LATE	
Total throughput time (sec)	168,483	164,952	209,480	288,669	58,522	75,355	47,232	46,664	64,080	81,244	53,095	66,525	137,940	149,495	143,745	149,495	84%
Average flow time (sec)	2,748	2,842	3,209	3,990	1,281	1,505	1,130	1,123	1,355	1,584	1,209	1,388	2,340	2,494	2,417	2,494	72%
Makespan (sec)	4,897	5,384	4,750	5,105	4,134	3,386	3,003	2,732	4,134	3,299	3,290	3,012	5,370	4,958	5,138	4,958	49%
Tardiness (sec)	168,557	175,623	203,157	261,734	58,522	75,355	47,232	46,664	64,080	81,244	53,095	66,525	137,940	149,495	143,745	149,495	82%
Earliness/Tardiness (sec)	168,557	175,623	203,157	261,734	58,522	75,355	47,232	46,664	64,080	81,244	53,095	66,525	137,940	149,495	143,745	149,495	82%
Number of tardy jobs (jobs)	75	75	75	75	60	60	60	60	60	60	60	60	60	60	60	60	20%
Fill rate (%)	0	0	0	0	20	20	20	20	20	20	20	20	20	20	20	20	-
Average machine utilisation (%)	77	70	79	74	81	89	88	88	82	89	89	89	64	65	66	65	9%
WIP level (jobs)	42	40	51	59	28	96	54	96	55	97	106	113	11	13	13	13	71%
Total time in queue (sec)	168,557	175,623	203,157	261,734	90,703	325,700	151,904	269,275	207,319	319,745	351,382	350,442	28,805	35,466	33,610	35,466	83%

Performance Metric	Best Performance			
	LA21		LA25	
	Rule	Mode	Rule	Mode
Total throughput time (sec)	WINQ	Kanban	FCFS/LATE	Kanban
Average flow time (sec)	WINQ	Kanban	FCFS/LATE	Kanban
Makespan (sec)	FCFS/LATE	Kanban	FCFS/LATE	Kanban
Tardiness (sec)	WINQ	Kanban	FCFS/LATE	Kanban
Earliness/Tardiness (sec)	WINQ	Kanban	FCFS/LATE	Kanban
Number of tardy jobs (jobs)	All rules	All pull	All rules	All pull
Fill rate (%)	All rules	Push	All rules	Push
Average machine utilisation (%)	SIPT	CONWIP	SIPT	CONWIP
WIP level (jobs)	SIPT	CONWIP	SIPT	CONWIP
Total time in queue (sec)	SIPT	CONWIP	SIPT	CONWIP

**Key:**  
■ Most inferior performance  
■ Most superior performance  
■ Best push performance used to assess performance differential with best performing pull mode

**Table 6.7 Performance output for LA26 (20x10) 100 job instances**

Performance Metric	Push				Kanban				Base stock				CONWIP				Pull adoption performance differential
	SIPT	EDD	WINQ	FCFS/LATE	SIPT	EDD	WINQ	FCFS/LATE	SIPT	EDD	WINQ	FCFS/LATE	SIPT	EDD	WINQ	FCFS/LATE	
Total throughput time (sec)	312,698	285,272	370,789	477,172	107,950	151,759	101,869	100,245	141,675	163,871	120,902	142,412	237,436	266,087	242,125	266,087	79%
Average flow time (sec)	3,951	3,807	4,308	5,209	1,605	2,043	1,544	1,528	1,943	2,164	1,735	1,950	2,900	3,187	2,947	3,187	71%
Makespan (sec)	6,365	6,847	6,184	6,457	5,278	5,119	5,736	4,872	5,283	5,742	5,834	4,872	6,954	6,631	6,044	6,631	21%
Tardiness (sec)	342,489	328,075	378,199	468,297	107,950	151,759	101,869	100,245	141,675	163,871	120,902	142,412	237,436	266,087	242,125	266,087	79%
Earliness/Tardiness (sec)	342,489	328,075	378,199	468,297	107,950	151,759	101,869	100,245	141,675	163,871	120,902	142,412	237,436	266,087	242,125	266,087	79%
Number of tardy jobs (jobs)	100	100	100	100	80	80	80	80	80	80	80	80	80	80	80	80	20%
Fill rate (%)	0	0	0	0	20	20	20	20	20	20	20	20	20	20	20	20	-
Average machine utilisation (%)	83	77	85	81	85	89	83	91	87	84	82	91	70	72	77	72	9%
WIP level (jobs)	62	56	70	81	40	119	52	129	82	121	118	151	15	18	18	18	73%
Total time in queue (sec)	342,489	328,075	378,199	468,297	179,273	607,322	271,358	632,208	417,048	693,887	686,204	750,887	57,150	73,324	64,174	73,324	83%

**Table 6.8 Performance output for LA30 (20x10) 100 job instances**

Performance Metric	Push				Kanban				Base stock				CONWIP				Pull adoption performance differential
	SIPT	EDD	WINQ	FCFS/LATE	SIPT	EDD	WINQ	FCFS/LATE	SIPT	EDD	WINQ	FCFS/LATE	SIPT	EDD	WINQ	FCFS/LATE	
Total throughput time (sec)	324,896	256,426	345,904	514,929	121,246	139,218	107,480	95,551	145,702	159,183	109,208	137,048	248,765	258,582	249,037	258,582	81%
Average flow time (sec)	4,179	3,832	4,354	5,588	1,746	1,926	1,609	1,490	1,991	2,126	1,626	1,904	3,022	3,120	3,024	3,120	73%
Makespan (sec)	7,156	7,615	7,098	6,916	5,480	4,994	5,927	4,942	5,727	6,413	5,849	5,420	8,400	6,324	6,496	6,324	29%
Tardiness (sec)	364,509	329,835	382,021	505,384	121,246	139,218	107,480	95,551	145,702	159,183	109,208	137,048	248,765	258,582	249,037	258,582	81%
Earliness/Tardiness (sec)	364,509	329,835	382,021	505,384	121,246	139,218	107,480	95,551	145,702	159,183	109,208	137,048	248,765	258,582	249,037	258,582	81%
Number of tardy jobs (jobs)	100	100	100	100	80	80	80	80	80	80	80	80	80	80	80	80	20%
Fill rate (%)	0	0	0	0	20	20	20	20	20	20	20	20	20	20	20	20	-
Average machine utilisation (%)	75	70	75	77	84	90	80	91	84	78	84	89	61	72	74	72	14%
WIP level (jobs)	58	50	61	81	35	97	45	110	69	101	117	139	13	18	18	18	75%
Total time in queue (sec)	364,509	329,835	382,021	505,384	153,542	463,937	226,845	528,088	364,770	623,607	667,691	739,824	56,517	70,912	68,831	70,912	83%

Performance Metric	Best Performance			
	LA26		LA30	
	Rule	Mode	Rule	Mode
Total throughput time (sec)	FCFS/LATE	Kanban	FCFS/LATE	Kanban
Average flow time (sec)	FCFS/LATE	Kanban	FCFS/LATE	Kanban
Makespan (sec)	FCFS/LATE	Kanban	FCFS/LATE	Kanban
Tardiness (sec)	FCFS/LATE	Kanban	FCFS/LATE	Kanban
Earliness/Tardiness (sec)	FCFS/LATE	Kanban	FCFS/LATE	Kanban
Number of tardy jobs (jobs)	All rules	All pull	All rules	All pull
Fill rate (%)	All rules	Push	All rules	Push
Average machine utilisation (%)	SIPT	CONWIP	SIPT	CONWIP
WIP level (jobs)	SIPT	CONWIP	SIPT	CONWIP
Total time in queue (sec)	SIPT	CONWIP	SIPT	CONWIP

**Key:**

- Most inferior performance
- Most superior performance
- Best push performance used to assess performance differential with best performing pull mode



**Table 6.9 Performance output for LA31 (30x10) 150 job instances**

Performance Metric	Push				Kanban				Base stock				CONWIP				Pull adoption performance differential
	SIPT	EDD	WINQ	FCFS/LATE	SIPT	EDD	WINQ	FCFS/LATE	SIPT	EDD	WINQ	FCFS/LATE	SIPT	EDD	WINQ	FCFS/LATE	
Total throughput time (sec)	665,152	554,199	776,256	1,016,464	230,630	318,621	198,308	201,807	297,378	345,625	234,727	306,168	495,185	506,199	502,224	506,199	80%
Average flow time (sec)	5,663	5,088	6,021	7,333	2,044	2,631	1,828	1,852	2,489	2,811	2,071	2,547	3,808	3,881	3,855	3,881	75%
Makespan (sec)	9,193	10,041	8,995	9,675	7,568	6,759	7,403	6,644	7,796	8,247	8,803	7,136	10,003	8,108	8,875	8,108	34%
Tardiness (sec)	773,525	687,218	827,156	1,023,999	230,630	318,621	198,308	201,807	297,378	345,625	234,727	306,168	495,185	506,199	502,224	506,199	81%
Earliness/Tardiness (sec)	773,525	687,218	827,156	1,023,999	230,630	318,621	198,308	201,807	297,378	345,625	234,727	306,168	495,185	506,199	502,224	506,199	81%
Number of tardy jobs (jobs)	150	150	150	150	120	120	120	120	120	120	120	120	120	120	120	120	20%
Fill rate (%)	0	0	0	0	20	20	20	20	20	20	20	20	20	20	20	20	-
Average machine utilisation (%)	83	76	84	79	88	94	89	94	88	87	82	93	72	82	80	82	4%
WIP level (jobs)	92	76	100	114	55	194	91	207	116	195	175	242	20	28	27	28	74%
Total time in queue (sec)	773,525	687,218	827,156	1,023,999	362,504	1,317,554	640,777	1,384,770	875,120	1,605,578	1,529,064	1,742,137	128,975	169,034	172,153	169,034	81%

**Table 6.10 Performance output for LA35 (30x10) 150 job instances**

Performance Metric	Push				Kanban				Base stock				CONWIP				Pull adoption performance differential
	SIPT	EDD	WINQ	FCFS/LATE	SIPT	EDD	WINQ	FCFS/LATE	SIPT	EDD	WINQ	FCFS/LATE	SIPT	EDD	WINQ	FCFS/LATE	
Total throughput time (sec)	696,349	539,957	747,606	1,064,837	297,380	332,502	238,173	228,269	346,618	357,461	278,409	315,187	513,111	529,874	511,725	529,874	79%
Average flow time (sec)	5,766	5,102	5,961	7,539	2,499	2,733	2,104	2,038	2,827	2,899	2,372	2,617	3,937	4,049	3,928	4,049	73%
Makespan (sec)	9,658	10,085	9,801	9,834	9,009	7,381	8,076	7,240	9,096	8,323	8,088	7,259	9,900	8,504	8,904	8,504	28%
Tardiness (sec)	787,463	687,890	816,740	1,053,367	297,380	332,502	238,173	228,269	346,618	357,461	278,409	315,187	513,111	529,874	511,725	529,874	78%
Earliness/Tardiness (sec)	787,463	687,890	816,740	1,053,367	297,380	332,502	238,173	228,269	346,618	357,461	278,409	315,187	513,111	529,874	511,725	529,874	78%
Number of tardy jobs (jobs)	150	150	150	150	120	120	120	120	120	120	120	120	120	120	120	120	20%
Fill rate (%)	0	0	0	0	20	20	20	20	20	20	20	20	20	20	20	20	-
Average machine utilisation (%)	80	77	79	79	80	91	85	92	81	86	87	93	75	79	79	79	3%
WIP level (jobs)	90	76	91	115	47	179	74	190	94	190	169	232	20	28	27	28	74%
Total time in queue (sec)	787,463	687,890	816,740	1,053,367	368,635	1,316,449	552,371	1,374,966	813,965	1,575,817	1,355,041	1,697,159	128,358	179,445	177,330	179,445	81%

Performance Metric	Best Performance			
	LA31		LA35	
	Rule	Mode	Rule	Mode
Total throughput time (sec)	WINQ	Kanban	FCFS/LATE	Kanban
Average flow time (sec)	WINQ	Kanban	FCFS/LATE	Kanban
Makespan (sec)	FCFS/LATE	Kanban	FCFS/LATE	Kanban
Tardiness (sec)	WINQ	Kanban	FCFS/LATE	Kanban
Earliness/Tardiness (sec)	WINQ	Kanban	FCFS/LATE	Kanban
Number of tardy jobs (jobs)	All rules	All pull	All rules	All pull
Fill rate (%)	All rules	Push	All rules	Push
Average machine utilisation (%)	SIPT	CONWIP	SIPT	CONWIP
WIP level (jobs)	SIPT	CONWIP	SIPT	CONWIP
Total time in queue (sec)	SIPT	CONWIP	SIPT	CONWIP

**Key:**  
■ Most inferior performance  
■ Most superior performance  
■ Best push performance used to assess performance differential with best performing pull mode

Table 6.11 Performance output for SWV15 (50x10) 250 job instances

Performance Metric	Push				Kanban				Base stock				CONWIP				Pull adoption performance differential
	SIPT	EDD	WINQ	FCFS/LATE	SIPT	EDD	WINQ	FCFS/LATE	SIPT	EDD	WINQ	FCFS/LATE	SIPT	EDD	WINQ	FCFS/LATE	
Total throughput time (sec)	1,428,325	956,352	2,603,249	3,967,214	803,001	920,476	723,756	750,261	820,225	962,260	761,134	856,368	1,314,640	1,454,966	1,391,792	1,454,966	82%
Average flow time (sec)	8,810	8,443	12,706	17,530	3,712	4,182	3,395	3,501	3,781	4,349	3,544	3,925	5,758	6,320	6,067	6,320	81%
Makespan (sec)	15,634	16,948	19,629	22,386	13,515	11,230	11,986	10,904	13,539	12,864	12,926	10,976	14,400	13,421	15,842	13,421	51%
Tardiness (sec)	2,077,468	1,985,860	3,051,475	4,257,659	803,001	920,476	723,756	750,261	820,225	962,260	761,134	856,368	1,314,640	1,454,966	1,391,792	1,454,966	83%
Earliness/Tardiness (sec)	2,077,468	1,985,860	3,051,475	4,257,659	803,001	920,476	723,756	750,261	820,225	962,260	761,134	856,368	1,314,640	1,454,966	1,391,792	1,454,966	83%
Number of tardy jobs (jobs)	250	250	250	250	200	200	200	200	200	200	200	200	200	200	200	200	20%
Fill rate (%)	0	0	0	0	20	20	20	20	20	20	20	20	20	20	20	20	-
Average machine utilisation (%)	80	74	64	56	89	97	91	97	89	93	92	97	84	84	77	84	-
WIP level (jobs)	141	125	162	196	76	352	113	354	180	359	321	440	34	48	38	48	73%
Total time in queue (sec)	2,077,468	1,985,860	3,051,475	4,257,659	922,396	3,957,711	1,281,554	3,866,017	2,365,666	4,615,754	4,130,979	4,864,498	377,270	539,980	496,282	539,980	81%

Table 6.12 Performance output for SWV20 (50x10) 250 job instances

Performance Metric	Push				Kanban SL=20%				Base stock				CONWIP				Pull adoption performance differential
	SIPT	EDD	WINQ	FCFS/LATE	SIPT	EDD	WINQ	FCFS/LATE	SIPT	EDD	WINQ	FCFS/LATE	SIPT	EDD	WINQ	FCFS/LATE	
Total throughput time (sec)	1,509,795	1,057,388	2,133,524	2,716,331	689,470	892,224	572,851	647,289	816,531	939,238	676,893	854,117	1,268,282	1,273,033	1,338,509	1,273,033	79%
Average flow time (sec)	8,469	7,282	9,967	11,653	3,250	4,061	2,784	3,082	3,758	4,249	3,200	3,909	5,565	5,584	5,846	5,584	76%
Makespan (sec)	14,491	14,764	14,164	14,127	13,043	11,375	12,008	10,921	13,043	12,530	12,539	11,329	15,356	12,333	14,399	12,333	29%
Tardiness (sec)	1,994,122	1,697,461	2,368,636	2,790,126	689,470	892,224	572,851	647,289	816,531	939,238	676,893	854,117	1,268,282	1,273,033	1,338,509	1,273,033	79%
Earliness/Tardiness (sec)	1,994,122	1,697,461	2,368,636	2,790,126	689,470	892,224	572,851	647,289	816,531	939,238	676,893	854,117	1,268,282	1,273,033	1,338,509	1,273,033	79%
Number of tardy jobs (jobs)	250	250	250	250	200	200	200	200	200	200	200	200	200	200	200	200	20%
Fill rate (%)	0	0	0	0	20	20	20	20	20	20	20	20	20	20	20	20	-
Average machine utilisation (%)	85	83	87	87	90	97	92	97	90	93	93	97	78	89	82	89	6%
WIP level (jobs)	146	123	176	206	77	359	142	381	183	369	349	435	31	48	39	48	75%
Total time in queue (sec)	1,994,122	1,697,461	2,368,636	2,790,126	911,283	4,077,781	1,640,599	4,163,858	2,319,795	4,619,137	4,357,211	4,940,775	360,139	489,747	458,157	489,747	79%

Performance Metric	Best Performance			
	SWV15		SWV20	
	Rule	Mode	Rule	Mode
Total throughput time (sec)	WINQ	Kanban	WINQ	Kanban
Average flow time (sec)	WINQ	Kanban	WINQ	Kanban
Makespan (sec)	FCFS/LATE	Kanban	FCFS/LATE	Kanban
Tardiness (sec)	WINQ	Kanban	WINQ	Kanban
Earliness/Tardiness (sec)	WINQ	Kanban	WINQ	Kanban
Number of tardy jobs (jobs)	All rules	All pull	All rules	All pull
Fill rate (%)	All rules	Push	All rules	Push
Average machine utilisation (%)	FCFS/LATE	Push	SIPT	CONWIP
WIP level (jobs)	SIPT	CONWIP	SIPT	CONWIP
Total time in queue (sec)	SIPT	CONWIP	SIPT	CONWIP

Key:  
 Most inferior performance  
 Most superior performance  
 Best push performance used to assess performance differential with best performing pull mode

In the majority of the static experiments, the most inferior performance in terms of the above criteria is observed when push control is implemented. However, in four out of the ten static experiments, that is, LA16, LA20, LA26 and LA30 it is the CONWIP mechanism that results in the worst makespan performance.

2. The best results in terms of WIP level in the system and the total time jobs spend queuing in input buffers is observed under CONWIP control across all ten experiments. CONWIP results in the lowest machine utilisation levels in all the ten static experiments except for SWV15 where it is outperformed by push control. The average performance differential for each of these three criteria is as follows:
  - Machine utilisation (8%)
  - WIP level (73%)
  - Total time in queue (82%)
3. The second best performance in terms of average machine utilisation, WIP level and time in queue is observed when the system operates under the closest alternative to CONWIP, i.e. the push system. Under both types of production control, jobs are pushed through the shop-floor according to their routings. As a result, machines responsible for carrying out the initial processing steps experience the lengthiest queues and highest WIP and machine utilisation levels. By contrast, when the Kanban mechanism is employed there is a greater dispersion of jobs and therefore the impact of job release is much wider. Although in the case of Kanban this dispersion cascades progressively to upstream production stages, under Base Stock control the dispersion is immediate. Therefore, the underlying reasons for the poor performance of Kanban and Base Stock in terms of these three shop-floor related performance criteria can be expected on the basis of the operating principles of these two mechanisms.
4. In all the ten static experiments, the Kanban mechanism combined with SIPT outperforms the push system regardless of the dispatching rule implemented. This is the case for all the performance metrics with the exception of average machine utilisation.
5. The obtained results verify the synergistic effect of control mechanism and selected dispatching rule on job-shop performance. The best performance with regards to six out of the nine criteria considered which relate to job flow time and tardiness performance is achieved using either the WINQ or FCFS/LATE

dispatching rules. These two rules seek to prioritise jobs by utilising shop-floor (queues in workstations downstream) and workstation data (requests for inventory logged with the workstation's output buffer) respectively. Consequently, they do facilitate dispatching decisions made from a much wider perspective. On the other hand, SIPT and EDD merely utilise processing time and due-date data for the jobs queuing in the workstation's input buffer. They therefore tend to be rather myopic. In addition, given that the static experiments involve multiple job orders with similar due-dates SIPT and EDD make little distinction between them.

With reference to the dynamic experiments, the results are shown in Tables 6.13-17. The first observation that needs to be made is that the initial level of inventory provided at system initialisation resulted in a higher fill rate than what was estimated in the experimental design phase. More specifically, fill rate levels ranged between 52-61% across the five sets of dynamic experiments. In addition, as set-up times were considered in this case, the evaluation of the job-shop's performance in the dynamic experiments is carried out across 10 criteria overall, including the total set-up time. Careful consideration of the computational results suggests that:

1. As was the case in the static experiments, the best performance in terms of the six job flow time and tardiness related performance criteria is achieved using one of the pull control mechanisms. However, in this case it is evident that Kanban failed to sustain its prevalence and was outperformed by Base Stock. The average performance differential that can be achieved across these six measures by replacing the traditional push control system with pull control exercised by the Base Stock mechanism is presented in brackets below:
  - Total throughput time (94%)
  - Average flow time (93%)
  - Makespan (95%)
  - Tardiness (95%)
  - Combined earliness/tardiness (95%)
  - Number of tardy jobs (56%)

The superior performance of Base Stock overall but mainly compared to that of Kanban is intriguing. Kanban resulted in the best job-shop performance in terms of the above criteria in the case of the static experiments which involved multiple orders of all the job types visiting the job-shop.

Table 6.13 Performance output for LA40 (15x15) 31 job instances including rush orders

Performance Metric	Push				Kanban				Base stock				CONWIP				Pull adoption performance differential
	LTPT	MS	LOPNR	RL	LTPT	MS	LOPNR	RL	LTPT	MS	LOPNR	RL	LTPT	MS	LOPNR	RL	
Total throughput time (sec)	17,692,405	18,290,985	17,895,647	20,977,966	1,757,706	1,814,090	1,338,957	1,425,533	1,778,301	1,759,695	830,381	1,629,004	4,951,768	4,556,064	4,936,314	4,741,915	96%
Average flow time (sec)	412,983	425,217	426,493	476,802	57,658	59,477	44,150	46,943	58,322	57,722	27,744	53,506	160,692	147,928	160,194	153,923	94%
Makespan (sec)	615,692	619,824	741,260	686,030	189,103	203,005	226,918	173,186	201,227	197,364	85,726	184,886	562,181	503,674	492,628	537,607	88%
Tardiness (sec)	20,185,947	19,516,099	20,422,884	22,364,874	1,757,706	1,814,090	1,338,957	1,425,533	1,778,301	1,759,695	830,381	1,629,004	4,951,768	4,556,064	4,936,314	4,741,915	96%
Earliness/Tardiness (sec)	20,185,947	19,516,099	20,422,884	22,364,874	1,757,706	1,814,090	1,338,957	1,425,533	1,778,301	1,759,695	830,381	1,629,004	4,951,768	4,556,064	4,936,314	4,741,915	96%
Number of tardy jobs (jobs)	31	31	31	31	15	15	15	15	15	15	15	15	15	15	15	15	52%
Fill rate (%)	0	0	0	0	52	52	52	52	52	52	52	52	52	52	52	52	-
Average machine utilisation (%)	53	52	46	50	97	97	95	97	97	97	100	98	59	61	58	57	-
WIP level (jobs)	32	31	27	32	282	275	273	240	294	297	324	263	33	34	28	34	-
Total time in queue (sec)	13,677,066	13,129,902	13,913,901	16,168,464	50,644,337	52,842,336	58,198,516	39,002,881	56,172,300	55,740,571	26,504,012	45,887,892	11,863,063	10,825,932	8,160,797	12,266,043	40%
Total set-up time (sec)	11,321	12,123	11,913	8,579	10,968	12,875	15,604	7,362	11,140	11,070	4,348	7,430	17,445	16,710	15,094	13,961	49%

**Best Performance**  
LA40

Performance Metric	Rule	Mode
Total throughput time (sec)	LOPNR	Base stock
Average flow time (sec)	LOPNR	Base stock
Makespan (sec)	LOPNR	Base stock
Tardiness (sec)	LOPNR	Base stock
Earliness/Tardiness (sec)	LOPNR	Base stock
Number of tardy jobs (jobs)	All rules	All pull
Fill rate (%)	All rules	Push
Average machine utilisation (%)	LOPNR	Push
WIP level (jobs)	LOPNR	Push
Total time in queue (sec)	LOPNR	CONWIP
Total set-up time (sec)	LOPNR	Base stock

**Key:**

- Most inferior performance
- Most superior performance
- Best push performance used to assess performance differential with best performing pull mode

Table 6.14 Performance output for ABZ7 (20x15) 38 Instances including rush orders

Performance Metric	Push				Kanban				Base stock				CONWIP			
	LTPT	MS	LOPNR	RL	LTPT	MS	LOPNR	RL	LTPT	MS	LOPNR	RL	LTPT	MS	LOPNR	RL
Total throughput time (sec)	5,590,490	5,670,923	3,965,153	5,817,843	758,246	944,837	892,686	785,610	760,820	780,328	445,886	812,335	1,802,145	1,964,007	1,835,025	2,003,591
Average flow time (sec)	142,817	143,531	123,439	150,322	20,385	25,295	23,923	21,105	20,453	20,966	12,165	21,808	47,856	52,116	48,721	53,157
Makespan (sec)	201,990	193,874	207,488	200,307	96,821	126,018	117,763	94,564	96,821	114,169	54,732	97,224	177,849	185,617	195,796	197,695
Tardiness (sec)	6,122,690	6,009,705	4,674,286	6,145,147	758,246	944,837	892,686	785,610	760,820	780,328	445,886	812,335	1,802,145	1,964,007	1,835,025	2,003,591
Earliness/Tardiness (sec)	6,122,690	6,009,705	4,674,286	6,145,147	758,246	944,837	892,686	785,610	760,820	780,328	445,886	812,335	1,802,145	1,964,007	1,835,025	2,003,591
Number of tardy jobs (jobs)	38	38	38	38	15	15	15	15	15	15	15	15	15	15	15	15
Fill rate (%)	0	0	0	0	61	61	61	61	61	61	61	61	61	61	61	61
Average machine utilisation (%)	64	66	61	64	99	94	96	100	99	98	99	100	66	63	62	60
WIP level (jobs)	30	31	23	31	412	309	670	353	421	399	416	362	27	25	23	23
Total time in queue (sec)	3,932,016	3,857,428	2,722,541	3,998,924	38,544,987	37,209,180	74,998,616	32,025,738	39,381,071	43,921,851	22,054,554	33,812,408	2,988,733	2,885,168	2,656,329	2,865,816
Total set-up time (sec)	6,054	6,111	6,259	4,773	5,252	8,014	7,783	3,510	5,224	6,669	2,914	3,438	8,159	8,038	7,895	7,083

Pull adoption performance differential

92%  
92%  
74%  
93%  
93%  
61%  
-  
3%  
-  
32%  
39%

Table 6.15 Performance output for ABZ9(20x15) 38 Instances including rush orders

Performance Metric	Push				Kanban				Base stock				CONWIP			
	LTPT	MS	LOPNR	RL	LTPT	MS	LOPNR	RL	LTPT	MS	LOPNR	RL	LTPT	MS	LOPNR	RL
Total throughput time (sec)	6,067,529	6,214,812	5,021,333	6,397,272	936,044	1,029,692	903,290	1,050,613	957,102	952,281	441,748	1,059,291	1,994,009	2,417,060	1,761,950	2,326,635
Average flow time (sec)	148,154	144,598	129,154	147,185	25,104	27,569	24,242	28,119	25,658	25,531	12,096	28,348	52,945	64,078	46,839	61,699
Makespan (sec)	225,528	238,408	259,069	226,709	100,275	106,368	113,600	122,543	102,915	100,156	45,192	122,516	185,798	212,480	176,860	257,721
Tardiness (sec)	6,645,093	6,485,649	5,921,139	6,749,603	936,044	1,029,692	903,290	1,050,613	957,102	952,281	441,748	1,059,291	1,994,009	2,417,060	1,761,950	2,326,635
Earliness/Tardiness (sec)	6,645,093	6,485,649	5,921,139	6,749,603	936,044	1,029,692	903,290	1,050,613	957,102	952,281	441,748	1,059,291	1,994,009	2,417,060	1,761,950	2,326,635
Number of tardy jobs (jobs)	38	38	38	38	16	16	16	16	16	16	16	16	16	16	16	16
Fill rate (%)	0	0	0	0	58	58	58	58	58	58	58	58	58	58	58	58
Average machine utilisation (%)	61	57	53	60	99	99	97	95	99	99	99	96	68	61	70	53
WIP level (jobs)	29	27	23	30	360	314	268	249	373	375	393	266	33	30	33	24
Total time in queue (sec)	4,255,020	4,145,966	3,601,758	4,331,558	34,722,266	31,900,810	28,719,208	28,791,786	36,996,442	36,115,181	17,169,292	30,939,400	3,872,614	4,127,462	3,502,641	3,714,684
Total set-up time (sec)	7,639	7,390	6,842	5,924	7,906	9,251	9,902	6,393	8,031	7,814	3,528	5,827	10,400	9,665	9,193	8,298

Pull adoption performance differential

93%  
92%  
83%  
93%  
93%  
58%  
-  
-  
-  
3%  
40%

Best Performance

Performance Metric	ABZ7		ABZ9	
	Rule	Mode	Rule	Mode
Total throughput time (sec)	LOPNR	Base stock	LOPNR	Base stock
Average flow time (sec)	LOPNR	Base stock	LOPNR	Base stock
Makespan (sec)	LOPNR	Base stock	LOPNR	Base stock
Tardiness (sec)	LOPNR	Base stock	LOPNR	Base stock
Earliness/Tardiness (sec)	LOPNR	Base stock	LOPNR	Base stock
Number of tardy jobs (jobs)	All rules	All pull	All rules	All pull
Fill rate (%)	All rules	Push	All rules	Push
Average machine utilisation (%)	RL	CONWIP	RL	CONWIP
WIP level (jobs)	LOPNR	Push	LOPNR	Push
Total time in queue (sec)	LOPNR	CONWIP	LOPNR	CONWIP
Total set-up time (sec)	LOPNR	Base stock	LOPNR	Base stock

Key:

- Most inferior performance
- Most superior performance
- Best push performance used to assess performance differential with best performing pull mode

Table 6.16 Performance output for YN1 (20x20) 38 Instances including rush orders

Performance Metric	Push				Kanban				Base stock				CONWIP				Pull adoption performance differential
	LTPT	MS	LOPNR	RL	LTPT	MS	LOPNR	RL	LTPT	MS	LOPNR	RL	LTPT	MS	LOPNR	RL	
Total throughput time (sec)	8,964,859	9,253,597	9,092,756	10,339,304	1,186,097	1,156,451	1,042,987	1,144,517	1,199,601	1,223,066	538,331	1,202,368	2,651,577	2,773,750	2,767,588	2,845,077	95%
Average flow time (sec)	189,959	185,120	179,647	193,858	31,918	31,138	28,152	30,824	32,273	32,891	14,871	32,346	70,483	73,698	73,536	75,575	92%
Makespan (sec)	274,805	271,136	318,807	282,193	121,569	126,073	143,187	126,383	124,441	141,052	56,706	126,383	241,196	249,637	291,282	269,653	82%
Tardiness (sec)	9,272,790	9,591,414	9,482,203	10,625,076	1,186,097	1,156,451	1,042,987	1,144,517	1,199,601	1,223,066	538,331	1,202,368	2,651,577	2,773,750	2,767,588	2,845,077	95%
Earliness/Tardiness (sec)	9,272,790	9,591,414	9,482,203	10,625,076	1,186,097	1,156,451	1,042,987	1,144,517	1,199,601	1,223,066	538,331	1,202,368	2,651,577	2,773,750	2,767,588	2,845,077	95%
Number of tardy jobs (jobs)	38	38	38	38	16	16	16	16	16	16	16	16	16	16	16	16	58%
Fill rate (%)	0	0	0	0	58	58	58	58	58	58	58	58	58	58	58	58	-
Average machine utilisation (%)	60	62	54	60	99	99	95	98	99	97	99	99	59	59	53	57	2%
WIP level (jobs)	34	35	30	38	483	457	358	414	498	465	554	429	29	31	27	27	8%
Total time in queue (sec)	5,253,024	5,446,826	5,161,213	6,254,005	56,336,082	55,146,696	48,495,426	49,863,128	59,520,362	62,942,802	30,341,615	51,766,445	3,868,644	4,415,635	4,345,337	4,083,505	25%
Total set-up time (sec)	11,400	11,118	10,712	9,194	11,983	13,765	14,260	8,415	12,069	14,459	4,869	7,992	15,536	15,126	14,686	14,648	47%

Table 6.17 Performance output for YN4 (20x20) 38 Instances including rush orders

Performance Metric	Push				Kanban				Base stock				CONWIP				Pull adoption performance differential
	LTPT	MS	LOPNR	RL	LTPT	MS	LOPNR	RL	LTPT	MS	LOPNR	RL	LTPT	MS	LOPNR	RL	
Total throughput time (sec)	16,121,194	16,025,837	14,909,230	17,562,619	1,394,014	1,478,702	1,196,268	1,110,384	1,394,014	1,394,014	647,860	1,149,482	3,857,059	4,114,936	3,679,913	3,964,552	96%
Average flow time (sec)	262,867	272,840	259,723	314,593	37,418	39,646	32,214	29,954	37,418	37,418	17,782	30,983	102,235	109,021	97,573	105,064	94%
Makespan (sec)	433,295	360,159	444,433	423,486	142,512	159,834	158,029	157,002	142,512	142,512	84,744	168,314	367,775	351,063	410,922	329,565	81%
Tardiness (sec)	17,566,588	16,597,150	16,576,864	18,202,804	1,394,014	1,478,702	1,196,268	1,110,384	1,394,014	1,394,014	647,860	1,149,482	3,857,059	4,114,936	3,679,913	3,964,552	96%
Earliness/Tardiness (sec)	17,566,588	16,597,150	16,576,864	18,202,804	1,394,014	1,478,702	1,196,268	1,110,384	1,394,014	1,394,014	647,860	1,149,482	3,857,059	4,114,936	3,679,913	3,964,552	96%
Number of tardy jobs (jobs)	38	38	38	38	18	18	18	18	18	18	18	18	18	18	18	18	53%
Fill rate (%)	0	0	0	0	53	53	53	53	53	53	53	53	53	53	53	53	-
Average machine utilisation (%)	54	58	50	53	99	96	95	97	99	99	99	94	58	61	52	60	-
WIP level (jobs)	40	46	37	43	473	409	362	358	490	490	505	378	40	47	33	49	11%
Total time in queue (sec)	11,070,209	10,810,078	10,481,279	12,697,388	64,634,767	62,289,809	53,892,751	53,192,265	66,987,319	66,990,685	41,154,688	60,514,044	9,111,552	10,828,458	8,556,837	10,910,781	18%
Total set-up time (sec)	12,221	11,988	11,298	9,701	13,144	15,972	14,637	9,987	12,933	12,966	7,176	10,210	16,585	16,166	13,698	11,661	26%

Best Performance

Performance Metric	YN1		YN4	
	Rule	Mode	Rule	Mode
Total throughput time (sec)	LOPNR	Base stock	LOPNR	Base stock
Average flow time (sec)	LOPNR	Base stock	LOPNR	Base stock
Makespan (sec)	LOPNR	Base stock	LOPNR	Base stock
Tardiness (sec)	LOPNR	Base stock	LOPNR	Base stock
Earliness/Tardiness (sec)	LOPNR	Base stock	LOPNR	Base stock
Number of tardy jobs (jobs)	All rules	All pull	All rules	All pull
Fill rate (%)	All rules	Push	All rules	Push
Average machine utilisation (%)	LOPNR	CONWIP	LOPNR	Push
WIP level (jobs)	RL	CONWIP	LOPNR	CONWIP
Total time in queue (sec)	LTPT	CONWIP	LOPNR	CONWIP
Total set-up time (sec)	LOPNR	Base stock	LOPNR	Base stock

Key:

- Most inferior performance
- Most superior performance
- Best push performance used to assess performance differential with best performing pull mode

Another extreme assumption made in the static experiments was that multiple orders of the same job type had the same due-date.

When the Base Stock mechanism was employed in the static experiments, the WIP replenishment process was triggered instantaneously in all the upstream stages for numerous orders. The impact on all the workstations across the system was enormous. As demonstrated by the performance statistics related to time in queue and WIP levels, in the case of Base Stock these were more than double compared to the equivalent results obtained when Kanban is used. The SWV15 experiment is a typical case in point.

When the WINQ rule is selected, the WIP level for Kanban is 113 jobs compared to 321 jobs in the case of Base Stock and total time in queue is 1,28 (million seconds) and 4,13 (million seconds) respectively. These findings suggest that the specific job mix in the static experiments hindered Base Stock's capability to replenish WIP and fulfil outstanding orders swiftly. In the dynamic experiments however, greater demand (order) diversity means that the replenishment process is instigated simultaneously in less workstations upstream thus improving Base Stock's performance in terms of job flow time and tardiness.

2. The Base Stock mechanism also delivers the best performance in terms of total time spent configuring the machines in the entire set of dynamic experiments.
3. In relation to the remaining three shop-floor performance metrics, i.e. average machine utilisation, WIP level and time in queue, no specific pattern can be observed. The best performance in terms of each of these three criteria is delivered by either push control or the push/pull hybrid CONWIP.
4. Overall, the most superior performance across the entire range of the ten criteria considered is delivered by the LOPNR rule which similarly to the best performing WINQ rule in the static experiments considers the conditions prevailing globally within the job-shop to prioritise jobs. However, there are some exceptions, e.g. experiments ABZ7 and ABZ9 in which the best average machine utilisation performance is delivered by COMWIP and the RL rule.



### 6.1.3 Sensitivity analysis

The five dynamic problems that is, LA40, ABZ7, ABZ9, YN1 and YN4 are modified to create a new set of experiments used to allow sensitivity analysis to be performed and assess the system's performance in significantly less favourable conditions involving:

- Damage to WIP resulting from all machine faults (the damage affects the portion of the batch which is processed when the machine develops the fault),
- Initial inventory level in pull control is lowered to 50 units for each job type. This practically means that the initial inventory used to fill the system at system initialisation is only 20% of the equivalent inventory level used in the original dynamic experiments.

In this final set of experiments, the job-shop's operation is tested under push, Kanban, Base Stock and CONWIP control by employing the dispatching rules which in the initial dynamic experiments led to the worst and best performance in the case of push control and the pull mechanisms respectively. More specifically, push is tested by applying the RL dispatching rule whereas the LOPNR rule is employed in all the experiments involving a pull mechanism.

The computational results obtained from the modified dynamic experiments are presented in Tables 6.18-22. From these results, it can be concluded that:

1. The batch-splitting behaviour modelled in BASS in the case of Kanban, Base Stock and CONWIP is implemented successfully allowing the job-shop to cope with high levels of demand with very low levels of initial inventory. For example, from reference to the stochastic data for experiment YN4 included in Appendix M, it can be observed that demand for certain job types can be quite high. One of these is job type 3 for which the system receives the following order pattern over time:
  - t=42 sec, batch size=170 units,
  - t=409 sec, batch size=60 units (rush order)
  - t=921 sec, batch size=120 units,
  - t=1,044 sec, batch size=140 units,
  - t=1,080 sec, batch size=80 units,
  - t=1,136 sec, batch size=160 units,
  - t=1,211 sec, batch size=110 units.

**Table 6.18 Performance output for LA40 (15x15) 31 job instances**

Performance Metric	Push	Kanban	Base stock	CONWIP
	RL	LOPNR	LOPNR	LOPNR
Total throughput time (sec)	20,977,948	5,096,654	3,634,156	11,254,200
Average flow time (sec)	476,802	165,366	118,189	363,997
Makespan (sec)	686,030	311,395	246,574	754,231
Tardiness (sec)	22,364,874	5,096,654	3,634,156	11,254,200
Earliness/Tardiness (sec)	22,364,874	5,096,654	3,634,156	11,254,200
Number of tardy jobs (jobs)	31	31	31	31
Fill rate (%)	0	0	0	0
Average machine utilisation (%)	50	79	85	46
WIP level (jobs)	33	61	168	24
Total time in queue (sec)	16,703,031	15,214,062	38,318,956	12,538,471
Total set-up time (sec)	8,579	43,854	32,078	37,553

Pull adoption performance differential

83%  
75%  
64%  
84%  
84%  
-  
-  
7%  
26%  
25%  
-

**Table 6.19 Performance output for ABZ7 (20x15) 38 job instances**

Performance Metric	Push	Kanban	Base stock	CONWIP
	RL	LOPNR	LOPNR	LOPNR
Total throughput time (sec)	5,777,267	2,654,725	1,582,502	4,244,193
Average flow time (sec)	149,331	70,292	42,076	112,120
Makespan (sec)	200,334	137,012	93,234	244,535
Tardiness (sec)	6,104,546	2,654,725	1,582,502	4,244,193
Earliness/Tardiness (sec)	6,104,546	2,654,725	1,582,502	4,244,193
Number of tardy jobs (jobs)	38	38	38	38
Fill rate (%)	0	0	0	0
Average machine utilisation (%)	64	88	96	50
WIP level (jobs)	32	98	245	16
Total time in queue (sec)	4,346,653	11,495,240	21,527,354	2,125,674
Total set-up time (sec)	4,981	18,620	13,281	17,585

Pull adoption performance differential

73%  
72%  
53%  
74%  
74%  
-  
-  
22%  
50%  
51%  
-

**Table 6.20 Performance output for ABZ9 (20x15) 38 job instances**

Performance Metric	Push	Kanban	Base stock	CONWIP
	RL	LOPNR	LOPNR	LOPNR
Total throughput time (sec)	6,554,752	2,728,420	1,857,175	4,774,655
Average flow time (sec)	150,682	72,272	49,344	126,120
Makespan (sec)	226,388	143,804	119,431	277,310
Tardiness (sec)	6,910,287	2,728,420	1,857,175	4,774,655
Earliness/Tardiness (sec)	6,910,287	2,728,420	1,857,175	4,774,655
Number of tardy jobs (jobs)	38	37	37	37
Fill rate (%)	0	3	3	3
Average machine utilisation (%)	59	86	92	47
WIP level (jobs)	33	167	207	20
Total time in queue (sec)	5,189,036	21,746,970	23,064,095	3,154,807
Total set-up time (sec)	5,704	25,811	18,519	24,076

Pull adoption performance differential

72%  
67%  
47%  
73%  
73%  
3%  
-  
20%  
39%  
39%  
-

**Table 6.21 Performance output for YN1 (20x20) 38 job instances**

Performance Metric	Push	Kanban	Base stock	CONWIP
	RL	LOPNR	LOPNR	LOPNR
Total throughput time (sec)	11,362,830	3,211,089	1,768,793	6,234,144
Average flow time (sec)	213,469	85,207	47,252	164,761
Makespan (sec)	295,833	156,932	101,717	294,569
Tardiness (sec)	11,702,221	3,211,089	1,768,793	6,234,144
Earliness/Tardiness (sec)	11,702,221	3,211,089	1,768,793	6,234,144
Number of tardy jobs (jobs)	38	38	38	38
Fill rate (%)	0	0	0	0
Average machine utilisation (%)	59	89	96	58
WIP level (jobs)	45	144	360	39
Total time in queue (sec)	8,709,191	19,693,353	34,693,488	7,045,893
Total set-up time (sec)	8,811	37,748	22,770	40,883

Pull adoption performance differential

84%  
78%  
66%  
85%  
85%  
-  
-  
2%  
2%  
12%  
-

**Table 6.22 Performance output for YN4 (20x20) 38 job instances**

Performance Metric	Push	Kanban	Base stock	CONWIP
	RL	LOPNR	LOPNR	LOPNR
Total throughput time (sec)	17,626,422	3,717,784	2,415,325	7,824,217
Average flow time (sec)	319,234	98,570	64,294	206,634
Makespan (sec)	430,663	195,436	151,408	477,526
Tardiness (sec)	18,154,759	3,717,784	2,415,325	7,824,217
Earliness/Tardiness (sec)	18,154,759	3,717,784	2,415,325	7,824,217
Number of tardy jobs (jobs)	38	38	38	38
Fill rate (%)	0	0	0	0
Average machine utilisation (%)	53	82	94	39
WIP level (jobs)	47	117	526	21
Total time in queue (sec)	14,304,594	19,386,290	76,750,737	5,618,772
Total set-up time (sec)	9,590	39,569	31,567	40,063

Pull adoption performance differential

86%  
80%  
65%  
87%  
87%  
-  
-  
25%  
55%  
61%  
-

Key:

- Most inferior performance
- Most superior performance
- Best push performance used to assess performance differential with best performing pull mode

The above equate to a total demand for 840 units in the production period considered which corresponds to merely 6% of the initial inventory available when the system is initialised. Yet, the job-shop succeeds in using this inventory to fill some of the demand and trigger production to replenish its stock and fulfil any remaining outstanding demand. This finding is of significant value in the validation of BASS.

2. Owing to the more adverse conditions considered in these experiments, i.e. lower initial inventory level and scrapped WIP caused by faults, all production control modes exhibit worse performance compared to the original dynamic experiments. This is reflected on all the performance statistics collected.
3. Base Stock continues to deliver the best performance in terms of job flow time and tardiness criteria with the average performance differential presented below:
  - Total throughput time (80%)
  - Average flow time (74%)
  - Makespan (59%)
  - Tardiness (81%)
  - Combined earliness/tardiness (81%)
  - Number of tardy jobs (<1%)

These are clearly lower than the equivalent percentages obtained for the original set of dynamic experiments but considering the adversity of the assumptions made in the modified dynamic experiments, these results continue to represent an extremely good performance differential following the adoption of pull control.

4. In all five experiments, the best performance in terms of set-up time is achieved under push control and not Base Stock as it was the case in the original dynamic experiments.
5. CONWIP consistently outperforms all other production control modes in terms of three shop-floor related metrics, that is, machine utilisation, WIP level and time in queue.

## 6.2 Application of pull production control to an industrial case-study

### 6.2.1 Company profile

Further to the extensive experimental testing of BASS, the designed agent-based scheduling system is utilised to investigate the application of pull control in a manufacturing company. The company wishes to remain anonymous and will henceforth be referred to as LCA. LCA is a member of a multinational group that produces components for the aerospace industry. The group has factories in multiple locations across the UK and is a major supplier for Europe's aerospace market. LCA specialises in the fabrication of aircraft fastening systems including ultra-light heavy duty aluminium bolts, screws and rivets produced to high tolerances. A sample of the products manufactured at LCA is illustrated in Figure 6.1.

Table 6.23 presents the list of operations performed to deliver the wider range of LCA's fastening components. These include a number of manufacturing processes such as cold heading where wire is cut to length and head clipping and trimming used to form and tidy the screw head. These are followed by grinding required for the high precision shaping of the screw, thread rolling used to roll the thread into the screw, tail extruding etc. The operations performed also include non-manufacturing processes for instance inspections and testing. Table 6.23 further shows the maximum output rate per machine in each workstation. The workstations performing these operations are arranged in typical job-shop layout.



**Figure 6.1 Aircraft fasteners produced by LCA**

Factory visits and follow-up interviews with LCA's production manager revealed a number of issues which are known to have an adverse impact on the factory's performance. For most of the jobs, the first operation is cold heading. Furthermore, at least 70% of the parts are routed through the two grinding workstations. These

processes are therefore bottlenecks. Most importantly, LCA appears to be suffering from extremely long set-up times which can be several times higher than the total processing times for the majority of its products. Not surprisingly, LCA reported queue lengths at most workstations that range between three to five days. The company considers a lean strategy in order to meet current challenges and increase its competitive advantage.

**Table 6.23 List of operations and workstation data LCA case study**

Workstation Number	Operation description	Ref.	Pieces/hour	Queue time
B04	Cold Heading	HED	3000	0 days
C01	Trim Head	TRM	3000	3 days
C02	Clip Head	CLP	3600	3 days
C04	Extrude Tail	EXT	3000	3 days
D01	Turn Head	TRN	450	3 days
D02	Face and Chamfer (auto)	F/C	4000	3 days
E01	Turn Head (manual)	TRN	200	3 days
E02	Mark Head	MRK	700	3 days
E03	Face and Chamfer (manual)	F/C	600	3 days
E05	Drill	DRL	80	3 days
E06	De-fraze	DEF	240	3 days
E08	Crop to length	CRP	500	3 days
F01	Grind	G/S	350	3 days
F02	Grind	G/S	350	3 days
H01	Thread Roll	T/R	4000	3 days
H03	Fillet Roll	F/R	1800	3 days
T09	CNC Turn	CNC	65	3 days
J01-J02	Inspect	INS-FLW		3 days
K01	Inspect	LAB		3 days
L01-L02-L03	Heat Treat	H/T-H/T-BL		3 days
N01	Plating	PLT		5 days

### 6.2.2 Data, limitations and implementation issues

The historical data supplied by LCA cover a production period of one calendar month. Each production week is 4.5 days (37 hours). The sample comprises 50 different jobs associated with specific part orders which however, concern the same product group. Batch sizes in the sample range between 100-1,500 pieces. Each part order in the dataset visits an average of 12 workstations. Table 6.24 presents the operations layout

**Table 6.24 Operations layout for sample of 10 orders LCA case study**

Part Number	Card	Batch size	Operations															
S107-04-06	519920	500					HED	CLP	EXT	F/C	H/T	BL	T/R	FLW	PLT	LAB	INS	
NAS1303-15	520625	300					HED	TRM	EXT	F/C	H/T	BL	T/R	G/S	FLW	PLT	LAB	INS
NAS6603-3	520840	198					HED	TRM	EXT	F/C	H/T	BL	T/R	FLW	PLT	LAB	INS	
JN0177-0406	521696	300						HED	H/T	T/R	TRN	G/S	TRN	FLW	PLT	LAB	INS	
NAS1100E3-8P	521181	536							HED	F/C	H/T	BL	T/R	FLW	PLT	LAB	INS	
NAS6703H1	521914	600					HED	TRM	F/C	DRL	DEF	H/T	BL	G/S	T/R	FLW	PLT	INS
S105H05-22	521775	150	HED	TRM	EXT	F/C	MRK	DRL	DEF	H/T	BL	T/R	G/S	F/R	FLW	PLT		
NAS1132-1	521411	400							HED	F/C	H/T	BL	G/S	T/R	PLT	LAB	INS	
NAS1102-3-16	521412	370							HED	CLP	F/C	H/T	BL	T/R	FLW	PLT	LAB	INS
PAN3008-05-023	521529	1000		HED	CLP	EXT	F/C	MRK	H/T	BL	T/R	G/S	FLW	PLT	LAB	INS		

for a sample of 10 orders. Apart from process routings, the part data include batch details including order number and quantity; operation code and description; workstation reference; processing and set-up times and finally order release times.

The part data for a small sample of two different orders is provided in Tables 6.25 and 6.26. Careful consideration of the data pointed out a number of important limitations:

1. Several parts produced by LCA require operations which the company has outsourced to its supply chain. As a result these operations have lengthy lead times. Such part orders were not included in the supplied data.
2. The dataset comprised orders associated with a specific product family, not the full range of orders handled by LCA in the production period considered.
3. All the jobs in the dataset are routed through single machine workstations.
4. Following on from point 2, using part orders which relate to one product family means that although each order has its unique process sequence, an element of routing commonality exists with several orders differentiated from others by skipping or re-visiting a small number of workstations.
5. Lead times were supplied in the form of operational due-dates rather than customer promised due-dates. These were quite relaxed allowing jobs to spend several working days on the shop-floor. This is a reflection of LCA's attempt to cope with long set-up times.
6. Most non-manufacturing operations have zero processing times in the data supplied but in most cases a set-up time is considered.
7. Due to lack of relevant data, it was assumed that the factory does not suffer any machine breakdowns and that all maintenance is carried outside working shift patterns.

Even though top management at LCA demonstrated a high level of awareness of lean manufacturing principles and techniques, it was evident that LCA is merely considering but has not yet embarked on its own lean transformation journey. This is most characteristically expressed by their passive response to long set-up times.

Further insight into the company's practices, suggests that apart from taking certain Kaizen-inspired initiatives, LCA has not adopted important lean scheduling and pull control tools including, mixed-model production and level loading. Furthermore, they do not appear to employ sophisticated dispatching rules with job prioritisation mainly performed on a FCFS basis thus further limiting LCA's scheduling performance.

**Table 6.25 Part data for sample order 1: LCA case study**

<b>NAS1132-1</b>											
Batch Details											
Work Order No.	Size										
521411	400	<b>Routing</b>									
Machine Description	HED	F/C	H/T	BL	G/S	T/R	PLT	LAB	INS		
Machine Code	B04	D02	L01	L03	F01	H01	N01	K01	J01		
Operation Code	MM1115	1702	H0017	3004	2101	2207	P0133	L0103	9000		
Processing Time (hrs)	0.08	0.05	0.00	0.00	0.89	0.06	0.00	0.00	0.00		
Set-up Time (hrs)	0.75	0.55	0.00	0.50	0.53	0.55	0.00	2.00	0.75		

**Table 6.26 Part data for sample order 2: LCA case study**

<b>PAN3008-05-023</b>													
Batch Details													
Work Order No.	Size												
521529	1000	<b>Routing</b>											
Machine Description	HED	CLP	EXT	F/C	MRK	H/T	BL	T/R	G/S	FLW	PLT	LAB	INS
Machine Code	B04	C02	C04	D02	E02	L01	L03	H01	F01	J02	N01	K01	J01
Operation Code	K1115	1560	1402	1702	8000	H0017	3004	2207	2161	F0011	P0129	L0021	9000
Processing Time (hrs)	0.28	0.31	0.21	0.13	3.33	0.00	0.00	0.14	2.22	0.00	0.00	0.00	0.00
Set-up Time (hrs)	0.75	0.36	0.36	0.55	0.07	0.00	0.50	0.55	0.53	0.08	0.00	2.00	0.75



### 6.2.3 Simulation results and analysis

The data supplied by LCA was utilised to run a series of experiments using BASS. It is accepted that the scheduling system currently used by LCA follows the general principles of push production control. The experimentations sought to simulate the operation of LCA's scheduling system following the adoption of pull control and establish any resulting performance improvement.

The experiments carried out aimed at contrasting the performance of the push control system with that of pull control exercised by the Kanban, Base Stock and CONWIP mechanisms. Each of these mechanisms was tested using three different levels of initial inventory enough to achieve an anticipated service level of 20%, 30% and 40%. Given the known issue of long set-up times experienced by LCA, the dispatching rule employed was SST. Overall, 10 simulation tests were performed.

The results obtained from these experiments are presented in Table 6.27. These clearly suggest that the designed pull production control system is applicable to LCA's job-shop without any adaptation of its shop-floor layout. However, when considering the potential benefits resulting from the adoption of pull production control, there is a huge contrast between this set of results and the overwhelming evidence of the superiority of pull control provided by the experiments carried out using benchmark problems.

The case-study results suggest that pull production control is significantly outperformed by push control across the majority of the 12 performance metrics considered. The only exceptions are average machine utilisation with the best performance obtained under the CONWIP mechanism and total throughput time where the lowest result was achieved when the Base Stock mechanism was applied and initial inventory level was set at 40%. Although seemingly outperforming pull control, the performance of the push control system is not satisfactory. Despite the relaxed operation due-dates, 42% of the scheduled orders are late. In addition, the total time spent configuring the machines is 351 hours and the total time in queue is 755 hours.

Further analysis into the underlying cause for the inferior performance of pull control in the case of LCA suggests that this lies in what appears to be a serious endemic issue for LCA, i.e. set-up times. When the system operates under push control, machines are set-up only

**Table 6.27 Performance of push and pull production control LCA case study**

Performance Metric	Kanban (SST)			Base Stock (SST)			CONWIP (SST)			
	Push	SL=20%	SL=30%	SL=40%	SL=20%	SL=30%	SL=40%	SL=20%	SL=30%	SL=40%
Total throughput time (hours)	1,148	4,192	926	917	2,789	822	692	12,502	4,974	3,970
Average flow time (hours)	25	102	37	37	74	35	32	269	118	98
Makespan (hours)	175	293	199	198	253	192	186	492	317	295
Earliness (hours)	184	0	0	0	0	0	0	0	0	0
Tardiness (hours)	485	4,192	926	917	2,789	822	692	12,502	4,974	3,970
Earliness/Tardiness (hours)	301	4,192	926	917	2,789	822	692	12,502	4,974	3,970
Number of tardy jobs (jobs)	21	50	50	50	50	50	50	50	50	50
Fill rate (%)	0	0	0	0	0	0	0	0	0	0
Average machine utilisation (%)	3	1	2	2	2	2	2	1	1	1
WIP level (jobs)	7	27	14	13	32	16	13	31	21	20
Total time in queue (hours)	755	6,909	2,012	1,879	7,417	2,567	1,861	13,648	5,803	4,872
Total set-up time (hours)	351	1,104	732	668	922	662	662	1,690	1,089	995

**Key:**  
 Most inferior performance  
 Most superior performance

once for each batch. However, under pull control the system is designed to meet demand using its available inventory. When this is not adequate to fulfil the full order, the order is fulfilled in part and the inventory consumed for this purpose is replenished triggering production in upstream workstations. Every time a batch is split, the set-up operation for the split component of the order needs to be repeated. It is therefore not surprising that the simulation output indicates that the total time spent setting up the machines under pull control is triple that required under push control. It can be justifiably argued that unaddressed prohibitively long set-up times have impaired the performance of pull production control.

Normally, in a MTO system, inventory levels should not be set as high as 40% of the anticipated demand. However, the pull control mechanisms were tested with inventory levels large enough to meet 40% of the demand simply to demonstrate that effect of the designed batch-splitting behaviour under pull control on set-up time requirements. The results do indeed support this premise. The lower the level of initial inventory, the higher the number of split batches and therefore the greater the total time spent configuring the machines.

### **6.3 Chapter summary**

This chapter presents a series of experiments carried out to validate BASS and answer the research questions of this thesis. Based on benchmark job-shop problems reported in the literature three different sets of experiments were devised. The first comprised static experiments involving a fairly stable mix of orders and several simplifications with regard to job arrival and set-up times. The static experiments were particularly useful. They provided the first indications of the superior performance of the Kanban mechanism in relation to the majority of the criteria considered in this thesis. This was demonstrated consistently in all the 160 static tests.

Given that one of the strengths of BASS is its ability to exhibit dynamic behaviour and cope with unexpected disturbances, a second set of experiments involving stochastic job data, dynamic order arrivals and machine breakdowns were designed and carried out. Similarly to the static experiments, the dynamic experiments also tested the job-shop's performance under the four different types of production control considered and a variety of dispatching rules. Machine breakdowns were assumed to have no impact on WIP and the initial inventory levels were set so that they could achieve an estimated fill rate of 20%. The obtained results illustrated that Base Stock outperformed all other control mechanisms in terms of most of the criteria considered in these experiments.

A third and final set of experiments was designed to gain further insight into the performance of pull mechanisms. The original dynamic experiments were modified to consider tighter conditions involving scrap parts caused by machine faults and significantly lower levels of initial inventory. The sensitivity analysis performed confirmed the dominance of Base Stock as the best performing production control mechanism.

Overall, the following primary conclusions can be drawn:

1. The experimentation results confirm that BASS successfully overcame the technical barriers inhibiting the implementation of pull production control in job-shops.
2. The experimentation results demonstrate patterns of behaviour which are repeated across the complete set of experiments. This practically proves that the designed job-shop scheduling system behaves consistently under both push and pull control.
3. Results obtained from the experiments carried out, confirm that indeed, a performance differential can be achieved from the application of pull control in job-shops.
4. Analysis of the results verifies the superior performance of the pure pull control mechanisms, namely Kanban and Base Stock in terms of the majority of performance criteria considered.
5. More specifically, Kanban and Base Stock outperform the push system in terms of job flow time and tardiness which are client-oriented criteria compared to machine utilisation, WIP and time in queue metrics which have a shop-floor focus.
6. The best machine utilisation, WIP and time in queue performance is observed when the job-shop operates under either push or hybrid push/pull control exercised by CONWIP.

The above findings are extremely significant as they relate directly to the two research questions posed in this thesis. Conclusions 1 and 2 support the view that pull production control can be directly applied to job-shops therefore answering the first research question. Points 3, 4 and 5 above indicate that the application of pull control to job-shops improves their performance across of a range of criteria thus providing an answer to the second research question.

Secondary findings resulting from the experimentations presented in this chapter are as follows:

1. The performance of Kanban and Base Stock are directly influenced by the stability of the demand and order mix. Kanban outperformed all other control systems in the static experiments, whereas the most superior performance was achieved by Base Stock under conditions of variable demand.
2. The initial inventory is an important initialisation parameter which determines the system's fill rate (service level) under pull control as well as its overall job flow time and tardiness performance.
3. The decision to consider several dispatching rules in the experimentations was justified. Dispatching rules are key performance determinants.

Using case data sourced from a manufacturing company which operates within the aerospace sector, simulation tests were carried out to study the introduction of pull control to an existing production system. The company operates a MTO policy and its shop-floor has a functional layout. It is therefore a typical non-repetitive production system. Simulation results confirmed the feasibility of implementing pull production control in a real-life production system without any reconfiguration of its shop-floor.

However, the supplied scheduling input data was found to be subject to several limitations. In particular, the data included uncharacteristically long set-up times. These in fact represented a serious data anomaly which distorted the performance of the pull control mechanisms. The simulation results showed that set-up times are three times higher under pull production control compared to the traditional push system. This is due to the heavy reliance of pull control mechanisms on batch splitting which in turn results in multiple set-ups per order.

Since the inception of JIT, set-up time reduction using SMED techniques has been emphatically advocated as one of the most important prerequisites for the successful implementation of pull production control (Sugimori et al., 1977; Shingo, 1981; Pegels, 1984; Ohno, 1988; Mehra and Inman, 1992; Flynn et al., 1995; Zhu and Meredith, 1995, Sakakibara et al., 1997; Shah and Ward, 2003; Jayaram et al., 2010). One of the main findings from Chapter 2 is that in order to reap the benefits of lean production, manufacturers need to embrace the lean philosophy and tools holistically. This case-study verified the findings resulting from the review of the lean manufacturing literature. It produced substantial evidence that had the adoption of pull control not been entirely piece-meal in the case of LCA, the potential of pull production control would not be so severely compromised.

Although the case study findings are of some value, the inherent limitations of the data reduce their generalisability. On the contrary, this chapter presented extensive

simulation testing performed using job-shop scheduling problem instances reported in the literature. These are sufficiently representative test-beds for the evaluation of the performance of pull production control in job-shops. The simulation results obtained from 260 experiments produced overwhelming evidence in favour of the superiority of pull production control under tight testing conditions.

The next chapter presents the overall conclusions of this thesis and provides directions for further research.

## **7 Conclusions**

This chapter summarises and reiterates the motivation to conduct this research. It restates the aim and main contribution of this thesis and concludes with recommendations for further extensions of this research.

### **7.1 Summary of research rationale, aims and key contributions**

The lean paradigm has been hailed as one of the most significant breakthroughs in the history of modern manufacturing. Since its inception almost a century ago, Lean Manufacturing (LM) has withstood the test of time and remained at the focal point of academic research and industry practice. As a management philosophy, leanness has transcended the boundaries of manufacturing with other industrial sectors seeking to embrace it in order to achieve world-class performance.

The benefits resulting from the adoption of LM are well-documented. Despite its overwhelming success, leanness is surrounded by a series of contentious issues. At the core of this controversy lies the question of its universality with opponents challenging its general applicability to all production systems.

Evidence from the literature suggests that LM has been primarily applied to serial production systems. It further reveals that this is mainly due to the fact that its forerunner, the Toyota Production System (TPS) was conceived in a mass production environment and all its tools including Just-in-Time (JIT), the engine behind the TPS, were designed for repetitive mass production systems. The practical complexity of applying JIT scheduling and control to non-serial production systems gave impetus to the argument against the transferability of LM into non-repetitive manufacturing.

The growing interest in the lean paradigm and its indisputable success provided the initial stimulus for this research which set out to explore the potential of lean in non-repetitive manufacturing. With lean production scheduling and control being perceived as the main obstacles for the adoption of the lean paradigm, this thesis sought to investigate the applicability of these practices in non-repetitive production systems employing functional job-shop layouts.

The universality of LM is not unexplored. There is an emerging body of research that attempts to shed light on this issue. Previous studies exemplify two distinct ways in which lean scheduling and control can be applied to non-repetitive manufacturing. The first stream of research proposes the integration of JIT with non-JIT production planning and control systems. The second stream promotes shop-floor layout

reconfiguration as the key enabler for the application of JIT scheduling and control to job-shops. This approach relies on the creation of physical or virtual manufacturing cells in order to standardise product flows and increase the degree of production repetitiveness in job-shops.

Research advocating the integration of JIT with conventional production planning and control systems presents conceptual models which seek to modify the logic of diametrically opposed systems in order to make the resulting hybrid adaptable to the characteristics of job-shops. In their majority, such conceptual models fail to adequately address the practical implications of the proposed integration. They are also designed for specific small-scale problems. This proposed hybridisation is often a compromise approach resulting in some loss of the benefits which could otherwise be delivered from the implementation of the original JIT scheduling and control practices. Conversely, the proponents of shop-floor layout adaptation do not consider the significant investment costs for such reconfigurations and the disruption to production.

This thesis examined the transferability of JIT scheduling and production control by arguing that these can be directly applied to job-shops. This contention rejects the view that shop-floor reconfiguration and/or integration of JIT with non-JIT production planning systems are prerequisites for the implementation of JIT in non-repetitive manufacturing systems.

Moreover, this thesis has sought to achieve a secondary aim concerned with the impact of JIT scheduling and control on job-shop performance. The aims of this thesis are founded on the premise that novel agent-based simulation can provide the enabling technology for the direct application of JIT scheduling and control to job-shops.

The investigation carried out in the context of this research commenced with a historical review of LM with the view of demystifying leanness as a concept.

### **7.1.1 Historical evolution, implementation and current state of the lean paradigm**

The purpose of this historical review was to examine the origins of LM and develop an insight into its overarching principles and constituent parts. The review considered the context and circumstances which led to the inception of leanness. It further highlighted that in the same way that LM developed out of need, it continues to evolve in line with its core principle of continuous improvement. The analysis of the evolutionary trajectory



of leanness drew attention to its most contemporary form encapsulated in the lean enterprise model.

The review of the LM literature identified a plethora of recent success stories. These showcase the role of leanness in attaining and sustaining competitive advantage and world-class manufacturing status. In addition to the superior performance of LM, the investigation confirmed its currency and revealed that many rival paradigms are in fact founded on the core values of leanness.

In particular, emphasis was placed on the drivers and barriers to the implementation of LM. This exposed the following misconceptions surrounding leanness which are highly relevant in the analysis of the main findings of this thesis:

- From a strategic perspective, leanness is a management philosophy whereas at the operational level it is a multifaceted production system. The system comprises interlocking components which synergistically deliver a wide range of lean benefits. Separating components from the whole compromises success as it removes the necessary infrastructure for the operation of certain lean tools. The lean enterprise model epitomises the holistic nature of the lean paradigm.
- The adoption of leanness requires judicious planning, top management support, employee buy-in, cultural change and long-term commitment to continuous improvement in order to sustain its success.

The historical review of leanness pointed out the complexity of classifying lean practices and organising them into categories. It also reaffirmed preliminary indications of the limited adoption of the lean paradigm in non-repetitive manufacturing thus reinforcing the rationale for this research.

### **7.1.2 Pull production mechanisms within Production Planning and Control (PPC) hierarchical frameworks**

This thesis is concerned with the application of JIT scheduling and control techniques originally designed for repetitive manufacturing to non-repetitive production systems. A comparative analysis of these systems was carried out to explore the characteristics of their shop-floor layouts, namely flow-shops and job-shops respectively, and their impact on production scheduling and control.

The objectives and scope of scheduling and control were investigated in-depth. The review revealed that although loading and sequencing performed in the context of scheduling are relevant for both job-shops and serial flow lines, their degree of

importance in these systems varies, as do the tools used to carry them out. It was further stressed that scheduling and control are intertwined practices as control seeks to coordinate product flows in line with baseline production schedules. This finding had certain importance for the agent-based simulation model developed in this thesis. In order to test the applicability of JIT production control in job-shops, the model had to simultaneously emulate the scheduling and control functions performed in their context.

The comparative study further delved into pull production control mechanisms utilised in the context of JIT, contrasting these with push control implemented in conventional non-JIT production systems. The review considered recent developments in the field of pull production control including the creation of hybrid push/pull control mechanisms e.g. CONWIP and modifications to the first pull control mechanism originally introduced in the context of JIT, namely Kanban. This investigation indicated that:

- Research attention remains focused on Kanban, Base Stock and CONWIP. This finding provided the justification for embedding the aforementioned pull control mechanisms in the agent-based simulation model developed in this thesis. Kanban and Base Stock provided the pure pull mechanisms used to mainly answer the first research question posed by the thesis. On the contrary, hybrid CONWIP was considered mainly in relation to the second research question concerned with the impact of various push/pull mechanisms on job-shop performance.
- The majority of studies reviewing the application of pull control mechanisms and their effect on system performance consider repetitive production systems. This finding provided further compelling evidence in support of the need to carry out this research.

The review of production scheduling and control identified that in practice these are not carried out in isolation. On the contrary, they form part of a broader Production Planning and Control (PPC) hierarchy. Production planning is a centralised function typically performed by Material Requirements Planning (MRP) systems. MRP determines medium-term production plans comprising order releases. Scheduling performed at the shop-floor level converts these into detailed short-term operation schedules.

The investigation substantiated the interdependency of planning, scheduling and control. However, no categorical evidence was produced to support the view that combined with conventional MRP planning systems, pull control mechanisms can be successfully applied to job-shops (Geraghty and Heavey, 2005). Therefore,

investigating the direct application of pull production control to non-repetitive manufacturing systems, in line with the first research question proposed by this research, is highly relevant in that context.

### **7.1.3 The transferability of pull production control using simulation**

A deductive approach was adopted in order to identify the most suitable tool for carrying out the LM transferability test. Research into production scheduling and control revealed that the relevant literature is primarily concerned with solving scheduling problems to optimality despite the latter having limited practical value from an industrial viewpoint. It further demonstrated the heavy reliance of this body of literature on prescriptive modelling. With the main focus of this thesis being on researching and assessing rather than optimising system performance, a descriptive approach was found to be more appropriate.

Computer simulation is currently the most prevalent descriptive modelling approach. The comparative review of different simulation techniques suggested that intermittent production in job-shop settings can be best modelled using Discrete Event Simulation (DES). A fine distinction was also drawn between conventional and agent-based DES.

The analysis produced overwhelming evidence in support of the unique ability of Agent-Based Simulation (ABS) and Multi-Agent Systems (MAS) to cope with the modelling complexities of decentralised pull production control applied to distributed heterogeneous and dynamic environments such as job-shops.

Despite this, the review of applications of simulation in the field of production planning and control revealed that studies employing agent-based simulation are significantly outnumbered by those using conventional simulation. This finding suggests that the novelty and pre-eminence of agent-based simulation in this area remains largely unexplored.

### **7.1.4 Gaps and limitations of applications of pull production control to non-repetitive manufacturing systems**

This thesis presents a state-of-the-art review of the applications of lean scheduling and control in job-shops using simulation modelling. This showed that research in this area is still in its infancy. Another inference derived from this review concerns the significant limitations and gaps of studies conducted in this area.

In several investigations, lean scheduling and control are considered separately despite being recognised as integrated functions. Given the practical complexity of applying JIT production in these settings, pull control is often in the sidelines of research carried out to date. Moreover, emphasis is often placed on the adoption of selective JIT scheduling tools and principles. Claiming to have created a lean job-shop by merely optimising its performance in terms of the JIT earliness/tardiness scheduling objective (Wu and Weng, 2005) is a case in point.

Reconfiguration of functional into cellular layouts or part grouping and routing through virtual cells has been argued as a prerequisite for the implementation of pull control in job-shops. However, the impact of the proposed shop-floor adaptation on the job-shop's ability to accommodate High Variety Low Volume (HVLV) production and associated costs are not sufficiently considered in research subscribing to this view.

The review identified a number of studies which attempted to apply pull production control to job-shops. Hybridisation and adaptation of pull control mechanisms was found to be the common approach employed in these studies. However, there is evident complacency towards modifications of pull control mechanisms which, on closer examination, violate the fundamental principles of JIT pull control.

Owing to the complexity of applying pull control in job-shops, several studies merely present conceptual models of JIT-controlled job-shops. Other studies develop simulation models of job-shops implementing pull production control which are experimentally tested using simplistic small-scale problem scenarios. Very few of these models are empirically validated. These limitations raise legitimate concerns with regards to the contribution of previous research in extending pull production control as originally conceived in the context of JIT to job-shops.

#### **7.1.5 Design and development of an intelligent agent decision support system for the extension of pull production control to lean job-shops**

The first step in designing a MAS to schedule and control production in lean job-shops was to identify the practical challenges associated with the application of pull control mechanisms namely Kanban, Base Stock and CONWIP. The design process focused on establishing the agent types needed to emulate the infrastructural and behavioural components of the proposed job-shop scheduling system.

The analysis delved into the volume and complexity of interactions and coordinated decision-making required to model the operation of pull production control, emphatically arguing these can only be handled by intelligent software agents. The key role of the Job Agents (JA) and Workstation Output Buffer Agents (WOBA) in delivering most of the functionality required to implement the pull control logic was an aspect which was discussed at length.

The modelling phase highlighted the need to maintain an initial level of inventory in the system in order to facilitate the operation of pull control mechanisms. It further identified issues arising from the application of pull control which had remained unaddressed in the relevant literature. These are primarily concerned with the prevention of system deadlocks when the inventory available in the system is not sufficient to meet demand at the final production stage.

The adopted pull control mechanisms were designed to allow batches to be split and partially fulfilled in case of low inventory levels. The job-shop was designed to trigger production in upstream stages in order to replenish the consumed inventory and fulfil outstanding or partially fulfilled orders.

Based on the above, the **main contribution of this thesis to knowledge in the field of production and operations management** is summarised in the following points:

1. This research extended the range of applicability of lean production scheduling and control techniques into job-shops with functional layouts where their adoption was previously dismissed a priori.
2. In the context of this work, JIT production control mechanisms were enhanced with the view of optimising the overall operational efficiency of job-shops.
3. Using experimental and empirical data, this research quantified the performance differential resulting from the application of lean scheduling and production control in job-shops.

#### **7.1.6 Evaluation of job-shop performance following the introduction of pull production control**

Having achieved its primary aim, this research sought to assess the impact of applying pull production control to job-shops. To achieve this, the developed MAS was designed to simulate the operation of a hypothetical job-shop under both push and pull control and further measure its performance using a selection of criteria.

An extensive set of experiments was designed to study the job-shop's performance under static and dynamic conditions. The latter are concerned with the arrival of orders, their mix and batch sizes as well as unexpected disturbances caused by machine breakdowns and the receipt of high-priority orders.

In all the experiments, the pure pull control mechanisms, namely Kanban and Base Stock resulted in better system performance compared to push and the hybrid push/pull CONWIP mechanism. The superior performance of pull control was mainly observed in relation to criteria such as flow time, tardiness and throughput all of which are client-oriented. Conversely, Kanban and Base Stock were only outperformed by push and CONWIP in terms of the WIP, machine utilisation and time in queue metrics which are tightly linked to shop-floor performance.

What do these findings mean for a production system which strives to be lean? The experiments ostensibly showed that the application of pull control in job-shops improves system responsiveness at the expense of leanness. Leanness seeks to eliminate waste. In manufacturing systems waste can take several forms, but is mainly associated with overproduction and excessive WIP. These in turn are the result of machine over-utilisation in line with the mass production mentality.

However, the analysis of the experimentation results reveals that in fact the high levels of WIP and machine utilisation observed when pull control is applied are not the result of overproduction. They are rather a natural consequence of the system's attempt to replenish its inventory and fulfil outstanding orders. Therefore, the poor performance of pull production control in terms of the aforementioned criteria is not an accurate reflection of the job-shop's leanness.

Moreover, the substantial performance differential achieved in terms of flow time, throughput and tardiness need to be viewed in light of the central importance the notion of value has in the lean paradigm. On this basis, the enhanced job-shop responsiveness must be emphasised as it directly adds value to the service provided to customers.

The MAS developed in the context of this research was empirically validated using case data supplied by a manufacturer specialising in the production of fastening systems for aircrafts. The company's Make-to-Order (MTO) policy of responding to demand and its functional shop-floor layout meant that the case-study was a suitable test-bed for the application of pull production control.

The simulation results confirmed the feasibility of using the pull mechanisms to control production in a real-life job-shop which was not reconfigured in any way. However, extremely long set-up times in the case data constituted a serious anomaly which yielded diluted results concerning the performance of the pull control mechanisms. Specifically, Kanban, Base Stock and CONWIP were all outperformed by push production control.

Set-up time reduction is emphatically argued in the literature as an important prerequisite for JIT scheduling and pull production control. The case study results confirmed the prevailing view that the piece-meal adoption of lean tools and practices compromises the success of LM. The generalisability of the case study results was found to be significantly reduced due to inherent limitations in the data and the attempt to selectively apply pull control to a production system lacking the necessary lean infrastructure prerequisites.

## **7.2 Implications for theory**

The MAS developed in this thesis replaces the physical Kanban cards used in early implementations of pull control mechanisms with messages exchanged by software agents. These electronic messages contain instructions for the replenishment of inventory consumed in order to fulfil production orders in full or partially. This innovation goes some way to minimise the need for further research in the design, modelling and optimisation of the parameters of traditional token-based pull control mechanisms, such as for instance the number of Kanban cards and size of associated transportation containers circulated within the system.

In designing a MAS which applies pull production control to non-repetitive manufacturing, the analysis drew attention to an important initialisation parameter in the modelled system, namely the job-shop's initial inventory. This inventory was found to be vital for the operation of the enhanced Kanban, Base Stock and CONWIP mechanisms. In the experimental and empirical validation of the MAS, the initial system inventory was set at assumed predetermined levels. The levels were kept low consistently with the MTO policy typically exercised in job-shops. However, given the dynamic nature of non-repetitive production it would be useful to further investigate how such an important parameter could be determined and optimised using sophisticated demand forecasting models.

### **7.3 Implications for current industry practice**

In relation to LM, the findings of this research reinforced the currency and world-class manufacturing status of the lean paradigm. They also provided further support to the dominant view that manufacturers who wish to adopt the lean paradigm need to embrace it holistically and commit to an on-going policy of continuous improvement.

This research proved that pull production control can be successfully applied to job-shops thus overturning the findings of previous research which strongly opposed the transferability of JIT tools. The potential for more industrial applications of pull production control must be explored further.

This also holds true for agent-based simulation. In the context of this research, an intelligent agent decision support system allowed the application (both empirically and experimentally) of pull production control to job-shops. Agent-based simulation should also be investigated as the enabling technology for developing JIT scheduling and control software for industrial applications.

From a more practical standpoint, the application of pull production control to real-life job-shop settings would result in the more frequent transportation of parts within and between workstations. It would further necessitate the provision of additional space in output buffers which would need to temporarily store multiple types of WIP. Both logistical implications are directly related to the batch-splitting function that the system is designed to perform to replenish its inventory and meet outstanding demand.

### **7.4 Directions for further research**

The application of JIT pull production control to job-shops opens new avenues for research in the exploration and understanding of the transferability of leanness. The following are recommendations for possible extensions of the work presented in this thesis:

1. Job-shops controlled by pull mechanisms are designed to operate as inventory replenishment systems. The interoperability of pull production control with an effective inventory management system able to cope with the complex inventory transactions that would typically occur in such settings should be investigated further.
2. Further research may be worthwhile into the use of Automatic Identification and Data Capture (AIDC) technology, e.g. Radio Frequency Identification (RFID) (Stevenson, 2006) in pull-controlled job-shops. Such wireless technology will



allow the tracking of WIP in real-time and can therefore enhance the responsiveness and adaptability of pull production control to dynamic disturbances.

3. For pull control to be applied to job-shops, the latter would need to maintain intermediate inventory for a variety of products. WIP levels can be reduced by linking pull control with a planning system which would seek to group orders and schedule the production of a small mix of products in each production period. This type of integration should be investigated further.
4. Considering the logistical challenges associated with the frequency of temporary storing and transporting WIP, the design of material handling and storage systems suitable for JIT-controlled job-shops presents another area for further exploration.

It may also be of interest to extend the MAS developed in this research by including variations of the Kanban, Base Stock and CONWIP mechanisms with the view of assessing their impact on job-shop performance.

Another opportunity will be to source and use case data which avoids some of the limitations of the industrial data used to empirically validate the applicability of pull control to job-shops in this work. Finally, the empirical application of the developed MAS to real-life non-repetitive manufacturing systems which have embarked on the lean transformation journey and invested in a supportive JIT infrastructure could be another important step in quantifying the real benefits resulting from the extension of pull control to job-shops.

## References

- Abdulmalek, F. A. and Rajgopal, J. (2007) "Analysing the benefits of lean manufacturing and value stream mapping via simulation: A process sector case study", *International Journal of Production Economics*, 107(1), pp. 223-236.
- Ahmad, S., Schroeder, R. G., and Sinha, K. K. (2003) "The role of infrastructure practices in the effectiveness of JIT practices: implications for plant competitiveness", *Journal of Engineering Technology Management*, 20(3), pp. 161-191.
- Akillioglu, H. and Onori, M. (2011) Evolvable production systems and impacts on production planning. In: *Proceedings of IEEE International Symposium on Assembly and Manufacturing (ISAM)*, 25-27 May 2011, Tampere, Finland, pp. 1-6.
- Akturk, M. S. and Erhun, F. (1999) "An overview of design and operational issues of kanban systems", *International Journal of Production Research*, 37(17), pp. 3859-3881.
- Anand, G. and Kodali, R. (2009) "Selection of lean manufacturing systems using the analytic network process - a case study", *Journal of Manufacturing Technology Management*, 20(2) pp. 258-289.
- Andijani, A. (1997) "Trade-off between maximizing throughput rate and minimizing system time in kanban systems", *International Journal of Operations & Production Management*, 17(5), pp. 429-445.
- AOS (2012a) *JACK™ Academic licence*. AOS Group. [online] Available at: <<http://www.aosgrp.com/products/jack/registration/regform.html>> [Accessed: 1 October 2012]
- AOS (2012b) *JACK™ Agent language*. AOS Group. Available at: <[http://www.aosgrp.com/documentation/jack/Agent\\_Manual\\_WEB/index.html#The JACK Agent Language](http://www.aosgrp.com/documentation/jack/Agent_Manual_WEB/index.html#The_JACK_Agent_Language)> [Accessed: 1 October 2012]
- AOS (2012c) *JACK™ product information*. AOS Group. [online] Available at: <<http://www.aosgrp.com/products/jack/>> [Accessed: 1 October 2012]
- AOS (2012d) *JACK™ Agent oriented concepts*. AOS Group. [online] Available at: <[http://www.aosgrp.com/documentation/jack/Agent\\_Manual\\_WEB/index.html#Agent Oriented Concepts](http://www.aosgrp.com/documentation/jack/Agent_Manual_WEB/index.html#Agent_Oriented_Concepts)> [Accessed: 1 October 2012]
- AOS (2012e) *JACK™ JDE manual*. AOS Group. [online] Available at: <[http://www.aosgrp.com/documentation/jack/JDE\\_Manual\\_WEB/index.html](http://www.aosgrp.com/documentation/jack/JDE_Manual_WEB/index.html)> [Accessed: 1 October 2012]

AOS (2012f) *JACK™ JACOB manual*. AOS Group. [online] Available at: <[http://www.aosgrp.com/documentation/jack/JACOB\\_Manual\\_WEB/index.html](http://www.aosgrp.com/documentation/jack/JACOB_Manual_WEB/index.html)>

[Accessed: 1 October 2012]

Arasteh, A., Aliahmadi, A., Sefidkoohi, H. M. and Omran, M. M. (2011) "Lateral solutions for optimizing holding costs in job shops", *International Journal of Advanced Manufacturing Technology*, 56(1-4), pp. 261-272.

Arnheiter, E. D. and Maleyeff, F. (2005) "The integration of lean management and Six Sigma", *The TQM Magazine*, 17(1), pp. 5-18.

Askin, R. G. and Chen, J. (2006) "Dynamic task assignment for throughput maximization with worksharing", *European Journal of Operational Research*, 168(3), pp. 853-869.

Azaron, A., Fynes, B. and Modarres, M. (2011) "Due date assignment in repetitive projects", *International Journal of Production Economics*, 129 (1), pp. 79-85.

Bahaji, N. and Kuhl, M. E. (2008) "A simulation study of new multi-objective composite dispatching rules, CONWIP, and push lot release in semiconductor fabrication", *International Journal of Production Research*, 46(14), pp. 3801-3824.

Bamber, L. and Dale, B. G. (2000) "Lean production: A study of application in a traditional manufacturing environment", *Production Planning & Control*, 11(3), pp. 291-298.

Banks, J., Carson II, J. S., Nelson, B. L. and Nicol, D. M. (2010) *Discrete-event system simulation*. 5<sup>th</sup> ed. Upper Saddle River, NJ: Pearson Education, Inc.

Barnes, D. (2008) *Operations Management: An international perspective*. London: Thomson Learning.

Bartezzaghi, E. (1999) "Evolution of production models", *International Journal of Operations & Production Management*, 19(2), pp. 2-15.

Baybars, I. (1986) "A survey of exact algorithms for the simple assembly line balancing problem", *Management Science*, 32(8), pp. 909-932.

Baykasoğlu, A. and Durmuşoğlu, Z. D. U. (2012) "Flow time analyses of a simulated flexible job shop by considering jockeying", *International Journal of Advanced Manufacturing Technology*, 58(5-8), pp. 693-707.

- Baykasoğlu, A. and Özbakır, L. (2010) "Analyzing the effect of dispatching rules on the scheduling performance through grammar based flexible scheduling system", *International Journal of Production Economics*, 124(2), pp. 369-381.
- Bayou, M. E. and De Korvin. A. (2008) "Measuring the leanness of manufacturing systems - A case-study of Ford Motor company and General Motors", *Journal of Engineering Technology Management*, 25(4), pp. 287-304.
- Beamon, B. M. and Bermudo, J. M. (2000) "A hybrid push/pull control algorithm for mult-stage, multi-line production systems", *Production Planning & Control*, 11(4), pp. 349-356.
- Becker, C. and Scholl, A. (2006) "A survey on problems and methods in generalized assembly line balancing", *European Journal of Operational Research*, 168 (3), pp. 694-715.
- Benton, W. C. and Shin, H. (1998) "Manufacturing planning and control: The evolution of MRP and JIT integration", *European Journal of Operational Research*, 110(3), pp. 411-440.
- Berger, G., (1987) Ten ways MRP can defeat you. In: *Proceedings of APICS Conference*, pp. 240-243.
- Bhasin, S. (2012) "An appropriate change strategy for lean success", *Management Decision*, 50(3), pp. 439-458.
- Bhasin, S. and Burcher, P. (2006) "Lean viewed as a philosophy", *Journal of Manufacturing Technology Management*, 17(1), pp. 56-72.
- Bitichi, U. S., Mendibil, K., Martinez, V. and Albores, P. (2005) "Measuring and managing performance in extended enterprises", *International Journal of Operations & Production Management*, 25(4), pp. 333-353.
- Black, J. T. (2007) "Design rules for implementing the Toyota Production System", *International Journal of Production Research*, 45(16), pp. 3639-3664.
- Blackstone Jr, J. H., Cox III, J. F. and Schleier Jr, J. G. (2009) "A tutorial on project management from a theory of constraints perspective", *International Journal of Production Research*, 47(24), pp. 7029-7046.
- Blackstone, J.H., Phillips, D.T., and Hogg, G.L. (1982) "A state-of-the-art survey of dispatching rules for manufacturing job shop operations", *International Journal of Production Research*, 20(1), pp. 27-45.

- Blum, C. (2008) "Beam-ACO for Simple Assembly Line Balancing", *INFORMS Journal on Computing*, 20(4), pp. 618-627.
- Bonvik, A. M., Couch, C. E. and Gershwin, S. B. (1997) "Comparison of production-line control mechanisms", *International Journal of Production Research*, 35(3), pp. 789-804.
- Bordini, R. H., Dastani, M., Dix, J. and Seghrouchni, A. E. F. eds. (2009) *Multi-agent programming: Languages, tools and applications*. New York, NY: Springer Science+Business Media, LLC.
- Borshchev, A. and Filippov, A. (2004) From system dynamics and discrete event to practical agent based modeling: Reasons, techniques, tools. In: *Proceedings of 22<sup>nd</sup> International Conference of the System Dynamics Society*, 25 - 29 July, 2004, Oxford, England.
- Boyle, T. A. and Scherrer-Rathje, M. (2009) "An empirical examination of the best practices to ensure manufacturing flexibility – Lean alignment", *Journal of Manufacturing Technology Management*, 20(3), pp. 348-366.
- Boysen, N., Fliedner, M. and Scholl, A. (2007) "A classification of assembly line balancing problems", *European Journal of Operational Research*, 183(2), pp. 674-693.
- Boysen, N., Fliedner, M. and Scholl, A. (2008) "Assembly line balancing: Which model to use when?", *International Journal of Production Economics*, 111(2), pp. 509-528.
- Boysen, N., Fliedner, M. and Scholl, A. (2009) "Sequencing mixed-model assembly lines: Survey, classification and model critique", *European Journal of Operational Research*, 192(2), pp. 349-373.
- Boysen, N., Kiel, M. and Scholl, A. (2011) "Sequencing mixed-model assembly lines to minimise the number of work overload situations", *International Journal of Production Research*, 49(16), pp. 4735-4760.
- Bozdogan, K., Milauskas, R., Mize, J., Nightingale, D., Taneja, A. and Tonaszuck, D. (2000) *Transition to a lean enterprise: A Guide for leaders. Volume II*. Massachusetts Institute of Technology. [online] Available at: <[http://lean.mit.edu/downloads/cat\\_view/94-products/577-enterprise-transition-to-lean](http://lean.mit.edu/downloads/cat_view/94-products/577-enterprise-transition-to-lean)> [Accessed: 1 October 2012]
- Brandon-Jones, A. and Slack, N. (2008) *Quantitative analysis in operations management*. Harlow: Pearson Education Ltd.

- Brockman, J. B. (2009) *Introduction to engineering modelling and problem solving*. Hoboken, NJ: John Wiley & Sons, Inc.
- Browne, J., Sackett, P.J. and Wormann, J. C. (1995) "Future manufacturing systems – towards the extended enterprise", *Computers in Industry*, 25(3), pp. 235-254.
- Buede, D. M. (2009) *The engineering design of systems: Models and methods*. 2<sup>nd</sup> ed. Hoboken, NJ: John Wiley & Sons, Inc.
- Bussmann, S., Jennings, N. R. and Wooldridge, M. (2004) *Multiagent systems for manufacturing control: A design methodology*. Berlin-Heidelberg: Springer-Verlag.
- Buzacott, J. A. (1989) "Queueing models of Kanban and MRP controlled production systems", *Engineering Cost and Production Economics*, 17(1-4) pp.3-20.
- C. Duri, C., Frein, Y. and Di Mascolo (2000) "Comparison among three pull control policies: kanban, base stock and generalized kanban", *Annals of Operations Research*, 93(1-4), pp. 41-69.
- Cakir, B., Altiparmak, F. and Dengiz, B. (2011) "Multi-objective optimization of a stochastic assembly line balancing: A hybrid simulated annealing algorithm", *Computers & Industrial Engineering*, 60(3), pp. 376-384.
- Caridi, M. and Cavalieri, S. (2004) "Multi-agent systems in production planning and control: an Overview", *Production Planning & Control: The Management of Operations*, 15(2), pp. 106-118.
- Çayirci, E. and Marinčić, D. (2009) *Computer assisted exercises and training: A reference guide*. Hoboken, NJ: John Wiley & Sons, Inc.
- Chang, T. M. and Yih, Y. (1994) "Generic kanban systems for dynamic environments", *International Journal of Production Research*, 32(4), pp. 889-902.
- Chary, S. N. (2009) *Production and operations management*. 4<sup>th</sup> ed. New Delhi: Tata McGraw-Hill Education Private Limited.
- Chase, R. B., Jacobs, F. R. and Aquilano, N. J. (2006) *Operations management for competitive advantage*. 11th ed. New York, NY: McGraw-Hill/Irwin.
- Cheeseman, M. J., Swann, P., Hesketh, G. B. and Barnes, S. (2005) "Adaptive manufacturing scheduling: a flexible and configurable agent-based prototype", *Production Planning & Control: The Management of Operations*, 16(5), pp. 479-487.

- Chen, K. and Ji, P. (2007) "A mixed integer programming model for advanced planning and scheduling (APS)", *European Journal of Operational Research*, 181(1), pp. 515-522.
- Chiang, T. C. and Fu, L. C. (2007) "Using dispatching rules for job shop scheduling with due date-based objectives", *International Journal of Production Research*, 45(14), pp. 3245-3262.
- Childe, S. J. (1998) "The extended enterprise – a concept of cooperation", *Production Planning & Control*, 9(4), pp. 320-327.
- Chong, C. S., Low, M. Y. H., Sivakumar, A. I. and Gay, K. L. (2006) A bee colony optimization algorithm to job shop scheduling. In: *Proceedings of Winter Simulation Conference (WSC)*, 3-6 December, Monterey, CA, pp. 1954 -1961.
- Cochran, J. K. and Kaylani, H. A. (2008) "Optimal design of a hybrid push/pull serial manufacturing system with multiple part types", *International Journal of Production Research*, 46(4), pp. 949-965.
- Cohen, Y., Vitner, G. and Sarin, S. C. (2006) "Optimal allocation of work in assembly lines for lots with homogenous learning", *European Journal of Operational Research*, 168(3), pp. 922-931.
- Cooney, R. (2002) "Is lean a universal production system? Batch production in the automotive industry", *International Journal of Operations & Production Management*, 22(10), pp.1130-1147.
- Crowson, R. (2006) *The handbook of manufacturing engineering: Product design and factory development*. 2<sup>nd</sup> ed. Boca Raton, FL: CRC Press.
- Cruickshanks, A.B., Drescher, R.D. and Graves, S.C. (1984) "A study of production smoothing in a job shop environment", *Management Science*, 30(3), pp. 368-380.
- Curry, G. L. and Feldman, R. M. (2010) *Manufacturing systems modeling and analysis*. 2<sup>nd</sup> ed. Berlin-Heidelberg: Springer-Verlag.
- Cusumano, M. A. (1985) *The Japanese automobile industry: Technology and Management at Nissan and Toyota*. Boston, MA: Harvard University Press.
- Dallery, Y. and Liberopoulos, G. (1995) A new kanban-type pull control mechanism for multi-stage manufacturing systems. In: *Proceedings of 3<sup>rd</sup> European Control Conference (ECC)*, Rome, Italy, pp. 3543-3548.

- David, F., Pierreval, H. and Caux, C. (2006) "Advanced planning and scheduling systems in aluminium conversion industry", *International Journal of Computer Integrated Manufacturing*, 19(7), pp. 705-715.
- De Toni, A., Caputo, M. and Vinelli, A. (1988) "Production management techniques: Push-pull classification and application conditions", *International Journal of Operations & Production Management*, 8(2), pp. 35-51.
- Demeter, K. and Matyusz, Z. (2011) "The impact of lean practices on inventory turnover", *International Journal of Production Economics*, 133(1), pp. 154-163.
- Detty, R. B. and Yingling, J. C. (2000) "Quantifying benefits of conversion to lean manufacturing with discrete event simulation: a case study", *International Journal of Production Research*, 38(2), pp. 429-445.
- Di Marzo Serugendo, G., Gleizes, M.-P. and Karageorgos, A. eds., (2011) *Self-organising software: From natural to artificial adaptation*. Berlin-Heidelberg: Springer-Verlag.
- Diaz, R. and Ardalan, A. (2010) "An Analysis of Dual-Kanban Just-In-Time Systems in a Non-Repetitive Environment", *Production and Operations Management*, 19(2), pp. 233-245.
- Dickson, E. W., Singh, S., Cheung, D. S., Wyatt, C. C. and Nugent, A. S. (2009) "Application of lean manufacturing in the emergency department", *Journal of Emergency Medicine*, 37(2), pp. 177-182.
- Dolgui, A., Guschinsky, N. and Levin, G. (2006) "A special case of transfer lines balancing by graph approach", *European Journal of Operational Research*, 168(3), pp. 732-746.
- Dominic, P. D. D., Kaliyamoorthy, S. and Kumar, M. S. (2004) "Efficient dispatching rules for dynamic job shop scheduling", *International Journal of Advanced Manufacturing Technology*, 24(1-2), pp. 70-75.
- Doolen, T. L. and Hacker, M. E. (2005) "A review of lean assessment in organizations: An exploratory study of lean practices by electronics manufacturers", *Journal of Manufacturing Systems*, 24(1), pp. 55-67.
- Dos Santos, N. R. and De Araújo, L. O. (2003) "Computational system for group technology – PFA case study", *Integrated Manufacturing Systems*, 14(2), pp. 138-152.



- Dunin-Keplicz, B. and Verbrugge, R. (2010) *Teamwork in multi-agent systems: A formal approach*. Chichester: John Wiley & Sons, Ltd.
- El-Bouri, A. and Shah, P. (2006) "A neural network for dispatching rule selection in a job shop", *International Journal of Advanced Manufacturing Technology*, 31(3-4), pp. 342-349.
- Emde, S., Boysen, N. and Scholl, A. (2010) "Balancing mixed-model assembly lines: a computational evaluation of objectives to smoothen workload", *International Journal of Production Research*, 48(11), pp. 3173-3191.
- Ennigrou, M. and Ghédira, K. (2008) "New local diversification techniques for flexible job shop scheduling problem with a multi-agent approach", *Autonomous Agents and Multi-Agent Systems*, 17(2), pp. 270–287.
- Eroglu, C. and Hofer, C. (2010) "Lean, leaner or too lean? The inventory performance link revisited", *Journal of Operations Management*, 29(4), pp. 356-369.
- Eswaramoorthi, M., Kathiresan, G.R., Jayasudhan, T.J., Prasad, P.S.S. and Mohanram, P.V. (2012) "Flow index based line balancing: a tool to improve the leanness of assembly line design", *International Journal of Production Research*, 50(12), pp. 3345-3358.
- Evertsz, R., Fletcher, M., Jones, R. Jarvis, J., Brusey, J. and Dance, S. (2004) *Implementing industrial multi-agent systems using JACK™*. In: Dastani, M., Dix, J. and Seghrouchni, A. E. F. eds. (2004) *PROMAS 2003, LNAI 3067*. Berlin-Heidelberg: Springer-Verlag, pp. 18–48.
- Feld, W. M. (2001) *Lean manufacturing tools, techniques and how to use them*. Boca Raton, FL: St Lucie Press.
- Ferber, J., (1999). *Multi-Agent Systems: An Introduction to Distributed Artificial Intelligence*. Reading, MA: Addison-Wesley.
- FIPA (2012) *FIPA specifications*. [online] Available at: <http://www.fipa.org/specifications/index.html> [Accessed: 1 October 2012]
- Fishwick, P. A. (2007) *Handbook of dynamic systems modelling*. Boca Raton, FL: Chapman & Hall/CRC, Taylor & Francis Group LLC.

- Fletcher, M., McFarlane, D., Lucas, A., Brusey, J. and Jarvis, D. (2003) *The Cambridge packing cell – A holonic enterprise demonstrator*. In: Mařík, V., Müller, J. and Pechoucek, M. eds. (2003) *Multi-agent systems and applications III: 3<sup>rd</sup> International central and eastern European conference on multi-agent systems, CEEMAS 2003, Prague Czech Republic Proceedings*. Berlin-Heidelberg: Springer-Verlag, pp. 533-543.
- Floudas, C. A. and Lin, X. (2004) "Continuous-time versus discrete-time approaches for scheduling of chemical processes: a review", *Computers and Chemical Engineering*, 28 (11), pp. 2109-2129.
- Flynn, B. B. (1987) "Repetitive lots: the use of a sequence-dependent set-up time scheduling procedure in group technology and traditional shops", *Journal of Operations Management*, 7(1-2), pp. 203-216.
- Flynn, B. B., Sakakibara, S. and Schroeder, R. G. (1995) "Relationship between JIT and TQM: Practices and performance", *Academy of Management Journal*, 38(5), pp. 1325-1360.
- Flynn, B., Morita M. and Machuca J. (2011) *Managing global supply chain relationships: Operations, strategies and practices*. Hershey, PA: IGI Global.
- Ford, A. (2009) *Modeling the environment*. 2<sup>nd</sup> ed. Washington, DC: Island Press.
- Framinan, J. M., and Ruiz-Usano, R. (2002) "On transforming job-shops into flow-shops", *Production Planning & Control*, 13(2), pp. 166-174.
- Framinan, J. M., Gonzalaz, P. L. and Ruiz-Usano, R. (2003) "The CONWIP production control system: review and research issues", *Production Planning & Control*, 14(3), pp. 255-265.
- Framinan, J. M., Ruiz-Usano, R. and Leisten, R. (2000) "Input control and dispatching rules in a dynamic CONWIP flow-shop", *International Journal of Production Research*, 38(18), pp. 4589-4598.
- Frayret, J.-M., D'Amours, S. and Montreuil, B. (2004) "Coordination and control in distributed and agent-based manufacturing systems", *Production Planning & Control: The Management of Operations*, 15(1), pp. 42-54.
- Frein, Y., Di Mascolo, M. and Dallery, Y. (1995) "On the design of generalized kanban control systems", *International Journal of Operations & Production Management*, 15 (9), pp. 158-184.

- Fullerton, R. R. and McWatters, C. S. (2001) "The production performance benefits from JIT implementation", *Journal of Operations Management*, 19 (1) pp. 81-96.
- Fullerton, R. R. and Wempe, W. F. (2009) "Lean manufacturing, non-financial performance measures, and financial performance", *International Journal of Operations & Production Management*, 29(3), pp. 214-240.
- Gaither, N. and Frazier, G. (2002) *Operations Management*. 9<sup>th</sup> ed. Cincinnati, OH: Thomson Learning.
- García-Hernando, A.-B., Martínez-Ortega, J.-F., López-Navarro, J.-M., Prayati, A. eds., (2008). *Problem solving for wireless sensor networks*. London: Springer-Verlag Ltd.
- Garey, M. R., Johnson, D. S., and Sethi, R. (1976) "The complexity of flowshop and jobshop scheduling", *Mathematics of Operations Research*, 1(2), pp. 117-129.
- Gaury, E. G. A., Pierreval, H. and Kleinjen, J. P. C. (2000) "An evolutionary approach to select a pull system among Kanban, Conwip and Hybrid", *Journal of Intelligent Manufacturing*, 112(), pp. 157-167.
- Geraghty, J. and Heavey, C. (2005) "A review and comparison of hybrid and pull-type production control strategies", *OR Spectrum*, 27(2/3), pp. 435-457.
- Giachetti, R. E. (2010) *Design of enterprise systems: Theory, architecture and methods*. Boca Raton, FL: CRC Press, Taylor & Francis Group LLC.
- Giere, R. N. (2004) "How Models Are Used to Represent Reality", *Philosophy of Science*, 71(5), pp. 742–752.
- Giordano, F. R., Fox, W. P., Horton, S. B. and Weir, M. D. (2009) *A first course in mathematical modelling*. 4<sup>th</sup> ed. Belmont, CA: Cengage Learning.
- Goldratt, E. (1997) *Critical Chain*. Great Barrington, MA: North River Press.
- Goldratt, E.M. and Cox, J. (1984) *The Goal*. Croton-on-Hudson, NY: North River Press.
- González-Benito, J. and Suarez-González, I. (2001) "Effect of Organizational Variables in JIT Purchasing Implementation", *International Journal of Production Research*, 39(10), pp. 2231-2249.
- González-R, P. L. and Framinan, J. M. (2009) "The pull evolution: from Kanban to customised token-based systems", *Production Planning & Control*, 20(3), pp. 276- 287.

- Gonzalez-R, P. L., Framinan, J. M. and Ruiz-Usano, R. (2010) "A multi-objective comparison of dispatching rules in a drum–buffer–rope production control system", *International Journal of Computer Integrated Manufacturing*, 23(2), pp. 155-167.
- Gravel, M. and Price, W. L. (1998) "Using the Kanban Job-shop Environment", *International Journal of Production Research*, 26(6), pp. 1105-1118.
- Greasley, A. (2008) *Operations Management*. London: SAGE Publications Ltd.
- Groover, M. P. (2008) *Automation, production systems, and computer-integrated manufacturing*. 3<sup>rd</sup> ed. Upper Saddle River, NJ: Prentice-Hall.
- Groover, M. P. (2011) *Principles of modern manufacturing*. New York, NY: John Wiley & Sons, Inc.
- Guinery, J. and MacCarthy, B. (2009) "Managing key interfaces in production planning and control", *Production Planning & Control*, 20(1), pp. 40-56.
- Gunasekaran, A. (1999) "Design and implementation of agile manufacturing systems", *International Journal of Production Economics*, 62(1-2), pp. 1-6.
- Gunasekaran, A. and Ngai, E. W. T. (2012) "The future of operations management: An outlook and analysis", *International Journal of Production Economics*, 135(2), pp. 687–70.
- Guo, Q. and Zhang, M. (2009) "A novel approach for multi-agent-based Intelligent Manufacturing System", *Information Sciences*, 179 (18), pp. 3079–3090.
- Gupta, J. N. D. (2002) "An excursion in scheduling theory: an overview of scheduling theory in the twentieth century", *Production Planning & Control*, 13(2), pp. 105-116.
- Gupta, J. N. D. (2002) "An excursion in scheduling theory: an overview of scheduling theory in the twentieth century", *Production Planning & Control*, 13(2), pp. 105-116.
- Gupta, M. and Kohli, A. (2006) "Enterprise resource planning systems and its implications for operations function", *Technovation*, 26 (5-6), pp. 687-696.
- Gupta, M. and Snyder, D. (2009) "Comparing TOC with MRP and JIT: a literature review", *International Journal of Production Research*, 47(13), pp. 3705- 3739.
- Gupta, S. M. and Al-Turki, Y. A. Y. (1997) "An algorithm to dynamically adjust the number of kanbans in stochastic processing times and variable demand environment", *Production Planning & Control*, 8 (2), pp. 133-141.

- Haizer, J. and Render, B. (2004) *Principles of operations management*. 5<sup>th</sup> ed. Upper Saddle River, NJ: Pearson Education Inc.
- Hall, R. W. (1982) *Kawasaki USA: Transferring Japanese Production Methods to the United States: a Case Study*. Falls Church, VA: APICS.
- Hall, R. W. (1983) *Zero inventories*. Homewood, IL: Dow Jones-Irwin.
- Hallgren, M. and Olhager, J. (2009) "Lean and agile manufacturing: external and internal drivers and performance outcomes", *International Journal of Operations & Production Management*, 29(10), pp. 976-999.
- Hallihan, A., Sackett, P. and Williams, G. M. (1997) "JIT manufacturing: the evolution to an implementation model founded in current practice", *International Journal of Production Research*, 35(4), pp. 901-920.
- Hamta, N., Fatemi Ghomi, S.M.T., Jolai, F. and Shirazi, M. A. (2012) "A hybrid PSO algorithm for a multi-objective assembly line balancing problem with flexible operation times, sequence-dependent setup times and learning effect", *International Journal of Production Economics*, 141(1), pp. 99-111.
- Hamzadayi, A. and Yildiz, G. (2012) "A genetic algorithm based approach for simultaneously balancing and sequencing of mixed-model U-lines with parallel workstations and zoning constraints", *Computers & Industrial Engineering*, 62(1), pp. 206-215.
- Harrison, A. (1992) *Just-in-time manufacturing in perspective*. London: Prentice-Hall.
- Harrod, S. and Kanet, J. J. (2012) "Applying work flow control in make-to-order job-shops", *International Journal of Production Economics*, <http://dx.doi.org/10.1016/j.ijpe.2012.02.017>.
- Hasan, S. M. K., Sarker, R., Essam, D. and Cornforth, D. (2009) "Memetic algorithms for solving job-shop scheduling problems", *Memetic Computing*, 1(1), pp. 69-83.
- Hasgül, S., Saricicek, I., Ozkan, M. and Parlaktuna, O. (2009) "Project-oriented task scheduling for mobile robot team", *Journal of Intelligent Manufacturing*, 20(2), pp. 151-158.
- Haupt, R., (1989) "A survey of priority rule-based scheduling", *OR Spektrum*, 11(1), pp. 3-16.

- Hay, E. J. (1988) *The just-in-time breakthrough: implementing the new manufacturing basics*. Chichester: John Wiley & Sons, Ltd.
- Hayes, R. H. (1981) "Why Japanese factories work", *Harvard Business Review*, July/August, pp. 57-66.
- Heizer and Render (2004) *Principles of operations management*. 5<sup>th</sup> ed. Upper Saddle River, NJ: Pearson Education Inc.
- Heragu, S. S., Graves, R. J., Kim, B.-I. and Onge, A. S. (2002) "Intelligent Agent Based Framework for Manufacturing Systems Control", *IEEE Transactions On Systems, Man, And Cybernetics – Part A: Systems And Humans*, 32(5), pp. 560-573.
- Herer, Y. T. and Shalom, L. (2000) "The Kanban assignment problem - A non-integral approach", *European Journal of Operational Research*, 120(2), pp. 260-276.
- Hill, A. (2012) *Operations Management*. 3<sup>rd</sup> ed. Basingstoke: Palgrave Macmillan.
- Hill, A. and Hill, T. (2011) *Essential operations management*. Basingstoke: Palgrave Macmillan.
- Hines, P., Holweg, M. and Rich, N. (2004) "Learning to evolve - A review of contemporary lean thinking", *International Journal of Operations & Production Management*, 24(10), pp. 994-1011.
- Ho, J. C. and Chang, Y.-L. (2001) "An integrated MRP and JIT framework", *Computers & Industrial Engineering*, 41(2), pp. 173-185.
- Hodgkinson, G. P. and Starbuck, W. H. (2008) *The Oxford handbook of organizational decision making*. Oxford: Oxford University Press, Inc.
- Hodgson, T. J., and Wang, D. (1991) "Optimal hybrid push/pull control strategies for a parallel multi-stage system: Part II", *International Journal of Production Research*, 29(7), pp. 1453–1460.
- Holweg, M. (2007) "The genealogy of lean production", *Journal of Operations Management*, 25(2), pp. 420-437.
- Hop, N. V. (2006) "A heuristic solution for fuzzy mixed-model line balancing problem", *European Journal of Operational Research*, 168(3), pp. 798-810.

- Horng, H.-C. and Cochran, J. K. (2001) "Project surface regions: A decision support methodology for multitasking workers assignment in JIT systems", *Computers & Industrial Engineering*, 39(1-2), pp. 159-171.
- Huang, C.-C. and Kusiak, A. (1996) "Overview of kanban systems", *International Journal of Computer Integrated Manufacturing*, 9(3), pp. 169-189.
- Huang, G. Q., Zhang, Y. F. And Jiang, P. Y. (2008) "RFID-based wireless manufacturing for real-time management of job shop WIP inventories", *International Journal of Advanced Manufacturing Technology*, 36(7-8), pp. 752-764.
- Huang, M., Wang, D. and Ip, W. H. (1998) "Simulation study of CONWIP for a cold rolling plant", *International Journal of Production Economics*, 54(3), pp. 257-266.
- Huang, X.-B. and Yang, L.-X. (2009) Setting lead-time in project manufacturing environment based on CCPM. In: *Proceedings of International Conference on Apperceiving Computing and Intelligence Analysis (ICACIA)*, 23-25 October 2009, Fuzhou, China, pp. 407-410.
- Hum, S.-H. and Lee, C.-K. (1998) "JIT scheduling rules: A simulation evaluation", *Omega International Journal of Management Science*, 26(3), pp. 381-395.
- Huq, Z. and Huq, F. (1994) "Embedding JIT in MRP: The case of job-shops", *Journal of Manufacturing Systems*, 13(3), pp. 153-164.
- Im, J. H. and Lee, S. M. (1989) "Implementation of Just-in-Time systems in US manufacturing firms", *International Journal of Operations & Production Management*, 9(1), pp. 5-14.
- Ioannou, P. A. and Pitsillides, A. (2008) *Modeling and control of complex systems*. Boca Raton, FL: Taylor & Francis Group, LLC.
- Jackson, M. C. (2003) *Systems thinking: Creative holism for managers*. Chichester: John Wiley & Sons, Ltd.
- Jacobs, F. R. and Weston, F. C. (2007) "Enterprise Resource Planning (ERP) – A brief history", *Journal of Operations Management*, 25(2), pp. 357-363.
- Jagdev, H. S. and Browne, J. (1998) "The extended enterprise – a context for manufacturing", *Production Planning & Control*, 9(3), pp. 216-229.

Jahangirian, M., Eldabi, T., Naseer, A. Stergioulas, L. K. and Young, T. (2010) "Simulation in manufacturing and business: A review", *European Journal of Operational Research*. 203(1), pp. 1-13.

Jain, A. S. and Meeran, S. (1999) "Deterministic job-shop scheduling: Past, present and future", *European Journal of Operational research*. 113(2), pp. 390-434.

Jamalnia, A. and Feili, A. (2011) "A simulation testing and analysis of aggregate production planning strategies", *Production Planning & Control: The Management of Operations*, <http://dx.doi.org.v-ezproxy.brunel.ac.uk:2048/10.1080/09537287.2011.631595>.

James-Moore, S.M. and Gibbons, A. (1997) "Is lean manufacture universally relevant? An investigative methodology", *International Journal of Operations & Production Management*, 17(9), pp. 899-911.

Japan Management Association (1989) *Kanban just-in-time at Toyota: Management begins at the workplace*. Revised ed. Portland, OR: Productivity Press.

Jayamohan, M. S. and Rajendran, C. (2000) "New dispatching rules for shop scheduling: A step forward", *International Journal of Production Research*, 38(3), pp. 563-586.

Jayaram, J., Das, A. and Nicolae, M. (2010) "Looking beyond the obvious: Unraveling the Toyota production system", *International Journal of Production Economics*, 128(1), pp. 280-291.

Jenner, R. A. (1998) "Dissipative enterprises, chaos and the principles of lean organisations", *Omega International Journal of Management Science*, 26(3), pp. 397-407.

Jennings, N. R., Sycara, K. P. and Wooldridge, M. (1998) "A roadmap for agent research and development", *Journal of Autonomous Agents and Multi-Agent Systems*, 1(1) pp. 7-36.

Jina, J., Bhattacharya, A. K. and Walton, A. D. (1997) "Applying lean principles for high product variety and low volumes: some issues and propositions", *Logistics Information Management*, 10(1), pp. 5-13.

Jodlbauer, H. and Huber, A. (2008) "Service-level performance of MRP, kanban, CONWIP and DBR due to parameter stability and environmental robustness", *International Journal of Production Research*, 46(8), pp. 2179-2195.



- Johnson, G. A. and Malucci, L. (1999) "Shift to supply chain reflects a more strategic approach", *APICS - The Performance Advantage*, October, pp. 28-31.
- Johnson, G. A., and Malucci, L. (1999) "Shift to supply chain reflects more strategic approach", *APICS- The Performance Advantage*, October(1999), pp. 28-31.
- Jonsson, P. and Mattsson, S.-A. (2002) "The selection and application of material planning methods", *Production Planning & Control*, 13(5), pp. 438-450.
- Jonsson, P. and Mattsson, S.-A. (2008) "Inventory management practices and their implications on perceived planning performance", *International Journal of Production Research*, 46(7), pp. 1787-1812.
- Joseph C. C., Li, Y. and Shady, B. D. (2010) "From value stream mapping toward a lean/sigma continuous improvement process: an industrial case study", *International Journal of Production Research*, 48(4), pp. 1069-1086.
- Junior, M. L. and Filho, M. G. (2010) "Variations of kanban system: Literature review and classification", *International Journal of Production Economics*, 125(1), pp. 13-21.
- Kachitvichyanukul, V. and Sitthitham, S. (2011) "A two-stage genetic algorithm for multi-objective job shop scheduling problems", *Journal of Intelligent Manufacturing*, 22(3), pp. 355-365.
- Kallrath, J. (2002) "Planning and scheduling in the process industry", *OR Spectrum*, 24(3), pp. 219-250.
- Kalos M. H. and Whitlock, P. A. (2009) *Monte Carlo methods*. 2<sup>nd</sup> ed.. Weinheim: Wiley-VCH Verlag.
- Kanet, J. J. and Li, X. (2004) "A weighted modified due date rule for sequencing to minimize weighted tardiness", *Journal of Scheduling*, 7(4), pp. 261-276.
- Kanet, J. J. and Stößlein, M. (2010) "Integrating production planning and control: towards a simple model for Capacitated ERP", *Production Planning & Control: The Management of Operations*, 21(3), pp. 286-300.
- Kara, Y. and Tekin, M. (2009) "A mixed integer linear programming formulation for optimal balancing of mixed-model U-lines", *International Journal of Production Research*, 47(15), pp. 4201-4233.
- Karlsson, C. and Åhlström, P. (1996) "Assessing changes towards lean production", *International Journal of Operations & Production Management*, 16(2), pp. 24-41.

- Katayama, H. and Bennett, D. (1996) "Lean production in a changing competitive world: a Japanese perspective", *International Journal of Production & Operations Management*, 16 (2), pp. 8-23.
- Katayama, H. and Bennett, D. (1999) "Agility, adaptability and leanness: a comparison of concepts and a study of practice", *International Journal of Production Economics*, 60-61(1), pp. 43-51.
- Kazazi, A. (1994) "A method for assessing Just-in-Time effectiveness", *Industrial Management & Data Systems*, 94(7), pp. 14-17.
- Kazazi, A. and Keller, A. Z. (1994) "Benefits derived from JIT by European manufacturing companies", *Industrial Management & Data Systems*, 94(10), pp. 12-14.
- Kelleher, J. P., (1986) JIT Application in the Job-shop Environment. In: *Proceedings of APICS Conference*, pp. 348-351.
- Keller, A. Z. and Kazazi, A. (1993) "Just-in-Time manufacturing systems: A literature review", *Industrial Management & Data Systems*, 93(7), pp. 2-32.
- Kelton, W. D., Sadowski, R. P. and Sadowski, D. A. (2007) *Simulation with Arena*. 4<sup>th</sup> ed. New York, NY: McGraw-Hill.
- Kempf, K., Uzsoy, R., Smith, S. and Gary, K. (2000) "Evaluation and comparison of production schedules", *Computers in Industry*, 42(2/3), pp. 203-220.
- Kesen, S. E. and Baykoç, Ö. F. (2007) "Simulation of automated guided vehicle (AGV) systems based on just-in-time (JIT) philosophy in a job-shop environment", *Simulation Modelling Practice and Theory*, 15 (3), pp. 272–284.
- Khojasteh-Ghamari, Y. (2012) "Developing a framework for performance analysis of a production process controlled by Kanban and CONWIP", *Journal of Intelligent Manufacturing*, 23(1), pp. 61-71.
- King, P. L. (2009) *Lean for the process industries: dealing with complexity*. New York, NY: CRC Press.
- Koenigsberg, E. and McKay, K. N. (2010) "A taxonomy for understanding the origins of production control", *Production Planning & Control*, 21(5), pp. 437-451.
- Koestler, A. (1967) *The ghost in the machine*, London: Arkana Books.

- Koufteros, X. A., Vonderembse, M. A. and Doll, W. J. (1998) "Developing measures of time-based manufacturing", *Journal of Operations Management*, 16(1), pp. 21-41.
- Kouider, A. and Bouzouia, B. (2012) "Multi-agent job shop scheduling system based on co-operative approach of idle time minimisation", *International Journal of Production Research*, 50(2), pp. 409-424.
- Krafcik, M. (1986) *Learning from NUMMI: IMVP Working Paper*. Cambridge, MA: Massachusetts Institute of Technology.
- Krajewski, L. J. and Ritzman, L. P. (2005) *Operations management : processes and value chains*. 7<sup>th</sup> ed. Upper Saddle River, NJ: Pearson Prentice Hall.
- Krajewski, L. J., King, B. E., Ritzman, L. P. and Wong, D. S. (1987) "Kanban, MRP and shaping the manufacturing environment", *Management Science*, 33(1), pp. 39-57.
- Krishnamurthy, A. and Suri, R. (2009) "Planning and implementing POLCA: a card-based control system for high variety or custom engineered products", *Production Planning & Control*, 20(7), pp. 596-610.
- Kumar, C. S. and Panneerselvam, R. (2007) "Literature review of JIT-KANBAN system", *International Journal of Advanced Manufacturing Technology*, 32(3-4), pp. 393-408.
- Kumar, S. and Meade, D. (2002) "Has MRP run its course? A review of contemporary developments in planning systems", *Industrial Management & Data Systems*, 102(8), pp. 453-462.
- Labarthe, O., Espinasse, B., Ferrarini, A., and Montreuil, B. (2007) "Toward a methodological framework for agent-based modelling and simulation of supply chains in a mass customization context", *Simulation Modelling Practice and Theory* 15 (2), pp. 113–136.
- LAI (2012) *Lean enterprise value – Phase I*. MIT, Lean Advancement Initiative. [online] Available at: <<http://lean.mit.edu/about/history/phase-i>> Accessed: [1 October 2012]
- Lambert, A. J. D. (2006) "Generation of assembly graphs by systematic analysis of assembly structures", *European Journal of Operational Research*, 168(3), pp. 931-951.
- Latif, S. and Araz, Ö. U. (2009) "Using the when/where rules in dual resource constrained systems for a hybrid push-pull control", *International Journal of Production Research*, 47(6), pp. 1661-1677.

- Lee, B.-H. and Jo, H.-J. (2007) "The mutation of the Toyota Production System: adapting the TPS at Hyundai Motor Company", *International Journal of Production Research*, 45(16), pp. 3665 -3679.
- Lee, C.-Y., Lin, C.-S., Uzsoy, R. and Wong, C.-C. (1994) "Implementation of a demand-pull system in a job-shop", *International Journal of Production Research*, 32(12), pp. 2915-2957.
- Lee, J.-H. and Kim, C.-O. (2008) "Multi-agent systems applications in manufacturing systems and supply chain management: a review paper", *International Journal of Production Research*, 46(1), pp. 233-265.
- Lei, D. (2009) "Multi-objective production scheduling: a survey", *International Journal of Advanced Manufacturing Technology*, 43(9-10), pp. 926-938.
- Leitão, P. (2009) "Agent-based distributed manufacturing control: A state-of-the-art survey", *Engineering Applications of Artificial Intelligence*. 22(7), pp. 979–991.
- Leitão, P. and Restivo, F. (2006) "ADACOR: A holonic architecture for agile and adaptive manufacturing control", *Computers in Industry*, 57(2), pp. 121–130.
- Leung, J. Y-T. (2004) *Handbook of scheduling: Algorithms models and performance analysis*. Boca Raton, FL: Chapman & Hall/CRC, Taylor & Francis Group LLC.
- Levasseur, G. A. and Storch, R. L. (1996) "A non-sequential just-in-time simulation model", *Computers in Industrial Engineering*, 30(4), pp. 741-752.
- Levinson, W.A. and Rerick, R.A. (2002) *Lean Enterprise: A Synergistic Approach to Minimizing Waste*. Milwaukee, WI: American Society for Quality.
- Levitin, G., Rubinovitz, J. and Shnits, B. (2006) "A genetic algorithm for robotic assembly line balancing", *European Journal of Operational Research*, 168(3), pp. 811-825.
- Lewis, M. A. (2000) "Lean production and sustainable competitive advantage", *International Journal of Operations & Production Management*, 20(8), pp. 959-978.
- Li et al. (2007) A unified model-based integration of process planning and scheduling. In: Wang, L. and Shen, W. eds. (2007) *Process planning and scheduling for distributed manufacturing*, London: Springer-Verlag Limited.

- Li, J., Gao, J. and Sun, L. (2012) "Sequencing minimum product sets on mixed-model U-lines to minimise work overload", *International Journal of Production Research*, 50(18), pp. 4977-4993.
- Li, J.-Q., Pan, Q.-K., Suganthan, P. N. and Chua, T. J. (2011) "A hybrid tabu search algorithm with an efficient neighborhood structure for the flexible job shop scheduling problem", *International Journal of Advanced Manufacturing Technology*, 52(5-8), pp. 683-697.
- Li, J.-W. (2003) "Simulation-based comparison of push and pull systems in a job-shop environment considering the context of JIT implementation", *International Journal of Production Research*, 41(3), pp.4683-4699.
- Li, J.-W. (2005) "Investigating the efficacy of exercising JIT practices to support pull production control in a job shop environment", *Journal of Manufacturing Technology Management*, 16(7), pp. 765-783.
- Li, J.-W. (2010) "Simulation study of coordinating layout change and quality improvement for adapting job shop manufacturing to CONWIP control", *International Journal of Production Research*, 48(3), pp. 879-900.
- Li, J.-W. and Barnes, D. J. (2000) "Investigating the Factors Influencing the Shop Performance in a Job-shop Environment with Kanban-based Production Control", *International Journal of Production Research*, 38(18), pp.4683-4699.
- Li, M., Li, Q., Guo, Q., Ding, R. and Xin, T. (2009) Scheduling optimization of refinery operations based on production continuity. In: *Proceedings of IEEE International Conference on Automation and Logistics (ICAL)*, 5-7 August 2009, Jinan, China, pp. 1807 - 1811.
- Li, Z. and Ierapetritou, M. (2008) "Process scheduling under uncertainty: Review and challenges", *Computers and Chemical Engineering*, 32 (4-5), pp. 715-727.
- Lian, K., Zhang, C., Gao, L. and Shao, X. (2012) "A modified colonial competitive algorithm for the mixed-model U-line balancing and sequencing problem", *International Journal of Production Research*, 50(18), pp. 5117-513.
- Liang, T.-F. (2007) "Application of interactive possibilistic linear programming to aggregate production planning with multiple imprecise objectives", *Production Planning & Control: The Management of Operations*, 18(7), pp. 548-560.

Liberopoulos, G. and Dallery, Y. (2000) "A unified framework for pull control mechanisms in multi-stage manufacturing systems", *Annals of Operations Research*, 93(1-4), pp. 325-355.

Liberopoulos, G. and Koukoumialos, S. (2005) "Tradeoffs between base stocklevels, numbers of kanbans, and planned supply lead times in production/inventory systems with advance demand information", *International Journal of Production Economics*, 96(2), pp. 213-232.

Liker, J. K. (1998) *Becoming lean: Inside stories of US manufacturers*. Portland, OR: Productivity Press.

Liker, J. K. (2004) *The Toyota way: 14 management principles from the world's greatest manufacturer*, New York, NY: McGraw-Hill.

Liker, J. K. and Meier, D. (2005) *The Toyota way fieldbook: A practical guide for implementing Toyota's 4ps*, New York, NY: McGraw-Hill.

Lim, M. K. and Zhang, D. Z. (2004) "An integrated agent-based approach for responsive control of manufacturing resources", *Computers & Industrial Engineering*, 46 (2), pp. 221–232.

Lim, M. K. and Zhang, Z. (2003) "A multi-agent based manufacturing control strategy for responsive manufacturing", *Journal of Materials Processing Technology*, 139 (1-3) 379–384.

Liu, N., Abdelrahman, M. A. and Ramaswamy, S. (2007) "A Complete Multiagent Framework for Robust and Adaptable Dynamic Job Shop Scheduling", *IEEE Transactions on Systems, Man, and Cybernetics – Part C: Applications and Reviews*, 37(5), pp. 904-916.

Lou, P., Liu, Q., Zhou, Z., Wang, H., and Sun, S. X. (2012) "Multi-agent-based proactive–reactive scheduling for a job shop", *International Journal of Advanced Manufacturing Technology*, 59(1-4), pp. 311–324.

Lou, P., Ong, S.K. and Nee, A.Y.C. (2010) "Agent-based distributed scheduling for virtual job shops", *International Journal of Production Research*, 48(13), pp. 3889-3910.

Luck, M., Ashri, R. and D'Inverno, M. (2004) *Agent-based software development*. Norwood, MA: Artech House Inc.

- Luger, G. F. (2009) *Artificial intelligence: Structures and strategies for complex problem solving*. 6<sup>th</sup> ed. Boston, MA: Pearson Education Inc.
- Luh, P. B., Zhou, X. and Tomastik R. N. (2000) "An effective way to reduce inventory in jobshops", *IEEE Transactions on Robotics and Automation*, 16(4), pp. 420-424.
- Macal, C. M. and North, M. J. (2010) "Tutorial on agent-based modelling and simulation", *Journal of Simulation*, 4(3), pp. 151-162.
- Maravelias, C. T. (2012) "General framework and modeling approach classification for chemical production scheduling", *American Institute of Chemical Engineers Journal*, 58(6), pp. 1812-1828.
- Marek, R. P., Elkins, D. A. and Smith, D. R. (2001) Understanding the fundamentals of Kanban and CONWIP pull systems using simulation. In: *Proceedings of Winter Simulation Conference (WSC)*, December 9-12, 2001, Arlington, VA, US, pp. 921-929.
- Mařík, V. and McFarlane, D. (2005) "Industrial adoption of agent-based technologies", *Intelligent Manufacturing Control*, 20(1), pp. 27-35.
- Markland, R. E., Vickery, S. K. and Davis, R. A. (1998) *Operations management concepts in manufacturing and services*, 2<sup>nd</sup> ed. Cincinnati, Ohio: South-Western College Publishing.
- Martin, A. D., Chang, T. M., Yih, Y. and Kincaid, R. K. (1998) "Using tabu search to determine the number of kanbans and lotsizes in a generic kanban system", *Annals of Operations Research*, 78(1), pp. 201 – 217.
- Martinich, J. S. (1996) *Production and operations management: An applied modern approach*. New York, NY: John Wiley & Sons, Inc.
- Martin-Vega, L. A., Pippin, M., Gerdon, E. and Burcham, R. (1989) "Applying just-in-time in a wafer fab: A case study", *IEEE Transactions on Semiconductor Manufacturing*, 2(1), pp. 16-22.
- Mascardi, V., Martelli, M. and Gungui, I. (2008) *DCaseLP: A prototyping environment for multi-language agent systems*. In: Dastani, M., Seghrouchni, A. E. F., Leite, J. and Torroni, P. eds. (2008) *Languages, methodologies and development tools for multi-agent systems: LNAI 5118*, Berlin, Heidelberg: Springer-Verlag, pp. 139-155.
- Matsui, Y. (2007) "An empirical analysis of just-in-time production in Japanese manufacturing companies", *International Journal of Production Economics*, 108 (1-2) pp. 153-164.

McLachlin, R. (1997) "Management initiatives and just-in-time manufacturing", *Journal of Operations Management*, 15(4), pp. 271-292.

Meeran, S. and Morshed, M. S. (2012) "A hybrid genetic tabu search algorithm for solving job shop scheduling problems: a case study", *Journal of Intelligent Manufacturing*, 23(4), pp. 1063-1078.

Mehra, S. and Inman, R. A. (1992) "Determining the critical elements of Just-in-Time implementation", *Decision Sciences*, 23(1), pp. 160-174.

Meyers, R. A. (2010) *Complex systems in finance and econometrics*. New York, NY: Springer Science+Business Media, LLC.

Mingers, J. (2006) "A Critique of Statistical Modelling in Management Science from a Critical Realist Perspective: Its Role within Multimethodology", *The Journal of the Operational Research Society*, 57(2), pp. 202-219.

Miyashita, K. and Rajesh, G. (2010) "Multiagent coordination for controlling complex and unstable manufacturing processes", *Expert Systems with Applications*, 37(3), pp. 1836–1845.

Mohammad D. Al-Tahat, M. D. and Rawabdeh, I. A. (2008) "Stochastic analysis and design of CONWIP controlled production systems", *Journal of Manufacturing Technology Management*, 19(2), pp. 253-273.

Mohanty, R. P., Kumar, S., and Tiwari, M.K. (2003) "Expert enhanced coloured fuzzy petri net models of traditional, flexible and reconfigurable kanban systems", *Production Planning & Control*, 14 (5), pp. 459-477.

Mollenkopf, D., Stolze, H., Tate, W. L. and Ueltschy, M. (2010) "Green, lean and global supply chains", *International Journal of Physical Distribution and Logistics Management*, 40(1/2), pp. 14-41.

Mönch, L. (2007) "Simulation-based benchmarking of production control schemes for complex manufacturing systems", *Control Engineering Practice*, 15 (11), pp. 1381–1393.

Monden Y. (1983b) *Applying Just In Time*. Norcross, GA: Industrial Engineering and Management Press.

Monden, Y. (1981a) "What makes the Toyota Production System really tick", *Industrial Engineering*, January, pp. 36-46.



- Monden, Y. (1981b) "Adaptable kanban system helps Toyota maintain Just-in-Time production", *Industrial Engineering*, May, pp. 29-46.
- Monden, Y. (1981c) "Toyota's production smoothing methods: Part II", *Industrial Engineering*, September, pp. 22-30.
- Monden, Y. (1981d) "How Toyota shortened supply lot production time, waiting time and conveyance time", *Industrial Engineering*, 13(9), pp. 22-29.
- Monden, Y. (1983a) *The Toyota Production System*. Portland, OR: Productivity Press.
- Monden, Y. (1998) *Toyota Production System: An integrated approach to Just-in-Time*, 3<sup>rd</sup> ed. Norcross, GA: Industrial Engineering and Management Press.
- Monostori, L., Váncza, J. and Kumara, S.R.T. (2006) "Agent-based systems for manufacturing", *Annals of the CIRP*, 55(2), pp. 697-720.
- Morton, T. E. (1999) *Production operations management*. Cincinnati, OH: South-Western College Publishing.
- Moscoso, P. G., Fransoo, J. C. and Fischer, D. (2010) "An empirical study on reducing planning instability in hierarchical planning systems" *Production Planning & Control: The Management of Operations*, 21(4), pp. 413-426.
- Moslehi, G. and Mahnam, M. (2011) "A Pareto approach to multi-objective flexible job-shop scheduling problem using particle swarm optimization and local search", *International Journal Production Economics*, 129 (1), pp. 14-22.
- Mouhoub, N. E., Benhocine, A. and Belouadah, H. (2011) "A new method for constructing a minimal PERT network", *Applied Mathematical Modelling*, 35 (9), pp. 4575-4588.
- Mukhopadhyay, S. K. and Shanker, S. (2005) "Kanban implementation at a tyre manufacturing plant: a case study", *Production Planning & Control*, 16(5), pp. 488-499.
- Mula, J., Poler, R., Garcia-Sabater, J. P. and Lario, F. C. (2006) "Models for production planning under uncertainty: A review", *International Journal of Production Economics*, 103(1), pp. 271-285.
- Müller, E., Tolujew, J. and Kienzle, F. (2012) "Push-kanban – a kanban-based production control concept for job-shops", *Production Planning & Control: The Management of Operations*, <http://dx.doi.org/10.1080/09537287.2012.7010211>.

- Nagendra, P. B. and Das, S. K. (1999) "MRP/Sfx: A kanban-oriented shop floor extension to MRP", *Production Planning & Control*, 10 (3), pp. 207-218.
- Nakamura, M., Sakakibara, S. and Schroeder, R. (1996) "Japanese manufacturing method at US manufacturing plants: empirical evidence", *Canadian Journal of Economics*, 29, pp. S468-S474.
- Narasimhan, R., Swink, M. and Kim, S. W. (2006) "Disentangling leanness and agility: An empirical investigation", *Journal of Operations Management*, 24(5), pp. 440-457.
- Naylor, J. (2002) *Introduction to operations management*. Harlow: Financial Times Prentice Hall.
- Naylor, J. B., Naim, M. M. and Berry, D. (1999) "Leagility: Integrating the lean and agile manufacturing paradigms in the total supply chain", *International Journal of Production Economics*, 62(1), pp. 107-118.
- New, S. J. (2007) "Celebrating the enigma: the continuing puzzle of the Toyota Production System", *International Journal of Production Research*, 45(16), pp. 3545-3554.
- Ngai, E.W.T., Law, C.C.H. and Wat, F.K.T. (2008) "Examining the critical success factors in the adoption of enterprise resource planning", *Computers in Industry*, 59 (6), pp. 548–564.
- Nguyen, G.T., Dang, T.T., Hluchy, L., Balogh, Z., Laclavik, M. and Budinska, I. (2002) *Agent platform evaluation and comparison, Technical Report for Pellucid 5FP IST-2001-34519*. Bratislava, Slovakia. [online] Available at: <<http://ups.savba.sk/~giang/publication/TR-2002-06.pdf>> [Accessed: 1 october 2012]
- Nguyen, N. T. and Jain, L. C. eds. (2009) *Intelligent agents in the evolution of web and applications*. Berlin-Heidelberg: Springer-Verlag.
- Nightingale, D. (2009) Principles of Enterprise Systems. In: *Proceedings of 2<sup>nd</sup> International Symposium on Engineering Systems MIT*, 15-17 June, Cambridge, Massachusetts, US, pp. 1-11.
- Nightingale, D. J. and Mize, J. H. (2002) "Development of a lean enterprise transformation maturity model", *Information, Knowledge, Systems Management*, 3(1) pp. 15-30.

Nightingale, D., Srinivasan, J. and Mize, J. H. (2010) Enterprise Transformation Roadmap. [online] Available at: <[http://lean.mit.edu/component/docman/search\\_result?search\\_phrase=nightingale&catid=0&ordering=newest&search\\_mode=any&search\\_where\[\]=search\\_name&search\\_where\[\]=search\\_description](http://lean.mit.edu/component/docman/search_result?search_phrase=nightingale&catid=0&ordering=newest&search_mode=any&search_where[]=search_name&search_where[]=search_description)> Accessed: [1 October 2012]

Nomura, J. and Takakuwa, S. (2004) Module-based modeling of flow-type multistage manufacturing systems adopting dual-card kanban system. In: *Proceedings of the Winter Simulation Conference (WSC)*, 5-8 December, Nagoya, Japan, pp. 1065-1072.

North, M. J. and Macal, C. M. (2007) *Managing business complexity: Discovering strategic solutions with agent-based modelling and simulation*. New York, NY: Oxford University Press, Inc.

Ohno, T. (1988) *Toyota Production System: Beyond large-scale production*. Portland, OR: Productivity Press.

Olhager, J. and Prajogo, D. (2012) "The impact of manufacturing and supply chain improvement initiatives: A survey comparing make-to-order and make-to-stock firms", *Omega*, 40(2), pp. 159-165.

Oliver, N., Schab, L. and Holweg, M. (2007) "Lean principles and premium brands: conflict or complement?", *International Journal of Production Research*, 45(16), pp. 3723-3739.

Otenti, S., (1991) A modified kanban system in a semiconductor manufacturing environment. In: *Proceedings of Advanced Semiconductor Manufacturing Conference and Workshop IEEE (ASCM)*, 21-23 October, Hudson, MA, US, pp. 43-45.

Ouelhadj, D. and Petrovic, S. (2009) "A survey of dynamic scheduling in manufacturing systems", *Journal of Scheduling*, 12(4), pp. 417-431.

Ounnar, F. and Pujo, P. (2012) "Pull control for job shop: holonic manufacturing system approach using multicriteria decision-making", *Journal of Intelligent Manufacturing*, 23(1), pp. 141-153.

Özbayrak, M. and Papadopoulou, T. C. (2004) Lean scheduling in HV/LV environment. In: *Proceedings of the 14th International Conference on Flexible Automation & Intelligent Manufacturing (FAIM)*, 12-14 July, 2004, Toronto, Canada, pp. 703-710.

Özbayrak, M., Papadopoulou, T. C. and Samaras, E. (2006) "A Flexible and Adaptable Planning and Control System for an MTO Supply Chain System", *Robotics and Computer-Integrated Manufacturing*, 22(5/6), pp.557-565.

Özcan, U., Kellegöz, T. and Toklu, B. (2011) "A genetic algorithm for the stochastic mixed-model U-line balancing and sequencing problem", *International Journal of Production Research*, 49(6), pp. 1605-1626.

Panizzolo, R., Garengo, P., Sharma, M. K. and Gore, A. (2012) "Lean manufacturing in developing countries: evidence from Indian SMEs", *Production Planning & Control: The Management of Operations*, 23(10-11), pp. 769-788.

Panurak, H. V. D. (2000) "A practitioners' review of industrial agent applications", *Autonomous Agents and Multi-Agent Systems*, 3(4), pp. 389-407.

Paolucci, M. and Sacile, R. (2005) *Agent-based manufacturing and control systems: New agile manufacturing solutions for achieving peak performance*. Boca Raton, FL: CRC Press LLC.

Papadopoulou, T. and Özbayrak, M. (2005) "Leanness: experiences from the journey to date", *Journal of Manufacturing Technology Management*, 16(7), pp. 784-807.

Papadopoulou, T. C. and Mousavi, A. (2007b) Performance modelling of dynamic lean job-shops with basestock shop-floor control using intelligent software agents. In: *Proceedings of 5th International Conference on Manufacturing Research (ICMR)*, 11-13 September 2007, Leicester, UK.

Papadopoulou, T. C. and Mousavi, A. (2008) Scheduling of non-repetitive lean manufacturing systems under uncertainty using intelligent agent simulation. In: *Proceedings of 6th International Conference on Manufacturing Research (ICMR)*, 9-11 September 2008, Uxbridge, UK.

Papadopoulou, T. C., Mousavi, A. and Broomhead, P. (2007) Extending lean pull production to the dynamic scheduling of high variety low volume systems using intelligent agents. In: *Proceedings of 17th International Conference on Flexible Automation and Intelligent Manufacturing (FAIM)*, 18 - 20 June 2007, Philadelphia, US.

Papadopoulou, T.C. and Mousavi, A. (2007a) "Control of constant work-in-progress in dynamic lean job shops using a multi-agent system approach", *International Journal of Agile Manufacturing*, 10(2), pp. 19-28.

- Parnell, G. S., Driscoll, P. J. and Henderson, D. L. (2008) *Decision making in systems engineering and management*. Hoboken, NJ: John Wiley & Sons, Inc.
- Parry, G. C. and Turner, C. E. (2006) "Application of lean visual process management tools", *Production Planning & Control*, 17(1), pp. 77- 86.
- Pastor, R. and Ferrer, L. (2009) "An improved mathematical program to solve the simple assembly line balancing problem", *International Journal of Production Research*, 47(11), pp. 2943-2959.
- Paton, S., Clegg, B., Hsuan, J. and Pilkington, A. (2011) *Operations Management*. London: McGraw-Hill Higher Education.
- Pattanaik, L. N. and Sharma B. P. (2009) "Implementing lean manufacturing with cellular layout: a case study", *International Journal of Advanced Manufacturing Technology*, 42(7-8), pp. 772-779.
- Pavnaskar, S. J. , Gershenson, J. K. and Jambekar, A. B.(2003) "Classification scheme for lean manufacturing tools", *International Journal of Production Research*, 41(13), pp. 3075-3090.
- Peeters, M. and Degraeve, Z. (2006) "An linear programming based lower bound for the simple assembly line balancing problem", *European Journal of Operational Research*, 168(3), pp. 716-731.
- Pegels, C. C. (1984) "The Toyota Production System – Lessons for American Management", *International Journal of Operations & Production Management*, 4(1), pp. 3-11.
- Petroni, A. (2002) "Critical factors of MRP implementation in small and medium-sized firms", *International Journal of Operations & Production Management*, 22(3), pp. 329-348.
- Pham, D. T., Eldukhri, E. E. and Soroka, A. J. eds. (2006) *Intelligent production machine and systems: Proceedings of the 2<sup>nd</sup> Virtual International Conference on Intelligent Production Machines and Systems*. Oxford: Elsevier Science.
- Pichappan, P., Ahmadi, H. and Ariwa, E. eds. (2011) *Innovative computing technology: First international conference, INCT 2011, Tehran, Iran, December 2011 Proceedings*. Berlin-Heidelberg: Springer-Verlag.
- Pickardt, C. W. and Branke, J. (2011) "Setup-oriented dispatching rules – a survey", *International Journal of Production Research*, 50(20), pp. 5823-5842.

- Pidd, M. (2004) *Computer simulation in management science*. 5<sup>th</sup> ed. Chichester: John Wiley & Sons, Ltd.
- Pidd, M. (2009) *Tools for thinking: Modelling in management science*. 3<sup>rd</sup> ed. Chichester: John Wiley & Sons, Ltd.
- Pil, F. K. and Fujimoto, T. (2007) "Lean and reflective production: the dynamic nature of production models", *International Journal of Production Research*, 45(16), pp. 3741-376.
- Pinedo, M. and Chao, X. (1999) *Operations scheduling with applications in manufacturing and services*. Boston, MA: Irwin/McGraw-Hill.
- Pinedo, M. L. (2009) *Planning and scheduling in manufacturing and services*. 2<sup>nd</sup> ed. New York, NY: Springer Science+Business Media, LLC.
- Plenert, G. (1999) "Focusing material requirements planning (MRP) towards performance", *European Journal of Operational Research*, 119(1), pp. 91-99.
- Porter, K., Little, D., Peck, M. and Rollins, R. (1999) "Manufacturing classifications: relationships with production control systems", *Integrated Manufacturing Systems*, 10(4), pp. 189-198.
- Portioli-Staudacher, A. and Tantardini, M. (2012) "A lean-based ORR system for non-repetitive Manufacturing", *International Journal of Production Research*, 50(12), pp. 3257-3273.
- Pujo, P. and Ounnar, F. (2008) Pull system control for job shop via a holonic, isoarchic and multicriteria approach. In: *Proceedings of the 17<sup>th</sup> World Congress The International Federation of Automatic Control*, 6-11 July, 2008, Seoul, Korea, pp. 15799-15804.
- Qi Hao, Q., Shen, W. and Zhang, Z. (2005) "An autonomous agent development environment for engineering applications", *Advanced Engineering Informatics*, 19(2), pp. 123-134.
- Ragsdale, C. T. (2008) *Managerial decision modelling*. Revised ed. Mason, OH: Thomson South-Western.
- Rajabinasab, A. and Mansour, S. (2011) "Dynamic flexible job shop scheduling with alternative process plans: an agent-based approach", *International Journal of Advanced Manufacturing Technology*, 54(), pp. 1091-1107.

- Rajendran, C. and Holthaus, O. (1999) "A comparative study of dispatching rules in dynamic flowshops and jobshops", *European Journal of Operational Research*, 116(1), pp. 156-170.
- Ramanan, G. V. and Rajendran, C. (2003) "Scheduling in kanban-controlled flowshops to minimise the makespan of containers", *International Journal of Advanced Manufacturing Technology*, 21(5), pp. 348-354.
- Ramasesh, R., (1990) "Dynamic job shop scheduling: A survey of simulation research", *OMEGA*, 18(1), pp. 43-57.
- Rappold, J. A. and Yoho, K. D. (2008) "A model for level-loading production in the process industries when demand is stochastic", *Production Planning & Control: The Management of Operations*, 19(7), pp. 686-701.
- Rashid, M. F. F. R., Hutabarat, W. and Tiwari, A. (2012) "A review on assembly sequence planning and assembly line balancing optimisation using soft computing approaches", *International Journal of Advanced Manufacturing Technology*, 59(1-4), pp. 335-349.
- Ravindran, A. and Kempf, K. G. (2011) "Production planning with load-dependent lead times and safety stocks for a single product", *International Journal of Planning and Scheduling*, 1(1-2), pp. 58-89.
- Razmi, J., Rahnejat, H. and Khan, M. K. (1998) "Use of analytic hierarchy process approach in classification of push, pull and hybrid push-pull systems for production planning", *International Journal of Operations & Production Management*, 18(11), pp. 1134-1151.
- Reichhart, A. and Holweg, M. (2007) "Lean distribution: concepts, contributions, conflicts", *International Journal of Production Research*, 45(16), pp. 3699-3722.
- Renna, P. (2010) "Job shop scheduling by pheromone approach in a dynamic environment", *International Journal of Computer Integrated Manufacturing*, 23(5), pp. 412-424.
- Ribas, I., Leisten, R. and Framiñan, J. M. (2010) "Review and classification of hybrid flow shop scheduling problems from a production system and a solutions procedure perspective", *Computers & Operations Research*, 37 (8), pp. 1439-1454.
- Riezebos, J., Klingenberg, W. and Hicks, C. (2009) "Lean Production and information technology: Connection or contradiction?", *Computers in Industry*, 60(4), pp. 237-247.

- Robinson, R. and Richards, R. (2010) Critical chain project management: Motivation and overview. In: *Proceedings of IEEE Aerospace Conference*, 6-13 March, Port Orchard, WA, US, pp. 1-10.
- Roy, R. N. (2005) *A modern approach to operations management*. New Delhi: New Age International (P) Limited.
- Russell and Taylor (2009) *Operations management along the supply chain*, 6<sup>th</sup> ed. Chichester: John Wiley & Sons, Ltd.
- Russell, S. and Norvig, P. (2010) *Artificial intelligence: A modern approach*. 3<sup>rd</sup> ed. Upper Saddle River, NJ: Pearson Education, Inc.
- Ryan, S. M. and Choobineh, F. F (2003) "Total WIP and WIP mix for a CONWIP controlled job-shop", *IIE Transactions*, 35(5), pp. 405-418.
- Ryan, S. M., Baynat, B. and Choobineh, F. F. (2000) "Determining inventory levels in a CONWIP controlled job shop", *IIE Transactions*, 32(2), pp. 105-114.
- Sacks, R., Radosavljevic, M. and Barak, R. (2010) "Requirements for building information modeling based lean production management systems for construction" *Automation in Construction*, 19(5), pp. 641-655.
- Sage, A. P. and Rouse, W. B. (2011) *Handbook of systems engineering and management*. 2<sup>nd</sup> ed. Hoboken, NJ: John Wiley & Sons, Inc.
- Sakakibara, S., Flynn, B. B., Schroeder, R. G. and Morris, W. T. (1997) "The impact of Just-in-Time Manufacturing and its infrastructure on manufacturing performance", *Management Science*, 43(9), pp. 1246-1257.
- Saltelli, A., Ratto, M., Andres, T., Campolongo, F., Cariboni, J., Gatelli, G., Saisana, M. and Tarantola, S. (2008) *Global sensitivity analysis: The primer*. Chichester: John Wiley & Sons, Ltd.
- Samaranayake, P. and Toncich, D. (2007) "Integration of production planning, project management and logistics systems for supply chain management, *International Journal of Production Research*, 45(22), pp. 5417-5447.
- Sánchez, A. M. and Pérez, M. P. (2001) "Lean indicators and manufacturing strategies", *International Journal of Operations & Production Management*, 21(11), pp. 1433-1451.
- Sandras, B. (1985) Job-shop JIT. In: *Proceedings of APICS Conference*, pp. 448-452.



- Santa-Eulalia, L.A., Ait-Kadi, D., D'Amours, S., Frayret, J.-M. and Lemieux, S. (2011) "Agent-based experimental investigations of the robustness of tactical planning and control policies in a softwood lumber supply chain", *Production Planning & Control: The Management of Operations*, 22(8), pp. 782-799.
- Sarin, S. C., Greco, M. P. and Dar-El, E. M. (2006) "Sequencing and loading of products on a flowline", *European Journal of Operational Research*, 168(3), pp. 905-921.
- Sarin, S. C., Varadarajan, A. and Wang, L. (2011) "A survey of dispatching rules for operational control in wafer fabrication", *Production Planning & Control: The Management of Operations*, 22(1), pp. 4-24.
- Sarker, R. A. and Newton, C. S. (2008) *Optimisation modelling: a practical introduction*. Boca Raton, FL: Taylor & Francis Group, LLC.
- Scallan, P. (2003) *Process planning: the design/manufacture interface*, Oxford: Elsevier Science Technology.
- Scherrer-Rathje, M., Boyle, T. A. and Deflorin, P. (2009) "Lean, take two! Reflections from the second attempt at lean implementation", *Business Horizons*, 52(1), pp. 79-88.
- Schneeweiss, C. (2003) *Distributed decision making*. 2<sup>nd</sup> ed. Berlin-Heidelberg: Springer-Verlag.
- Scholl, A. and Becker, C. (2006) "State-of-the-art exact and heuristic solution procedures for simple assembly line balancing", *European Journal of Operational Research*, 168 (3), pp. 666-693.
- Schonberger, R. J. (1982a) *Japanese manufacturing techniques*, New York, NY: Free Press.
- Schonberger, R. J. (1982b) "Some observations on the advantages and implementation issues of Just-in-Time production systems", *Journal of Operations Management*, 3(1), pp. 1-11.
- Schönsleben, P. (2007) *Integral logistics management: Operations and supply chain management in comprehensive value-added networks*, 3<sup>rd</sup> ed. Boca Raton: FL, Taylor & Francis Group LLC.
- Schuldt, A. (2011) *Multiagent coordination enabling autonomous logistics*. Berlin-Heidelberg: Springer-Verlag.

- Segerstedt, A. (2006) "Master Production Scheduling and a comparison of Material Requirements Planning and cover-time planning, *International Journal of Production Research*, 44(18-19), pp. 3585-3606.
- Selçuk, B., Ivo J. B. F. Adan, I. J. B. F., De Kok, T. G. and Fransoo, J. C. (2009) "An explicit analysis of the lead time syndrome: stability condition and performance evaluation", *International Journal of Production Research*, 47(9), pp. 2507-2529.
- Sha, D.Y. and Hsu, C. Y. (2006) "A hybrid particle swarm optimization for job shop scheduling problem", *Computers & Industrial Engineering*, 51(4), pp. 791-808.
- Shah, R. and Ward, P. T. (2003) "Lean manufacturing: context, practice bundles, and performance", *Journal of Operations Management*, 21(2) pp. 129-149.
- Shah, R. and Ward, P. T. (2007) "Defining and developing measures of lean production", *Journal of Operations Management*, 25(4), pp. 785-805.
- Shajari, M. and Ghorbani, A. A. (2004) Application of Belief-Desire-Intention agents in intrusion detection and response. In: *Proceedings of 2<sup>nd</sup> Annual Conference on Privacy, Security and Trust*, October 13-15, 2004, New Brunswick, Canada pp. 181-191.
- Shapiro, J. F. (2007) *Modelling the supply chain*. 2<sup>nd</sup> ed. Belmont, CA: Thomson-Brooks/Cole.
- Sharifi, H. and Zhang, Z. (1999) "A methodology for achieving agility in manufacturing organisations: An introduction", *International Journal of Production Economics*, 62(1-2), pp. 7-22.
- Sharma, S. and Agrawal, N. (2009) "Selection of a pull production control policy under different demand situations for a manufacturing system by AHP-algorithm", *Computers & Operations Research*, 36(5), pp. 1622-1632.
- Sharon, A., De Weck, O. L. and Dori, D. (2011) "Project management vs. systems engineering management: A practitioners' view on integrating the project and product domains", *Systems Engineering*, 14(4), pp. 427-440.
- Shen, L. and Buscher, U. (2012) "Solving the serial batching problem in job shop manufacturing systems", *European Journal of Operational Research*, 221(1), pp. 14-26.

- Shen, W., Hao, Q., Yoon, H. J. and Norrie, D. H. (2006) "Applications of agent-based systems in intelligent manufacturing: An updated review", *Advanced Engineering Informatics*, 20 (4) pp. 415-431.
- Shendarkar, A., Vasudevan, K., Lee, S. and Son, Y.-S. (2006) Crowd simulation for emergency response using BDI agent based on virtual reality. In: *Proceedings of the Winter Simulation Conference (WSC)*, 3-6 December, 2006, Tucson, AZ, US, pp. 545-553.
- Shingo, S. (1981) *A study of the Toyota Production System from an industrial engineering viewpoint*. Tokyo: Japan Management Association.
- Shobrys, D. E. and White, D. C. (2002) "Planning, scheduling and control systems: why cannot they work together", *Computers and Chemical Engineering*, 26(2), pp.149-160.
- Silver E. A., Pyke, D. F., Peterson, R. (1988) *Inventory management and production planning and scheduling*. 3<sup>rd</sup> ed. Chichester: John Wiley & Sons, Ltd.
- Simard, R. (2012) SSJ: Stochastic simulation in JAVA, Université de Montréal. [online] Available at: <<http://www.iro.umontreal.ca/~simardr/ssj/indexe.htm>> [Accessed: 1 October 2012]
- Simaria, A.S. and Xambre, A.R., Filipe, N.A. and Vilarinho, P.M. (2010) *A Decision Support System for Assembly and Production Line Balancing*. In: Gunasekaran, A and Sandhu, M, eds. (2010) *Handbook on Business Information Systems*. Singapore: World Scientific Publishing Company. pp. 383 - 414.
- Singh, n. and Brar, J. K. (1992) "Modelling and analysis of Just-in-Time manufacturing systems: A review", *International Journal of Operations & Production Management*, 12(2), pp. 3-14.
- Sipper, D. and Bulfin, R. L. (1997) *Production: planning, control and integration*. New York, NY: McGraw-Hill.
- Skinner, W. (1974) "The focused factory: New approach to managing manufacturing sees our productivity crisis as the problem of how to complete", *Harvard Business Review*, May/June, 113-121.
- Slack, N. and Lewis, M. (2011) *Operations strategy*. 3<sup>rd</sup> ed. Harlow: Pearson Education Limited.
- Slack, N., Chambers, S. and Johnston, R. (2010) *Operations management*. 6<sup>th</sup> ed. Harlow: Pearson Education Ltd.

- Smith, N. J. (2008) *Engineering project management*. 3<sup>rd</sup> ed. Oxford: Blackwell Publishing.
- Sokolowski, J. A and Banks, C. M. (2009) *Principles of modeling and simulation: A multidisciplinary approach*. Hoboken, NJ: John Wiley & Sons, Inc.
- Soriano-Meier, H. and Forrester, P. L. (2002) "A model for evaluating the degree of leanness of manufacturing firms", *Integrated Manufacturing Systems*, 13(2), pp. 104-109.
- Sotskov, Y. N., Dolgui, A. and Portmann, M.-C. (2006) "Stability analysis of an optimal balance for an assembly line with fixed cycle time", *European Journal of Operational Research*, 168(3), pp. 783-797.
- Spear, S. and Bowen, H. K. (1999) "Decoding the DNA of the Toyota Production System", *Harvard Business Review*, September/October, pp. 96-106.
- Spear, S. J. (2002) "The essence of Just-in-Time: embedding diagnostic tests in worksystems to achieve operational excellence", *Production Planning & Control*, 13(8), pp. 754-767.
- Spearman, M. L. (1988) *An analytic congestion model for closed production systems*, *Technical Report 88-23*. Evanston, IL: Northwestern University.
- Standard, C. and Davis, D. (1999) *Running today's factory - A proven strategy for lean manufacturing*. Dearborn, MI: Society of Manufacturing Engineers.
- Sterman, J. D. (1991). *A skeptic's guide to computer models*. In Barney, G. O., Kreutzer, W. B. and Garretteds, M. J. (1991) *Managing a Nation: The Microcomputer Software Catalog*. Boulder, CO: Westview Press. pp. 209-229.
- Stevenson, W. J. (2006) *Operations management*. 9<sup>th</sup> ed. New York, NY: McGraw-Hill/Irwin.
- Stockton, D. J. and Lindley, R. J. (1995) "Implementing Kanbans within High Variety/Low Volume Manufacturing Environments", *International Journal of Operations & Production Management*, 15(7), pp.47-59.
- Stoop, P. P. M. and Wiers, V. C. S. (1996) "The complexity of scheduling in practice", *International Journal of Operations & Production Management*, 16(10), pp. 37-53.

- Stump, B. and Badurdeen, F. (2012) "Integrating lean and other strategies for mass customization manufacturing: a case study", *Journal of Intelligent Manufacturing*, 23(1), pp. 109-124.
- Sugimori, Y. K., Kusunoki, K, Cho, F. and Uchikawa, S. (1977) "Toyota Production System and Kanban System: materialization of just-in-time and respect-for-human system", *International Journal of Production Research*, 15(6), pp. 553-564.
- Suri, R., (1998) *Quick Response Manufacturing*. Portland, OR: Productivity Press.
- Svensson, G. (2001) "Just-in-time: the reincarnation of past theory and practice", *Management Decision*, 39(10), pp. 866-879.
- Swamidass, P. M. (2000) *Encyclopedia of production and manufacturing management*. Boston, MA: Kluwer Academic Publishers.
- Swamidass, P. M. (2007) "The effect of TPS on US manufacturing during 1981-1998: inventory increased or decreased as a function of plant performance", *International Journal of Production Research*, 45(16), pp. 3763-3778.
- Swanson, C. A. and Lankford, W. M. (1998) "Just-in-time manufacturing", *Business Process Management Journal*, 4(4), pp. 333-341.
- Sycara, K. P. (1998) "Multiagent systems", *AI Magazine*, 19(2), pp. 79-92.
- T' kindt, V. and Billaut, J.-C. (2006) *Multicriteria scheduling: Theory, models and algorithms*. 2<sup>nd</sup> ed. Berlin-Heidelberg: Springer-Verlag.
- Taha, H. A. (2011) *Operations research: An introduction*. 9<sup>th</sup> ed. Upper Saddle River, NJ: Pearson Education, Inc.
- Tanenbaum, A. S. and Van Steen, M. (2007) *Distributed systems: Principles and paradigms*. 2<sup>nd</sup> ed. Upper Saddle River, NJ: Pearson Education, Inc.
- Tang, H. P. and Wong, T. N. (2005) "Reactive multi-agent system for assembly cell control", *Robotics and Computer-Integrated Manufacturing*, 21 (2), pp. 87–98.
- Tang, O. and Grubbström, R. W. (2002) "Planning and replanning the master production schedule under demand uncertainty", *International Journal of Production Economics*, 78 (3), pp. 323-334.

- Tenhiälä, A. (2011) "Contingency theory of capacity planning: The link between process types and planning methods", *Journal of Operations Management*, 29 (1), pp. 65-77.
- Teo, C.-C., Bhatnagar, R. and Graves, S. C. (2011) "Setting planned lead times for a make-to order production system with master schedule smoothing", *IIE Transactions*, 43(6), pp. 399-414.
- Thesen, A. (1999) "Some simple, but efficient, push and pull heuristics for production sequencing for certain flexible manufacturing systems", *International Journal of Production Research*, 37(7), pp. 1525-1539.
- Timm, T. and Blecken, A (2011) "A method for the hierarchical planning of the structure, dimension and material requirements of manufacturing systems", *International Journal of Production Research*, 49(12), pp. 3431-3453.
- Toksari, M. D., İşleyen, S. K., Güner, E. and Baykoç, Ö. F. (2010) "Assembly line balancing problem with deterioration tasks and learning effect", *Expert Systems with Applications*, 37(2), pp. 1223-1228.
- Topaloglu, S., Salum, L. and Supciller, A. A. (2012) "Rule-based modeling and constraint programming based solution of the assembly line balancing problem", *Expert Systems with Applications*, 39(3), pp. 3484-3493.
- Towill, D. R. (2007) "Exploiting the DNA of the Toyota Production System", *International Journal of Production Research*, 45(16), pp. 3619-3637.
- Turban, E., Sharda, R. and Delen, D. (2011) *Decision support and business intelligence systems*. 9<sup>th</sup> ed. Upper Saddle River, NJ: Pearson Education, Inc.
- Turesky, E. F. and Connell, P. (2010) "Off the rails: understanding the derailment of a lean manufacturing initiative", *Organization Management Journal*, 7, pp. 110-132.
- Turner, R., Lane, J. A., Ingold, D., Madachy, R. and Anderson, D. (2012) An event-driven, value-based, pull systems engineering scheduling approach. In: *Proceedings of IEEE International Systems Conference (SysCon)*, March 19-22, 2012, Vancouver, Canada, pp. 1-7.
- Uhrmacher, A. and Weyns, D. eds. (2009) *Multi-agent systems: Simulation and applications*. Boca Raton, FL: CRC Press, Taylor & Francis Group LLC.
- Unland, R., Klusch, M. and Calisti, M. eds. (2005) *Software agent-based applications, platforms and development kits*. Berlin, Birkhauser: Springer-Verlag.

- Upton, D. (1998) "Just-in-time and performance measurement systems", *International Journal of Operations & Production Management*, 18(11), pp. 1101-1110.
- Van der Krogt, R. Geraghty, J., Salman, M. R. and Little, J. (2010) "On supporting Lean methodologies using constraint-based scheduling", *Journal of Scheduling*, 13(1), pp. 301-314.
- Van der Zee, D. J. (2006) "Modeling decision making and control in manufacturing simulation", *International Journal of Production Economics*, 100 (1), pp. 155–167.
- Vinod, V. and Sridharan, R. (2011) "Simulation modeling and analysis of due-date assignment methods and scheduling decision rules in a dynamic job shop production system", *International Journal of Production Economics*, 129(1), pp. 127-146.
- Vinodh, S. and Chintha, S. K. (2010) "Leanness assessment using multi-grade fuzzy approach" *International Journal of Production Research*, 49(2), pp. 431-445.
- Vollmann, T. E., Berry, W. L. and Whybark, D. C. (1997) *Manufacturing planning and control systems*. 4<sup>th</sup> ed. New York, NY : Irwin/McGraw-Hill.
- Vonderembse, M. A. and White, G. P. (2004) *Operations management : concepts, methods, and strategies*. Hoboken, NJ: John Wiley & Sons, Inc.
- Voss, C. A. (1984) "Japanese manufacturing management practices in the UK", *International Journal of Operations & Production Management*, 4(2), pp. 31-38.
- Voss, G. A. and Robinson, S. J. (1987) "Application of Just-in-Time manufacturing techniques in the United Kingdom", *International Journal of Operations & Production Management*, 7(4), pp. 46-52.
- Vrba, P. (2003) *JAVA-based agent platform evaluation*. In: Mařík, V., McFarlane, D. and Valckenaers, P. eds. (2003) *HoloMAS 2003, LNAI 2744*. Berlin-Heidelberg: Springer-Verlag. pp. 47-58.
- Wacker, J. G. and Sheu, C. (2006) "Effectiveness of manufacturing planning and control systems on manufacturing competitiveness: evidence from global manufacturing data", *International Journal of Production Research*, 44(5), pp. 1015-1036.
- Wafa, M. A. and Yasin, M. M. (1998) "A conceptual framework for effective implementation of JIT: An empirical investigation", *International Journal of Operations & Production Management*, 18(11), pp. 1111-1124.

- Wainer, G. A. (2009) *Discrete-event modelling and simulation: A practitioner's approach*. Boca Raton. FL: Taylor & Francis Group, LLC.
- Walker, S. S., Brennan, R. W. and Norrie, D. H. (2005) "Holon job-shop scheduling using a multiagent system", *IEEE Intelligent Systems*, 20(1), pp. 50-57.
- Waller, D. L. (2003) *Operations management - a supply chain approach*. 2<sup>nd</sup> ed. London: Thomson Learning.
- Wallis, P., Rönquist, R., Jarvis, D. and Lucas, A. (2002) The automated wingman - Using JACK intelligent agents for Unmanned Autonomous Vehicles. In: *Proceedings of the IEEE Aerospace Conference*, 9-16 March 2002, Carlton, Victoria, Australia, pp. 2615-2622.
- Walters, C. D. J. (2003) *Inventory control and management*. Chichester: John Wiley & Sons, Ltd.
- Wan, H.-D. and Chen, F. F. (2008) "A leanness measure of manufacturing systems for quantifying impacts of lean initiatives", *International Journal of Production Research*, 46(23), pp. 6567-6584.
- Wang, C., Ghenniwa, H. and Shen, S. (2008) "Real time distributed shop floor scheduling using an agent-based service-oriented architecture", *International Journal of Production Research*, 46(9), pp. 2433-2452.
- Wang, L., Ng, A. H. C. Deb, K. eds. (2011) *Multi-objective evolutionary optimisation for product design and manufacturing*. London: Springer-Verlag Ltd.
- Wang, L.-C. and Lin, S.-K. (2009) "A multi-agent based agile manufacturing planning and control system", *Computers & Industrial Engineering*, 57 (2), pp. 620–640.
- Wang, Y.-C. and Usher, J. M. (2007) "A reinforcement learning approach for developing routing policies in multi-agent production scheduling", *International Journal of Advanced Manufacturing Technology*, 33(3-4), pp. 323-333.
- Wang, Z. and Liu, Y. (2006) A multi-agent agile scheduling system for job-shop problem. In: *Proceedings of the 6<sup>th</sup> International Conference on Intelligent Systems Design and Applications (ISDA)*, October 16-18, Jinan, China, pp. 1837-1841.
- Wee, H. M. and Wu, S. (2009) "Lean supply chain and its effect on product cost and quality: a case study on Ford Motor Company", *Supply Chain Management: An International Journal*, 14(5), pp. 335-341.



- Weng, M. X., Wu, Z., Qi, G. and Zheng, L. (2008) "Multi-agent-based workload control for make-to-order manufacturing", *International Journal of Production Research*, 46(8), pp. 2197-2213.
- Westbrook, R. (1988) "Time to forget Just-in-Time? – Observations on a visit to Japan", *International Journal of Operations & Production Management*, 8(4), pp. 5-21.
- White, R. E. and Prybutok, V. (2001) "Relationship between JIT practices and type of production system", *Omega International Journal of Management Science*, 29(2), pp. 113- 124.
- Wiendahl, H. -H., Von Cieminski, G. and Wiendahl, H.-P. (2005) 'Stumbling blocks of PPC: Towards the holistic configuration of PPC systems', *Production Planning & Control*, 16(7), pp. 634-651.
- Wild, R. (1994) *Production and operations management*. 4<sup>th</sup> ed. London: Cassell Educational Ltd.
- Wild, R. (2002) *Operations management*. 6<sup>th</sup> ed. London : Continuum.
- Wildemann, H. (1988) "Just-in-time production in West Germany", *International Journal of Production Research*, 26(3), pp. 521-538.
- Winikoff, M. (2005) *JACK™ intelligent agents: An industrial strength platform*. In: Bordini, R. H., Dastani, M., Dix, J. and Seghrouchni, A. E. F. eds. (2009) *Multi-agent programming: Languages, tools and applications*. York, NY: Springer Science+Business Media, LLC. pp 175-193.
- Womack, J. P., and Jones, D. T. (1994) "From lean production to the lean enterprise", *Harvard Business Review*, 72(2), pp. 93-104.
- Womack, J. P., and Jones, D. T. (2003) *Lean thinking: Banish waste and create wealth for your corporation*. 2<sup>nd</sup> ed. New York, NY: Simon and Schuster.
- Womack, J. P., Jones, D. T. and Roos, D. (1990) *The machine that changed the world*. New York, NY: Rawson Associates.
- Wong, T.N., Leung, C.W., Mak, K.L. and Fung, R.Y.K. (2006) "Dynamic shopfloor scheduling in multi-agent manufacturing systems", *Expert Systems with Applications*, 31(3), pp. 486–494.
- Wooldridge, M. (2002) *An introduction to multi agent systems*. Chichester: John Wiley & Sons, Ltd.

- Wooldridge, M. and Dunne, P. E. (2006) "On the computational complexity of coalitional resource games", *Artificial Intelligence* 170(10), pp. 835-871.
- Wooldridge, M. and Jennings, N. R. (1995) "Intelligent agents: Theory and practice", *The Knowledge Engineering Review*, 10(2), pp.115-152.
- Worley, J. M. and Doolen, T. L. (2006) "The role of communication and management support in a lean manufacturing implementation", *Management Decision*, 44(2), p. 228-245.
- Wu, Z. and Weng, M. X. (2005) "Multiagent Scheduling Method With Earliness and Tardiness Objectives in Flexible Job Shops", *IEEE Transactions on Systems, Man, and Cybernetics - Part B: Cybernetics*, 35(2), pp. 293-301.
- Xing, L.-N., Chen, Y.-W., Wang, P., Zhao, Q.-S. and Xiong, J. (2010) "A Knowledge-Based Ant Colony Optimization for Flexible Job Shop Scheduling Problems", *Applied Soft Computing*, 10 (3), pp. 888-896.
- Yang, K. K. (2000) "Managing a flow line with single-kanban, dual-kanban or CONWIP", *Production and Operations Management*, 9(4), pp. 349-366.
- Yavuz, M. and Akçali, E. (2007) "Production smoothing in just-in-time manufacturing systems: a review of the models and solution approaches", *International Journal of Production Research*, 45(16), pp. 3579 -3597.
- Yusuf, Y. Y., Sarhadi, M. and Gunasekaran, A. (1999) "Agile manufacturing: The drivers, concepts and attributes", *International Journal of Production Economics*, 62(1-2), pp. 33-43.
- Yusuf, Y.Y., Adeley, E.O. (2002) "A comparative study of lean and agile manufacturing with related survey of current practices in the UK", *International Journal of Production Research* 40(17), pp. 829-839.
- Zattar, I. C., Ferreira, J. C. E., Rodrigues, J. G. G. G. and De Sousa, C. H. B. (2010) "A multi-agent system for the integration of process planning and scheduling using operation-based time-extended negotiation protocols", *International Journal of Computer Integrated Manufacturing*, 23(5), pp. 441-452.
- Zhang, D. Z., Anosike, A. and Lim, M. K. (2007) "Dynamically Integrated Manufacturing Systems (DIMS) – A Multiagent Approach", *IEEE Transactions on Systems, Man, and Cybernetics – Part A: Systems And Humans*, 37(5), pp. 824-850.

Zhang, G., Gao, G. and Shi, Y. (2011) "An effective genetic algorithm for the flexible job-shop scheduling problem", *Expert Systems with Applications*, 38(4), pp. 3563-3573.

Zhu, Z. and Meredith, P. H. (1995) "Defining critical elements in JIT implementation: a survey", *Industrial Management and Data Systems*, 95(8), pp. 21-28.

Zipkin, P. (2000) *Foundations of inventory management*, Management & Organization Series, Boston, MA: McGraw-Hill.

## **APPENDIX A**

### **Most common dispatching rules in production scheduling**

- Cost over time (COVERT). Priority assigned to the job with the largest priority index (cost penalty imposed depending on the job's slack and uncompleted operations).
- Critical ratio (CR). Priority assigned to the job with the least critical ratio (time to due date divided by total remaining production time).
- Earliest due date (EDD). Priority assigned to the job which is due for delivery first.
- First arrival at shop first served (FASFS). Priority assigned to first job arriving at the shop.
- Alternative names: First in first served (FIFS).
- First come first served (FCFS). Priority assigned to first job arriving at a specific workstation (machine).
- Alternative names: Earliest release date (ERD).
- Fewest operations remaining (FOR). Priority assigned to the job with the least number of operations remaining in its processing sequence.
- Least changeover cost (LCC). Priority assigned to the job with the least changeover cost (at a specific workstation).
- Last come first served (LCFS). Priority assigned to the last job arriving at a specific workstation (machine).
- Least flexible job (LFJ). Priority assigned to the job that can be processed on the smallest number of non-identical parallel machines.
- Longest processing time (LPT). Priority assigned to the job with the maximum processing time (at a specific workstation).
- Alternative names: Longest imminent operation (LIO)
- Least total work (LTWK). Priority assigned to the job with the smallest total processing time (before any processing).
- Alternative names: Total work (TWORK)
- Least work remaining (LWKR). Priority assigned to the job with the smallest total processing time for all remaining operations.
- Most operations remaining (MOPNR). Priority assigned to the job with the most number of operations remaining in its processing sequence.
- Minimum slack (MS). Priority assigned to the job with the least slack (time to due date minus total remaining production time).
- Alternative names: Least Slack (LS).
- Most work remaining (MWKR). Priority assigned to the job with the largest total processing time for all remaining operations.
- Operation due date (ODD).

- RUSH. Jobs with the highest customer priority are processed first.
- Slack per remaining operation (S/RO). Priority assigned to the job with the lowest slack divided by the number of remaining operations.
- RANDOM. Priority assigned to any job selected at random.
- Alternative names: Service in random order (SIRO)
- Shortest processing time (SPT). Priority assigned to the job with the minimum processing time (at a specific workstation).
- Alternative names: Shortest imminent operation (SIO).
- Scheduled start date (SSD). Priority assigned to the job with the earliest start date computed using reverse scheduling by deducting from the due date the total work content increased by a contingency allowance.
- Shortest setup time (SST). Priority assigned to the job with the minimum setup time (at a specific workstation).
- Work in the next queue (WINQ). Priority assigned to the job whose next machine has the shortest queue.
- Alternative names: Anticipated work in the next queue (AWINQ) or Shortest queue at the next operation (SQNO).
- Weighted shortest processing time (WSPT). Priority assigned to the job with the highest ratio of weight over processing time.

**APPENDIX B**  
**Complete listing of external events in BASS**

Table B.1 Index of external events in BASS

Agent	Sends event (to Agent)	Handles event (sent from Agent)
Customer	CancelJob (SystemManager) InitiateJob (SystemManager)	
Dispatcher	ProcessTask (Machine) RequestInfoRequest (WorkstationOutputBuffer) TaskAssigned (WorkstationInputBuffer) TaskInfoRequest (WorkstationInputBuffer) <sup>1</sup>	AssignTask (WorkstationSupervisor) CancelTaskAssignmentRequest (WorkstationSupervisor ) TaskArrived (WorkstationInputBuffer) TaskInfoResponse (WorkstationInputBuffer) TaskRemoved (WorkstationInputBuffer)
FailureManager	DevelopFault (Machine)	
FGI	AgentFinished (Global message) InventoryResponse (Reply message to JobManager)	AddInventory (JobManager) InventoryRequest (JobManager) SystemFinished (SystemManager) RequestInfoRequest (Dispatcher)
JobManager	AddInventory (WorkstationOutputBuffer or FGI depending on control mode) AddTaskToBuffer (WorkstationInputBuffer) InventoryRequest (WorkstationOutputBuffer or FGI depending on control mode) JobFinished (SystemManager and PerformanceMonitor)	InventoryResponse (WorkstationOutputBuffer or FGI depending on control mode) JobCancelled (SystemManager) StartJob (SystemManager) SystemFinished (SystemManager) TaskCompleted (Machine) TaskFailed (Machine) TaskTerminated (Machine, WorkstationInputBuffer)
Machine	AgentFinished (Global message) JobCancelled (Confirmation message sent back to the WorkstationSupervisor) MachineFinished (PerformanceMonitor) MachineStatusChanged (WorkstationSupervisor) TaskCompleted (JobManager) TaskFailed (JobManager) TaskTerminated (JobManager)	DevelopFault (Failure Manager) JobCancelled (Workstation Supervisor) ProcessTask (Dispatcher) SystemFinished (WorkstationSupervisor)



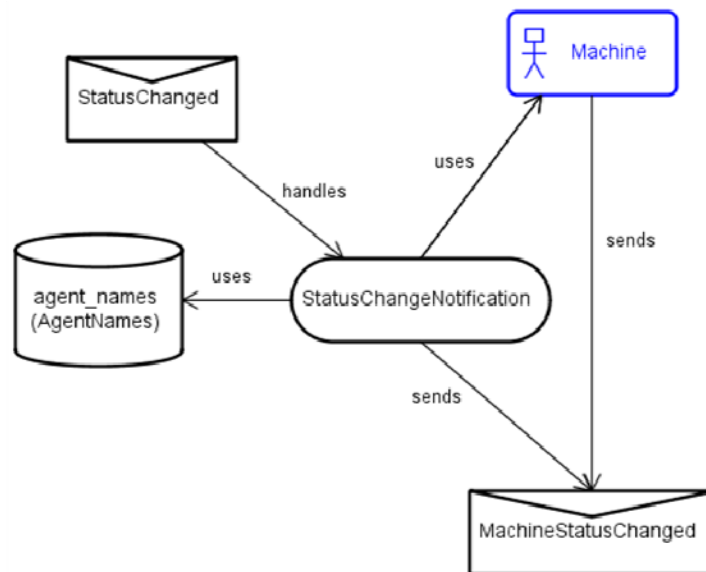
<b>Agent</b>	<b>Sends event (to Agent)</b>	<b>Handles event (sent from Agent)</b>
<b>PerformanceMonitor</b>		JobFinished (JobManager) MachineFinished (Machine) SystemFinished (SystemManager)
<b>SystemManager</b>	JobCancelled (JobManager and WorkstationSupervisor) StartJob (JobManager) SystemFinished (JobManager, WorkstationSupervisor, FGI, PerformanceMonitor)	CancelJob (Customer) InitiateJob (Customer) JobFinished (JobManager)
<b>WorkstationInputBuffer</b>	AgentFinished (Global message) JobCancelled (Confirmation message sent back to the WorkstationSupervisor) TaskArrived (Dispatcher) TaskInfoResponse (Dispatcher) TaskRemoved (Dispatcher) TaskTerminated (JobManager)	AddTaskToBuffer (JobManager) JobCancelled (WorkstationSupervisor) SystemFinished (WorkstationSupervisor) TaskAssigned (Dispatcher) TaskInfoRequest (Dispatcher)
<b>WorkstationSupervisor</b>	AgentFinished (Global message) AssignTask (Dispatcher) CancelTaskAssignmentRequestn(Dispatcher) JobCancelled (WorkstationInputBuffer and Machine) SystemFinished (WorkstationInputBuffer and Machine)	JobCancelled (SystemManager) MachineStatusChanged (Machine) SystemFinished (SystemManager)
<b>WorkstationOutputBuffer</b>	InventoryResponse (JobManager) RequestInfoResponse (Reply message sent to the Dispatcher)	AddInventory (JobManager) InventoryRequest (JobManager) RequestInfoRequest (Dispatcher)

<sup>1</sup> Relevant when the WINQ rule is selected

<sup>2</sup> Relevant when the FCFS/LATE rule is selected

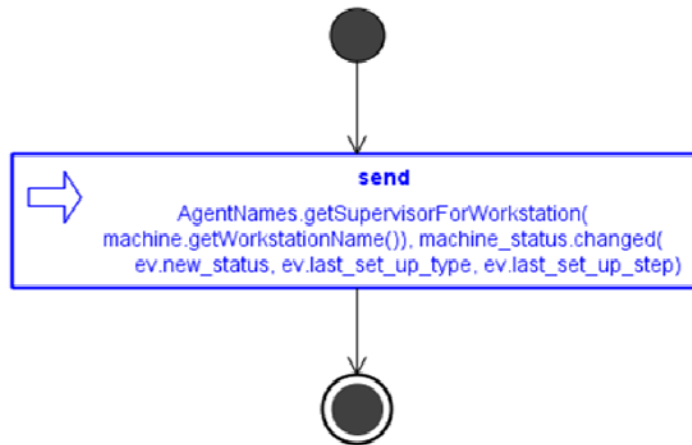
**APPENDIX C**  
**JACK™ Design of Machine Agent (MA)**

This section continues the overview of the MA discussed briefly in Section 5.5.2. The plans associated with the two capabilities of the MA are examined in detail. The StatusMonitoring capability uses four plans. The StatusChangeNotification is responsible for providing notification to the workstation's supervisor when the machine's status changes. It is invoked by the StatusChanged event which is posted internally to indicate that the machine's status has changed. The plan's overall interface is presented in Figure C.1.

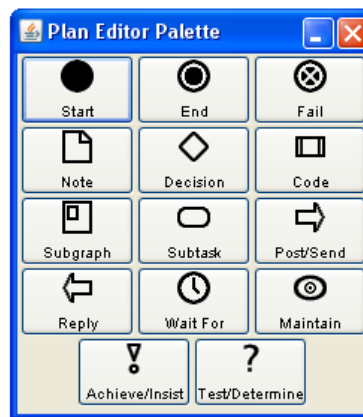


**Figure C.1 StatusChangeNotification plan interface**

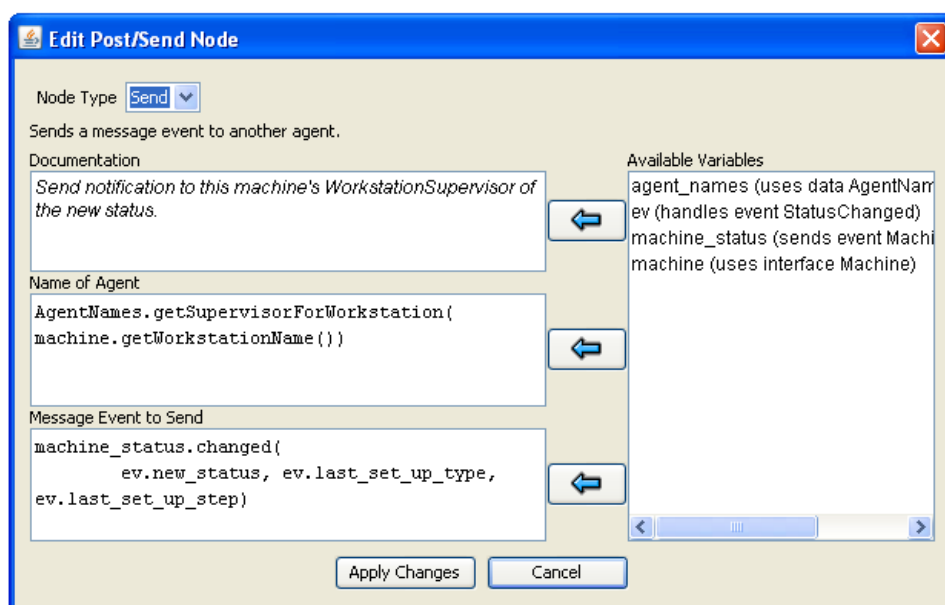
Once the plan is executed, a MachineStatusChanged message is sent to the WSA notifying it of the updated machine's status. The plan identifies the relevant WSA by accessing the agent\_names beliefset. The graphical illustration of the plan's reasoning method is depicted in Figure C.2. By referring to Figure C.3 which presents the graphical plan editor's palette, it can be seen that the plan's reasoning method contains merely a notification action. As shown in the plan's documentation presented in Figure C.4, the MachineStatusChanged message sent to the WSA also contains information about the last set-up operation performed by the machine.



**Figure C.2 Graphical illustration of reasoning method of the StatusChangeNotification plan**



**Figure C.3 Graphical plan editor's palette**



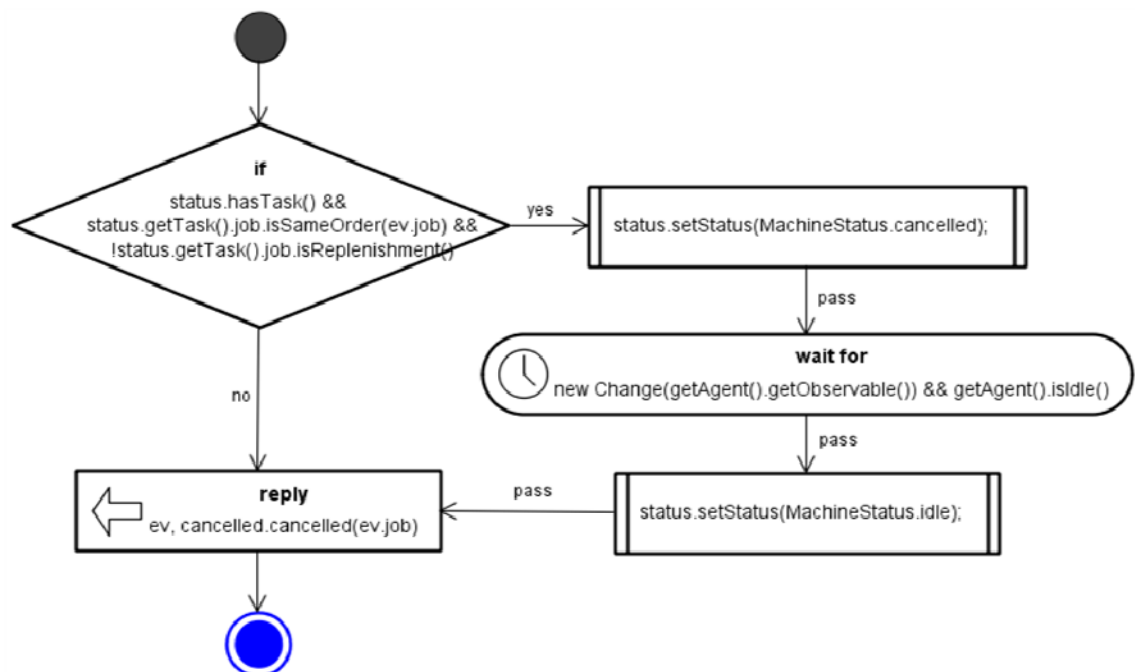
**Figure C.4 StatusChangeNotification plan documentation**

The JobCancellation plan changes the machine's status if the latter is currently processing a task belonging to a cancelled job. The plan is executed following the receipt of a JobCancelled event sent by the WSA and results in the MA updating its status beliefset as shown in Figure C.5.



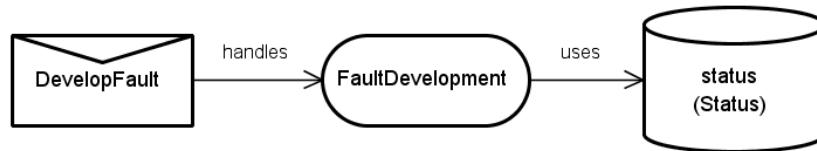
**Figure C.5 JobCancellation plan interface**

The plan's reasoning method presented in Figure C.6, includes a decision node which considers whether the task being processed belongs to the cancelled job. If this is the case, then the MA changes its status initially to "cancelled" and then "idle". If on the contrary the WIP is a replenishment task, the MA continues its processing. In both cases, the MA concludes its action by sending a response to the WSA to confirm that it handled the job cancellation appropriately.



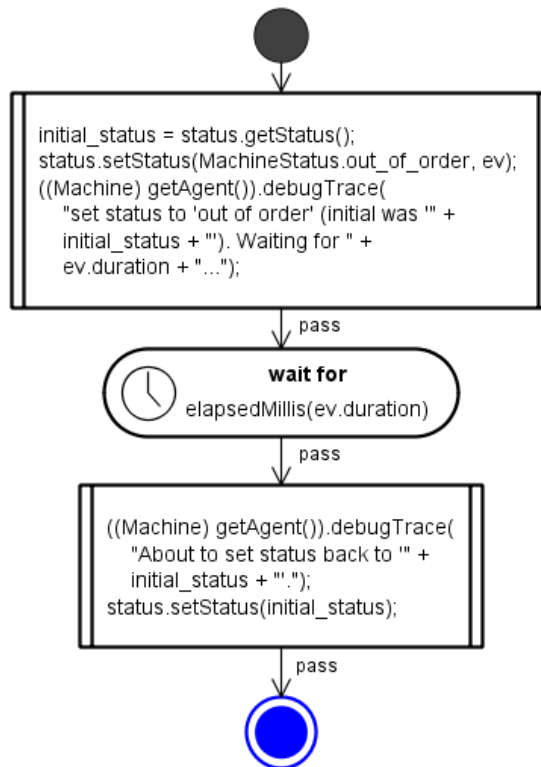
**Figure C.6 Graphical illustration of reasoning method of the JobCancellation plan**

This FaultDevelopment plan adjusts the machine's status when a fault develops. In order to achieve this, the plan has access to the status beliefset. The plan is invoked by DevelopFault events sent by the FMA. Figure C.7 depicts the plan's interface.



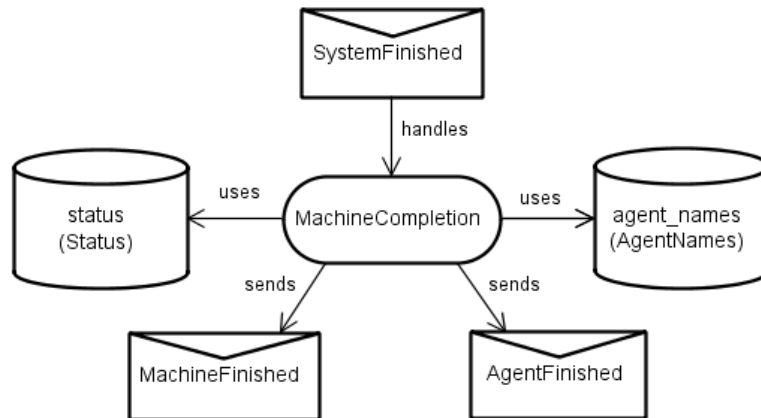
**Figure C.7 FaultDevelopment plan interface**

As shown in Figure C.8, the plan's reasoning method modifies the machine's status to "out-of-order" for the duration of the fault. Once the duration of the fault has elapsed, the plan restores the machine's status to what it was initially, i.e. prior to the development of the fault.



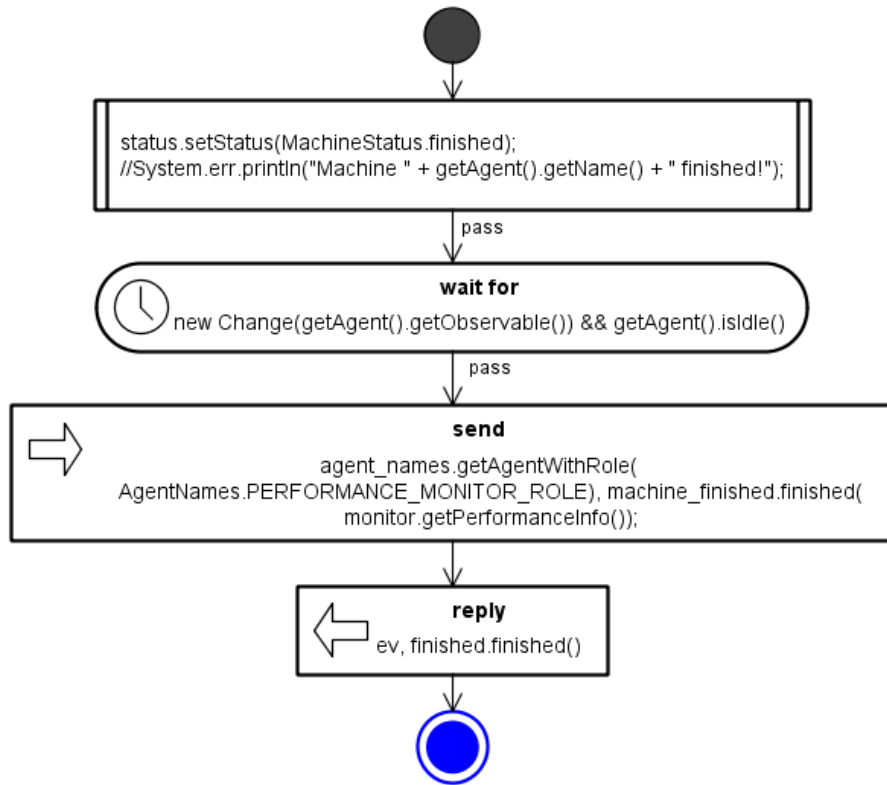
**Figure C.8 Graphical illustration of reasoning method of the FaultDevelopment plan**

MachineCompletion is the final plan associated with the StatusMonitoring capability. This plan reacts to SystemFinished messages sent by the WSA. These messages notify the MA that the system is finishing due to terminating conditions set in the system's configuration. Figure C.9 illustrates the messages handled and sent by the plan, in addition to the beliefsets accessed during the execution of this plan.



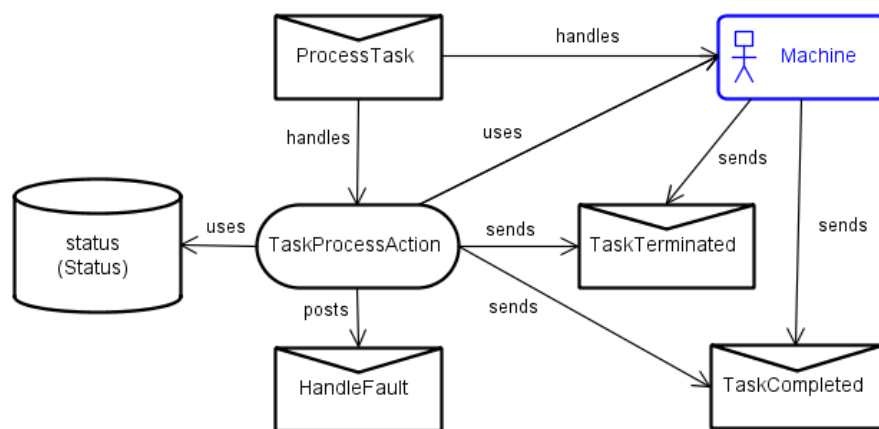
**Figure C.9 MachineCompletion plan interface**

As shown in Figure C.10, the plan's reasoning method sets the machine's status to "finished" and then waits until the machine becomes "idle". It then accesses the agent\_names beliefset and obtains the name of the agent, in this case, the PMA which needs to be notified that the machine has finished its operation and supplied with performance data (concerning the machine's total operation time). A response is sent to the WSA to confirm that the machine has finished its operation.



**Figure C.10 Graphical illustration of reasoning method of the MachineCompletion plan**

TaskProcessAction: This plan processes a given task. It handles ProcessTask events sent by the DA. The beliefsets used by the plan and all the messages which are posted or sent by the plan's reasoning method are shown in Figure C.11.

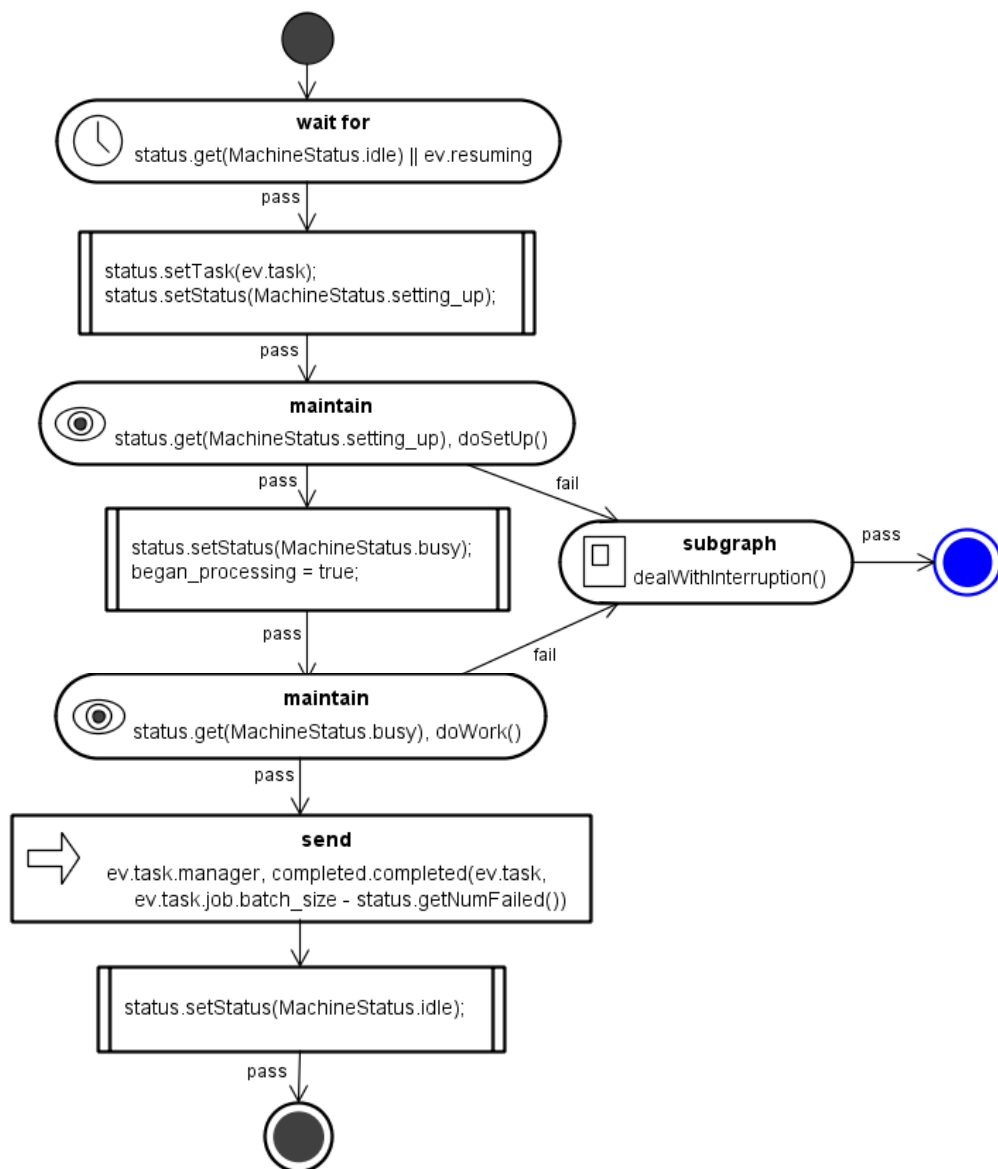


**Figure C.11 TaskProcessAction plan interface**



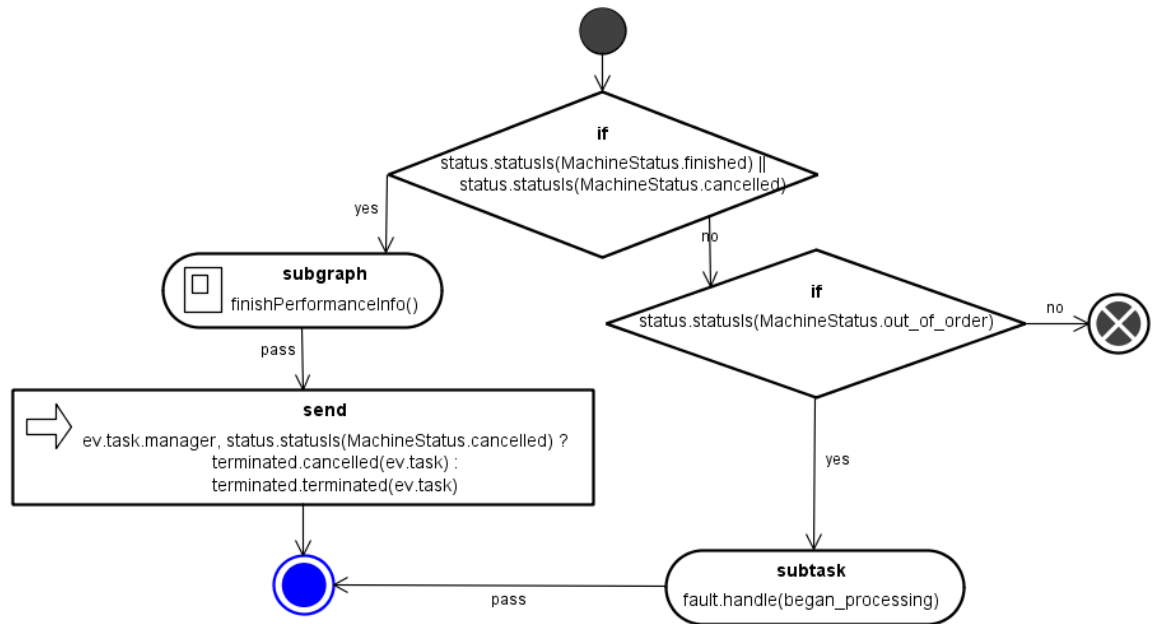
The plan's reasoning method depicted in Figure 12 initially places a delay timer until the machine's status changes to idle. It then denotes the machine status as "setting-up" for a time interval equivalent to the set-up time of the respective task. At the end of this interval, the machine's status changes to "busy" for an interval determined by the processing time of the respective task.

Once the agent completes the processing of the task, it sends a TaskCompleted message to the respective JA. The message confirms the size of the processed batch. This is appropriately adjusted in case of machine faults to show any damage caused to the WIP. Following this, the plan changes the machine's status to "idle".



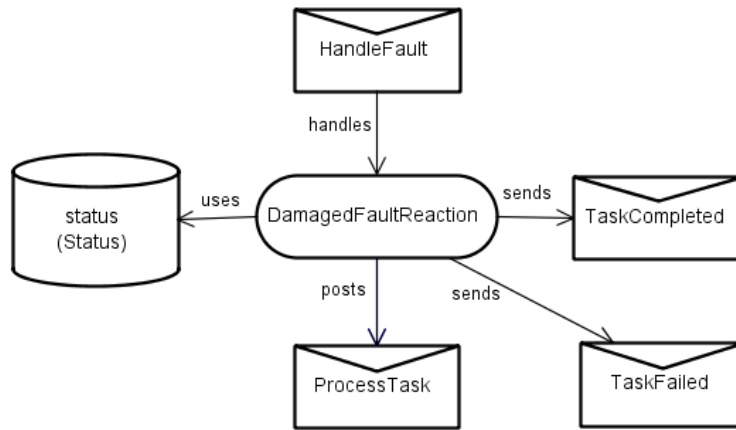
**Figure C.12 Graphical illustration of reasoning method of the TaskProcessAction plan**

The TaskProcessAction plan also deals with machine interruptions. Failure of the plan’s main reasoning method whilst the machine maintains its “setting-up” or “busy” status, that is whilst it is being configured or processing parts, calls the dealWithInterruption sub-plan shown in Figure C.13. The sub-plan’s initial decision node determines the nature of the interruption. If the interruption is caused by the system terminating its operation, performance information is collected and the MA sends a TaskTerminated message to the respective JA. However, if the interruption is caused by a fault, the MA posts an internal HandleFault event to deal with the fault.



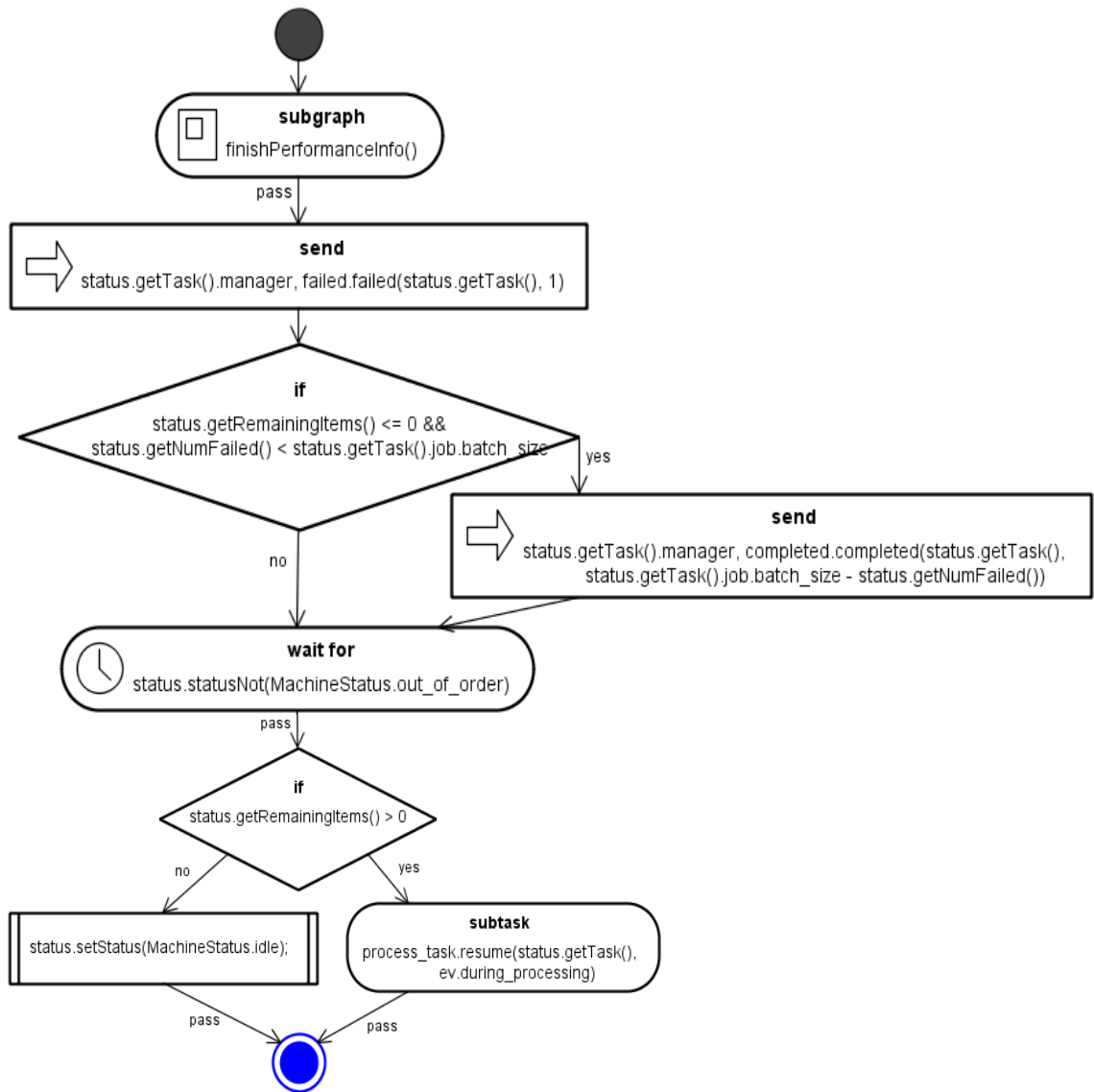
**Figure C.13 Graphical illustration of reasoning method of the dealWithInterruption sub-plan**

HandleFault events can invoke two different plans depending on the system’s configuration, that is whether certain faults are prescribed to cause damage to WIP or not. The DamagedFaultReaction plan processes HandleFault events posted by the MA during the execution of the dealWithInterruption sub-plan discussed above. It reacts to faults that have caused damage to the task being processed by the machine. The plan’s interface is presented in Figure C.14.



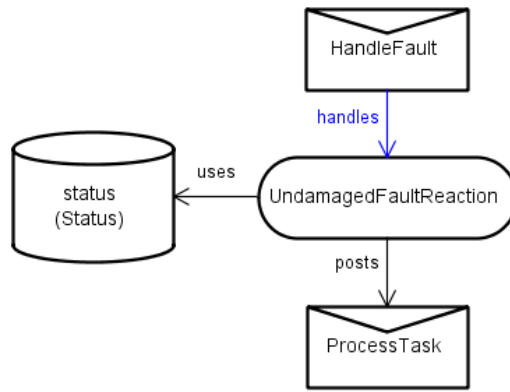
**Figure C.14 DamagedFaultReaction plan interface**

As shown in the graphical representation of the plan’s reasoning method presented in Figure C.15, the plan initially collects performance information about the job in progress and sends a TaskFailed message to the respective JA. If there are no more parts which require processing a TaskCompleted message is sent to the JA containing confirmation of the number of damaged parts. This message triggers an appropriate plan within the JA which seeks to replace the damaged portion of the batch. In case there are parts within the batch that still require processing, a delay timer is set for the duration of the fault. At the end of this interval, an internal ProcessTask event is posted and the MA resumes processing the parts currently on the machine.

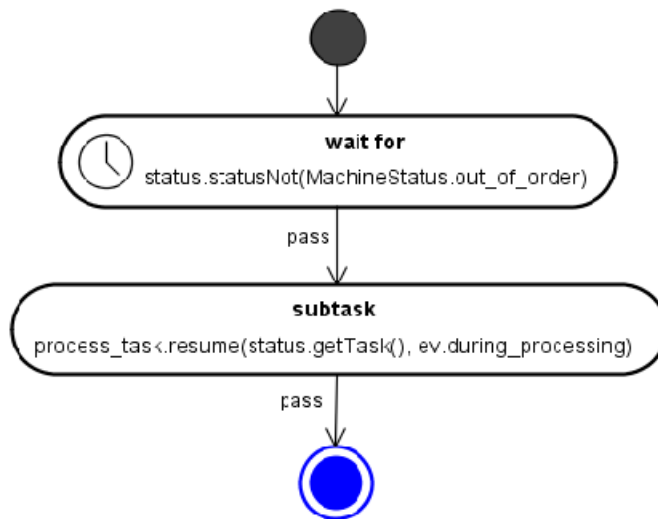


**Figure C.15 Graphical illustration of reasoning method of the DamagedFaultReaction plan**

In case a machine fault is set not to cause damage to the WIP, the HandleFault event calls upon the UndamagedFaultReaction plan. The plan's interface is depicted in Figure C.16. As illustrated in the plan's reasoning method shown in figure C.17, this sets a delay timer for the duration of the fault. A ProcessTask event is posted subsequently. This allows the MA to restore the machine's status to busy and resume processing the WIP.



**Figure C.16 UndamagedFaultReaction plan interface**



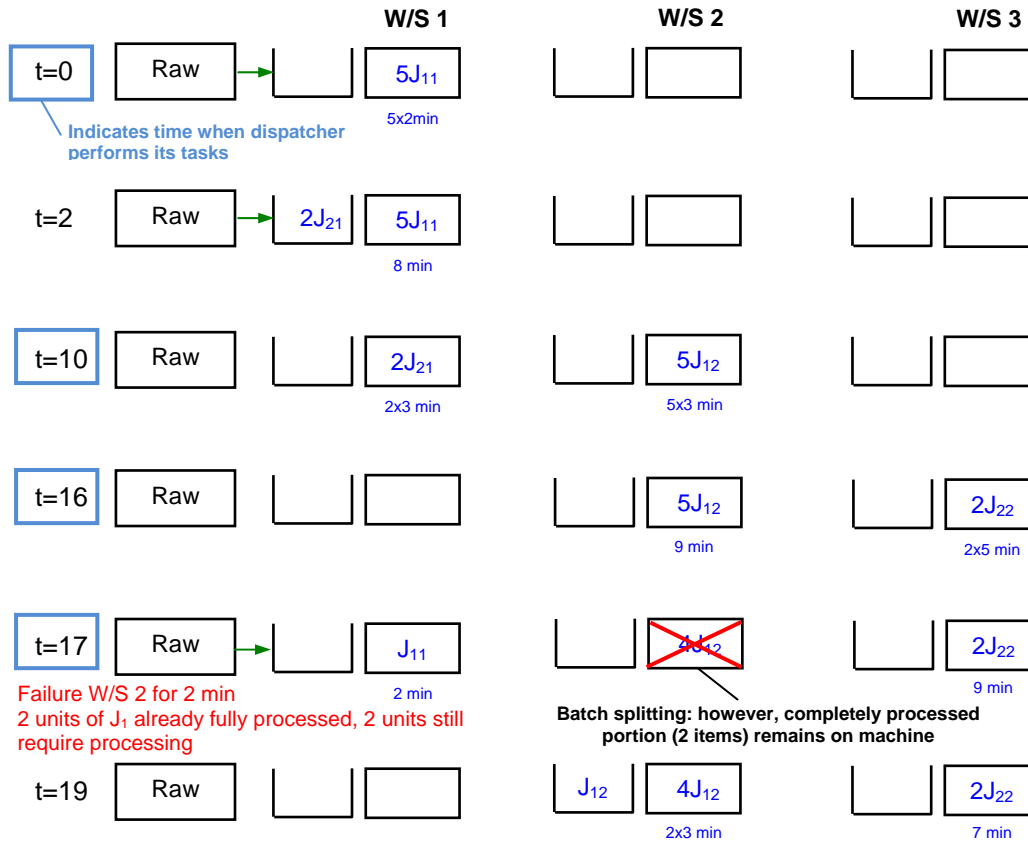
**Figure C.17 Graphical illustration of reasoning method of the UndamagedFaultReaction plan**

**APPENDIX D**  
**BASS verification using experiment BVE1**

**Push (SIPT/Failure/Damage)**

Machine Failure at t=17 min and t=28 min resulting in partial damage of batch in progress

time	Job	Machine (Process time)			TPT	Due Date=arrival time+1xTPT
0	5xJ <sub>1</sub>	J <sub>11</sub> =1(2)	J <sub>12</sub> =2(3)	J <sub>13</sub> =3(3)	8	DD=0+1x8=8
2	2xJ <sub>2</sub>	J <sub>21</sub> =1(3)	J <sub>22</sub> =3(5)	J <sub>23</sub> =2(2)	10	DD=2+1x10=12



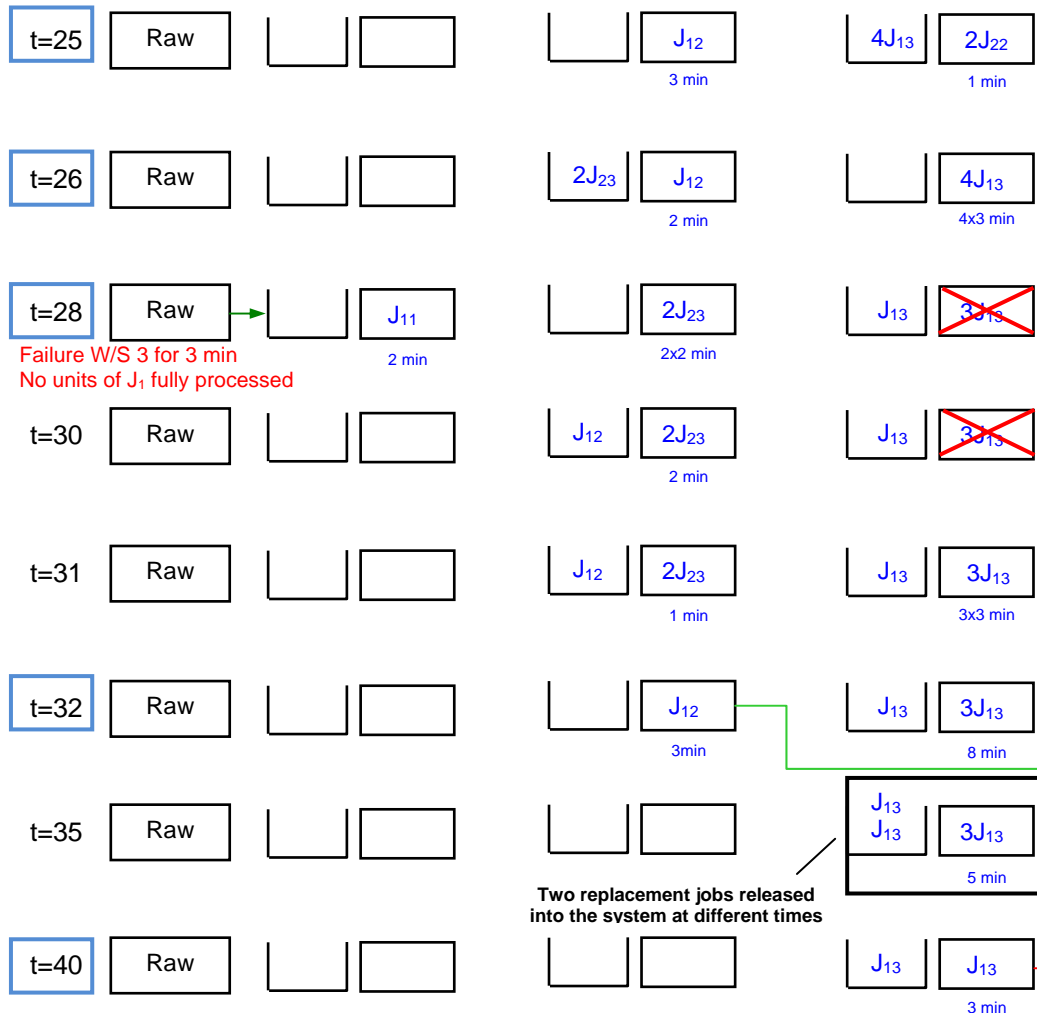
Time in Queue=0x2 min  
 WIP=1x2  
 Utilisation=1x2

Time in Queue=1x8 min  
 WIP=2x8  
 Utilisation=1x8

Time in Queue=0x6 min  
 WIP=2x6  
 Utilisation=2x6

Time in Queue=0x1 min  
 WIP=2x1  
 Utilisation=2x1

Time in Queue=0x2 min  
 WIP=3x2  
 Utilisation=2x2



Failure W/S 3 for 3 min  
No units of J<sub>1</sub> fully processed

Two replacement jobs released into the system at different times

Time in Queue=1x6 min  
WIP=3x6  
Utilisation=2x6

Time in Queue=1x1 min  
WIP=3x1  
Utilisation=2x1

Time in Queue=1x2 min  
WIP=3x2  
Utilisation=2x2

Time in Queue=1x2 min  
WIP=4x2  
Utilisation=2x2

Time in Queue=2x1 min  
WIP=4x1  
Utilisation=1x1

Time in Queue=2x1 min  
WIP=4x1  
Utilisation=2x1

Time in Queue=1x3 min  
WIP=3x3  
Utilisation=2x3

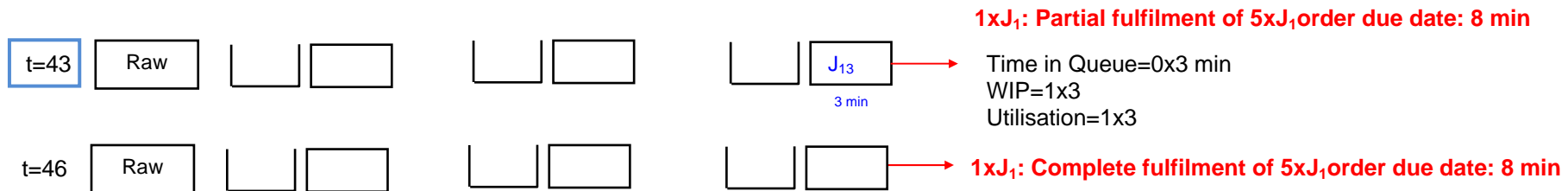
**1xJ<sub>2</sub>: Complete fulfilment of 2xJ<sub>2</sub> order due date: 12 min**

Time in Queue=2x5 min  
WIP=3x5  
Utilisation=1x5

**3xJ<sub>1</sub>: Partial fulfilment of 5xJ<sub>1</sub> order due date: 8 min**

Time in Queue=1x3 min  
WIP=2x3  
Utilisation=1x3





Performance Metric	Experimentation Results
Number of Tardy Jobs	= 2 jobs
Total Absolute Deviation of Earliness/Tardiness	=  (46-8)+(32-12) =58 min
Average Flow Time (Completion-Ready)/No. of Jobs	= [(46-0)+(32-2)]/2=38 min
Throughput (Completion-Start)	= (46-0)+(32-10)=68 min
Makespan (Completion Last-Start First)	= 46-0=46 min
Time in Queue	= 0+8+0+0+0+6+1+2+2+2+2+3+10+3+0=39 min
WIP	= [(1x2)+(2x8)+(2x6)+(2x1)+(3x2)+(3x6)+(3x1)+(3x2)+(4x2)+(4x1)+(4x1)+(3x3)+(3x5)+(2x3)+(1x3)]/46=114/46=2.48 jobs
Machine Utilisation	= [(1x2)+(1x8)+(2x6)+(2x1)+(2x2)+(2x6)+(2x1)+(2x2)+(2x2)+(1x1)+(2x1)+(2x3)+(1x5)+(1x3)+(1x3)]/(3x46)=70/138=50.72%
Set-up Time	=0 min
Fill Rate	= 0.00%

**JACK™ output**

Total throughput time (mins): 68.00  
 Average flow time (mins): 38.00  
 Makespan (mins): 46.00  
 Earliness (mins): 0.00  
 Tardiness (mins): 58.00  
 Earliness/Tardiness (mins): 58.00  
 Number of tardy jobs: 2  
 Fill rate: 0.00  
 Average machine utilisation: 50.72  
 WIP level: 2.48  
 Total time in queue (mins): 39.00  
 Total set-up time (mins): 0.00  
 Number of jobs finished at termination: 2

**APPENDIX E**  
**JACK™ Performance tracing log for experiment BVE1**

\*\*\*BVE1 Debug File\*\*\*\*

```
----- Beginning of run -----
No FGI manager defined in config, not creating one. 0: w1m1:
Changing status to :idle.
0: w2m1: Changing status to :idle. 0: w3m1:
Changing status to :idle.
0: Set manager to 'JobManager-1-0min-5270791196649818156' for job Job JobType 1 x 5, arrival 0min 0: Next fault
will be at 17mins for 2minson w2m1.
0: dispo: Attempting to assign task for machine 'w1m1@%portal' on workstation 'w1-supervisor@%portal'. 0: Added
incomplete portion Job-28899428; now [Job-28899428].
0: JobManager-1-0min-5270791196649818156: Added job 1:0min(Job-28899428).
0: JobManager-1-0min-5270791196649818156: About to initiate task 1:0min(Job-28899428):1. 0: dispo:
Adding task for w1-in-buffer@%portal.
0: dispo: Attempting to assign task for machine 'w3m1@%portal' on workstation 'w3-supervisor@%portal'. 0: dispo:
Attempting to assign task for machine 'w2m1@%portal' on workstation 'w2-supervisor@%portal'. 0: dispo: Got task and
semaphore for workstation 'w1'.
0: dispo: Got reply from buffer for workstation 'w1'. 0: dispo:
About to select task (from 1 available).
0: SIPTSelector: task 1:0min(Job-28899428):1 has imminent processing time of 600000 0:
SIPTSelector: Shortest is 1:0min(Job-28899428):1, with time 600000
0: dispo: Got selected task: '1:0min(Job-28899428):1'. 0: dispo:
Selected task for workstation 'w1'.
0: dispo: Assigned task for machine 'w1m1@%portal' on workstation 'w1-supervisor@%portal'. 0: w1m1:
Changing status to :setting_up.
0: w1m1: Setting up for task 1:0min(Job-28899428):1. 0: w1m1: Set-
up time is 0
0: w1m1: Finished setting up for task 1:0min(Job-28899428):1. 0: w1m1:
Changing status to :busy.
0: w1m1: Starting task 1:0min(Job-28899428):1.
2: Set manager to 'JobManager-2-2mins-3551044553175055070' for job Job JobType 2 x 2, arrival 2mins 2: Added
incomplete portion Job-12285785; now [Job-12285785].
2: JobManager-2-2mins-3551044553175055070: Added job 2:2mins(Job-12285785).
2: JobManager-2-2mins-3551044553175055070: About to initiate task 2:2mins(Job-12285785):1. 2: dispo:
Adding task for w1-in-buffer@%portal.
10: w1m1: Finished task 1:0min(Job-28899428):1. 10: w1m1:
Changing status to :idle.
10: dispo: Attempting to assign task for machine 'w1m1@%portal' on workstation 'w1-supervisor@%portal'. 10: dispo: Got
task and semaphore for workstation 'w1'.
10: Logging task 1 as complete for portion 1:0min(Job-28899428); next_step will be 2. 10: JobManager-
1-0min-5270791196649818156: Completed task 1:0min(Job-28899428):1.
10: JobManager-1-0min-5270791196649818156: Completed task 1:0min(Job-28899428):1; about to start task 1:0min(Job-
28899428):2.
```

```

10: dispo: Adding task for w2-in-buffer@%portal.
10: dispo: Got task and semaphore for workstation 'w2'. 10: dispo:
Got reply from buffer for workstation 'w2'. 10: dispo: About to
select task (from 1 available).
10: SIPTSelector: task 1:0min(Job-28899428):2 has imminent processing time of 900000 10:
SIPTSelector: Shortest is 1:0min(Job-28899428):2, with time 900000
10: dispo: Got selected task: '1:0min(Job-28899428):2'.
10: dispo: Selected task for workstation 'w2'.
10: dispo: Assigned task for machine 'w2m1@%portal' on workstation 'w2-supervisor@%portal'. 10: dispo:
Got reply from buffer for workstation 'w1'.
10: dispo: About to select task (from 1 available).
10: SIPTSelector: task 2:2mins(Job-12285785):1 has imminent processing time of 360000 10:
SIPTSelector: Shortest is 2:2mins(Job-12285785):1, with time 360000
10: dispo: Got selected task: '2:2mins(Job-12285785):1'. 10: dispo:
Selected task for workstation 'w1'.
10: dispo: Assigned task for machine 'w1m1@%portal' on workstation 'w1-supervisor@%portal'. 10: w1m1:
Changing status to :setting_up.
10: w1m1: Setting up for task 2:2mins(Job-12285785):1. 10: w1m1:
Set-up time is 0
10: w1m1: Finished setting up for task 2:2mins(Job-12285785):1. 10: w1m1:
Changing status to :busy.
10: w1m1: Starting task 2:2mins(Job-12285785):1. 10: w2m1:
Changing status to :setting_up.
10: w2m1: Setting up for task 1:0min(Job-28899428):2. 10: w2m1:
Set-up time is 0
10: w2m1: Finished setting up for task 1:0min(Job-28899428):2. 10: w2m1:
Changing status to :busy.
10: w2m1: Starting task 1:0min(Job-28899428):2. 16: w1m1:
Finished task 2:2mins(Job-12285785):1. 16: w1m1: Changing
status to :idle.
16: dispo: Attempting to assign task for machine 'w1m1@%portal' on workstation 'w1-supervisor@%portal'. 16:
Logging task 1 as complete for portion 2:2mins(Job-12285785); next_step will be 2.
16: JobManager-2-2mins-3551044553175055070: Completed task 2:2mins(Job-12285785):1.
16: JobManager-2-2mins-3551044553175055070: Completed task 2:2mins(Job-12285785):1; about to start task
2:2mins(Job-12285785):2.
16: dispo: Adding task for w3-in-buffer@%portal.
16: dispo: Got task and semaphore for workstation 'w3'. 16: dispo:
Got reply from buffer for workstation 'w3'. 16: dispo: About to
select task (from 1 available).
16: SIPTSelector: task 2:2mins(Job-12285785):2 has imminent processing time of 600000 16:
SIPTSelector: Shortest is 2:2mins(Job-12285785):2, with time 600000
16: dispo: Got selected task: '2:2mins(Job-12285785):2'. 16: dispo:

```

Selected task for workstation 'w3'.  
 16: dispo: Assigned task for machine 'w3m1@%portal' on workstation 'w3-supervisor@%portal'. 16: w3m1:  
 Changing status to :setting\_up.  
 16: w3m1: Setting up for task 2:2mins(Job-12285785):2. 16: w3m1:  
 Set-up time is 0  
 16: w3m1: Finished setting up for task 2:2mins(Job-12285785):2. 16: w3m1:  
 Changing status to :busy.  
 16: w3m1: Starting task 2:2mins(Job-12285785):2. 17: Next  
 fault will be at 28mins for 3minson w3m1. 17: w2m1: Changing  
 status to :out\_of\_order.  
 17: w2m1: set status to 'out of order' (initial was ':busy'). Waiting for 120000... 17: w2m1:  
 Failed for task 1:0min(Job-28899428):2.  
 17: JobManager-1-0min-5270791196649818156: Task 1:0min(Job-28899428):2 partly failed; splitting and trying again.  
 17: JobManager-1-0min-5270791196649818156: About to start replacement task 1:0min(Job-28899428-split-0):1. 17: dispo:  
 Adding task for w1-in-buffer@%portal.  
 17: dispo: Cancelling request to assign task for machine 'w2m1@%portal' on workstation 'w2-supervisor@%portal'.  
 17: dispo: Got task and semaphore for workstation 'w1'. 17: dispo:  
 Got reply from buffer for workstation 'w1'. 17: dispo: About to  
 select task (from 1 available).  
 17: SIPTSelector: task 1:0min(Job-28899428-split-0):1 has imminent processing time of 120000 17:  
 SIPTSelector: Shortest is 1:0min(Job-28899428-split-0):1, with time 120000  
 17: dispo: Got selected task: '1:0min(Job-28899428-split-0):1'. 17: dispo:  
 Selected task for workstation 'w1'.  
 17: dispo: Assigned task for machine 'w1m1@%portal' on workstation 'w1-supervisor@%portal'. 17: w1m1:  
 Changing status to :setting\_up.  
 17: w1m1: Setting up for task 1:0min(Job-28899428-split-0):1. 17: w1m1:  
 Set-up time is 0  
 17: w1m1: Finished setting up for task 1:0min(Job-28899428-split-0):1. 17: w1m1:  
 Changing status to :busy.  
 17: w1m1: Starting task 1:0min(Job-28899428-split-0):1. 19: w1m1:  
 Finished task 1:0min(Job-28899428-split-0):1. 19: w1m1: Changing  
 status to :idle.  
 19: w2m1: About to set status back to ':busy'. 19: w2m1:  
 Changing status to :busy.  
 19: w2m1: Changing status to :setting\_up.  
 19: w2m1: No set-up required for JobType 1 / 2. 19: w2m1:  
 Changing status to :busy.  
 19: w2m1: Restarting task 1:0min(Job-28899428):2.  
 19: dispo: Attempting to assign task for machine 'w1m1@%portal' on workstation 'w1-supervisor@%portal'. 19:  
 Logging task 1 as complete for portion 1:0min(Job-28899428-split-0); next\_step will be 2.  
 19: JobManager-1-0min-5270791196649818156: Completed task 1:0min(Job-28899428-split-0):1.  
 19: JobManager-1-0min-5270791196649818156: Completed task 1:0min(Job-28899428-split-0):1; about to start task

1:Omin(Job-28899428-split-0):2.  
19: dispo: Adding task for w2-in-buffer@%portal. 25: w2m1:  
Finished task 1:Omin(Job-28899428):2. 25: w2m1: Changing  
status to :idle.  
25: JobManager-1-Omin-5270791196649818156: Partly completed task 1:Omin(Job-28899428):2.  
25: JobManager-1-Omin-5270791196649818156: Completed task 1:Omin(Job-28899428):2; about to start task 1:Omin(Job-  
28899428-split-1):3.  
25: dispo: Attempting to assign task for machine 'w2m1@%portal' on workstation 'w2-supervisor@%portal'. 25: dispo:  
Got task and semaphore for workstation 'w2'.  
25: dispo: Adding task for w3-in-buffer@%portal.  
25: dispo: Got reply from buffer for workstation 'w2'. 25: dispo:  
About to select task (from 1 available).  
25: SIPTSelector: task 1:Omin(Job-28899428-split-0):2 has imminent processing time of 180000 25:  
SIPTSelector: Shortest is 1:Omin(Job-28899428-split-0):2, with time 180000  
25: dispo: Got selected task: '1:Omin(Job-28899428-split-0):2'. 25: dispo:  
Selected task for workstation 'w2'.  
25: dispo: Assigned task for machine 'w2m1@%portal' on workstation 'w2-supervisor@%portal'. 25: w2m1:  
Changing status to :setting\_up.  
25: w2m1: No set-up required for JobType 1 / 2.  
25: w2m1: Changing status to :busy.  
25: w2m1: Starting task 1:Omin(Job-28899428-split-0):2. 26: w3m1:  
Finished task 2:2mins(Job-12285785):2.  
26: w3m1: Changing status to :idle.  
26: Logging task 2 as complete for portion 2:2mins(Job-12285785); next\_step will be 3. 26:  
JobManager-2-2mins-3551044553175055070: Completed task 2:2mins(Job-12285785):2.  
26: JobManager-2-2mins-3551044553175055070: Completed task 2:2mins(Job-12285785):2; about to start task 2:2mins(Job-  
12285785):3.  
26: dispo: Adding task for w2-in-buffer@%portal.  
26: dispo: Attempting to assign task for machine 'w3m1@%portal' on workstation 'w3-supervisor@%portal'. 26: dispo: Got  
task and semaphore for workstation 'w3'.  
26: dispo: Got reply from buffer for workstation 'w3'. 26: dispo:  
About to select task (from 1 available).  
26: SIPTSelector: task 1:Omin(Job-28899428-split-1):3 has imminent processing time of 720000 26:  
SIPTSelector: Shortest is 1:Omin(Job-28899428-split-1):3, with time 720000  
26: dispo: Got selected task: '1:Omin(Job-28899428-split-1):3'. 26: dispo:  
Selected task for workstation 'w3'.  
26: dispo: Assigned task for machine 'w3m1@%portal' on workstation 'w3-supervisor@%portal'. 26: w3m1:  
Changing status to :setting\_up.  
26: w3m1: Setting up for task 1:Omin(Job-28899428-split-1):3. 26: w3m1:  
Set-up time is 0  
26: w3m1: Finished setting up for task 1:Omin(Job-28899428-split-1):3. 26: w3m1:  
Changing status to :busy.

26: w3m1: Starting task 1:0min(Job-28899428-split-1):3. 28: w2m1: Finished task  
 1:0min(Job-28899428-split-0):2. 28: w2m1: Changing status to :idle.  
 28: w3m1: Changing status to :out\_of\_order.  
 28: w3m1: set status to 'out of order' (initial was ':busy'). Waiting for 180000... 28: w3m1:  
 Failed for task 1:0min(Job-28899428-split-1):3.  
 28: Logging task 2 as complete for portion 1:0min(Job-28899428-split-0); next\_step will be 3. 28:  
 JobManager-1-0min-5270791196649818156: Completed task 1:0min(Job-28899428-split-0):2.  
 28: JobManager-1-0min-5270791196649818156: Completed task 1:0min(Job-28899428-split-0):2; about to start task  
 1:0min(Job-28899428-split-0):3.  
 28: JobManager-1-0min-5270791196649818156: Task 1:0min(Job-28899428-split-1):3 partly failed; splitting and trying  
 again.  
 28: JobManager-1-0min-5270791196649818156: About to start replacement task 1:0min(Job-28899428-split-1-  
 split-0):1.  
 28: dispo: Adding task for w1-in-buffer@%portal.  
 28: dispo: Cancelling request to assign task for machine 'w3m1@%portal' on workstation 'w3-  
 supervisor@%portal'.  
 28: dispo: Attempting to assign task for machine 'w2m1@%portal' on workstation 'w2-supervisor@%portal'. 28: dispo: Got  
 task and semaphore for workstation 'w2'.  
 28: dispo: Adding task for w3-in-buffer@%portal.  
 28: dispo: Got task and semaphore for workstation 'w1'. 28: dispo:  
 Got reply from buffer for workstation 'w2'. 28: dispo: About to  
 select task (from 1 available).  
 28: SIPTSelector: task 2:2mins(Job-12285785):3 has imminent processing time of 240000 28:  
 SIPTSelector: Shortest is 2:2mins(Job-12285785):3, with time 240000  
 28: dispo: Got selected task: '2:2mins(Job-12285785):3'.  
 28: dispo: Selected task for workstation 'w2'.  
 28: dispo: Assigned task for machine 'w2m1@%portal' on workstation 'w2-supervisor@%portal'. 28: dispo:  
 Got reply from buffer for workstation 'w1'.  
 28: dispo: About to select task (from 1 available).  
 28: SIPTSelector: task 1:0min(Job-28899428-split-1-split-0):1 has imminent processing time of 120000 28:  
 SIPTSelector: Shortest is 1:0min(Job-28899428-split-1-split-0):1, with time 120000  
 28: dispo: Got selected task: '1:0min(Job-28899428-split-1-split-0):1'. 28: dispo:  
 Selected task for workstation 'w1'.  
 28: dispo: Assigned task for machine 'w1m1@%portal' on workstation 'w1-supervisor@%portal'. 28: w1m1:  
 Changing status to :setting\_up.  
 28: w1m1: No set-up required for JobType 1 / 1. 28: w1m1:  
 Changing status to :busy.  
 28: w1m1: Starting task 1:0min(Job-28899428-split-1-split-0):1. 28: w2m1:  
 Changing status to :setting\_up.  
 28: w2m1: Setting up for task 2:2mins(Job-12285785):3. 28: w2m1:  
 Set-up time is 0  
 28: w2m1: Finished setting up for task 2:2mins(Job-12285785):3. 28: w2m1:

Changing status to :busy.  
 28: w2m1: Starting task 2:2mins(Job-12285785):3.  
 30: w1m1: Finished task 1:0min(Job-28899428-split-1-split-0):1. 30: w1m1:  
 Changing status to :idle.  
 30: dispo: Attempting to assign task for machine 'w1m1@%portal' on workstation 'w1-supervisor@%portal'. 30:  
 Logging task 1 as complete for portion 1:0min(Job-28899428-split-1-split-0); next\_step will be 2. 30: JobManager-  
 1-0min-5270791196649818156: Completed task 1:0min(Job-28899428-split-1-split-0):1.  
 30: JobManager-1-0min-5270791196649818156: Completed task 1:0min(Job-28899428-split-1-split-0):1; about to start task  
 1:0min(Job-28899428-split-1-split-0):2.  
 30: dispo: Adding task for w2-in-buffer@%portal. 31: w3m1:  
 About to set status back to ':busy'. 31: w3m1: Changing  
 status to :busy.  
 31: w3m1: Changing status to :setting\_up.  
 31: w3m1: No set-up required for JobType 1 / 3. 31: w3m1:  
 Changing status to :busy.  
 31: w3m1: Restarting task 1:0min(Job-28899428-split-1):3. 32: w2m1:  
 Finished task 2:2mins(Job-12285785):3.  
 32: w2m1: Changing status to :idle.  
 32: Logging task 3 as complete for portion 2:2mins(Job-12285785); next\_step will be 4.  
 32: JobManager-2-2mins-3551044553175055070: Completed task 2:2mins(Job-12285785):3. Job finished. 32: Removed  
 incomplete portion Job-12285785; now [].  
 32: Comparing 'known\_jobs' (2 jobs) with 'completed\_jobs' (1 jobs) 32:  
 Differences: Set of 1 jobs: {1:0min:5[Job-28899428],}  
 32: dispo: Attempting to assign task for machine 'w2m1@%portal' on workstation 'w2-supervisor@%portal'. 32: dispo:  
 Got task and semaphore for workstation 'w2'.  
 32: dispo: Got reply from buffer for workstation 'w2'. 32: dispo:  
 About to select task (from 1 available).  
 32: SIPTSelector: task 1:0min(Job-28899428-split-1-split-0):2 has imminent processing time of 180000 32:  
 SIPTSelector: Shortest is 1:0min(Job-28899428-split-1-split-0):2, with time 180000  
 32: dispo: Got selected task: '1:0min(Job-28899428-split-1-split-0):2'. 32: dispo:  
 Selected task for workstation 'w2'.  
 32: dispo: Assigned task for machine 'w2m1@%portal' on workstation 'w2-supervisor@%portal'.  
 32: w2m1: Changing status to :setting\_up.  
 32: w2m1: Setting up for task 1:0min(Job-28899428-split-1-split-0):2. 32: w2m1:  
 Set-up time is 0  
 32: w2m1: Finished setting up for task 1:0min(Job-28899428-split-1-split-0):2. 32: w2m1:  
 Changing status to :busy.  
 32: w2m1: Starting task 1:0min(Job-28899428-split-1-split-0):2. 35: w2m1:  
 Finished task 1:0min(Job-28899428-split-1-split-0):2. 35: w2m1: Changing  
 status to :idle.  
 35: Logging task 2 as complete for portion 1:0min(Job-28899428-split-1-split-0); next\_step will be 3. 35:  
 JobManager-1-0min-5270791196649818156: Completed task 1:0min(Job-28899428-split-1-split-0):2.



35: JobManager-1-Omin-5270791196649818156: Completed task 1:Omin(Job-28899428-split-1-split-0):2; about to start task 1:Omin(Job-28899428-split-1-split-0):3.

35: dispo: Attempting to assign task for machine 'w2m1@%portal' on workstation 'w2-supervisor@%portal'. 35: dispo: Adding task for w3-in-buffer@%portal.

40: w3m1: Finished task 1:Omin(Job-28899428-split-1):3. 40: w3m1: Changing status to :idle.

40: JobManager-1-Omin-5270791196649818156: Partly completed task 1:Omin(Job-28899428-split-1):3. This split portion finished.

40: Removed incomplete portion Job-28899428-split-1-split-1; now [Job-28899428].

40: Not all steps complete for portion 1:Omin(Job-28899428); step 3 (of 3 steps) is not complete:

40: dispo: Attempting to assign task for machine 'w3m1@%portal' on workstation 'w3-supervisor@%portal'. 40: dispo: Got task and semaphore for workstation 'w3'.

40: dispo: Got reply from buffer for workstation 'w3'. 40: dispo: About to select task (from 2 available).

40: SIPTSelector: task 1:Omin(Job-28899428-split-1-split-0):3 has imminent processing time of 180000 40: SIPTSelector: task 1:Omin(Job-28899428-split-0):3 has imminent processing time of 180000

40: SIPTSelector: Shortest is 1:Omin(Job-28899428-split-0):3, with time 180000 40: dispo: Got selected task: '1:Omin(Job-28899428-split-0):3'.

40: dispo: Selected task for workstation 'w3'.

40: dispo: Assigned task for machine 'w3m1@%portal' on workstation 'w3-supervisor@%portal'. 40: w3m1: Changing status to :setting\_up.

40: w3m1: No set-up required for JobType 1 / 3. 40: w3m1: Changing status to :busy.

40: w3m1: Starting task 1:Omin(Job-28899428-split-0):3. 43: w3m1: Finished task 1:Omin(Job-28899428-split-0):3. 43: w3m1: Changing status to :idle.

43: Logging task 3 as complete for portion 1:Omin(Job-28899428-split-0); next\_step will be 4.

43: JobManager-1-Omin-5270791196649818156: Completed task 1:Omin(Job-28899428-split-0):3. Job finished. 43: Removed incomplete portion Job-28899428-split-0; now [Job-28899428].

43: Not all steps complete for portion 1:Omin(Job-28899428); step 3 (of 3 steps) is not complete:

43: dispo: Attempting to assign task for machine 'w3m1@%portal' on workstation 'w3-supervisor@%portal'. 43: dispo: Got task and semaphore for workstation 'w3'.

43: dispo: Got reply from buffer for workstation 'w3'. 43: dispo: About to select task (from 1 available).

43: SIPTSelector: task 1:Omin(Job-28899428-split-1-split-0):3 has imminent processing time of 180000 43: SIPTSelector: Shortest is 1:Omin(Job-28899428-split-1-split-0):3, with time 180000

43: dispo: Got selected task: '1:Omin(Job-28899428-split-1-split-0):3'. 43: dispo: Selected task for workstation 'w3'.

43: dispo: Assigned task for machine 'w3m1@%portal' on workstation 'w3-supervisor@%portal'.

43: w3m1: Changing status to :setting\_up.

43: w3m1: No set-up required for JobType 1 / 3. 43: w3m1: Changing status to :busy.

43: w3m1: Starting task 1:0min(Job-28899428-split-1-split-0):3. 46: w3m1:  
Finished task 1:0min(Job-28899428-split-1-split-0):3. 46: w3m1: Changing  
status to :idle.  
46: Logging task 3 as complete for portion 1:0min(Job-28899428-split-1-split-0); next\_step will be 4. 46:  
JobManager-1-0min-5270791196649818156: Completed task 1:0min(Job-28899428-split-1-split-0):3. Job finished.  
46: Removed incomplete portion Job-28899428-split-1-split-0; now [Job-28899428].  
46: Merging component 1:0min(Job-28899428-split-0) (size 1) into 1:0min(Job-28899428) 46: Merging  
component 1:0min(Job-28899428-split-1) (size 4) into 1:0min(Job-28899428) 46: Removed incomplete  
portion Job-28899428; now [].  
46: Comparing 'known\_jobs' (2 jobs) with 'completed\_jobs' (2 jobs)  
46: dispo: Attempting to assign task for machine 'w3m1@%portal' on workstation 'w3-supervisor@%portal'. 46: w1m1:  
Changing status to :finished.  
46: w2m1: Changing status to :finished. 46: w3m1:  
Changing status to :finished.

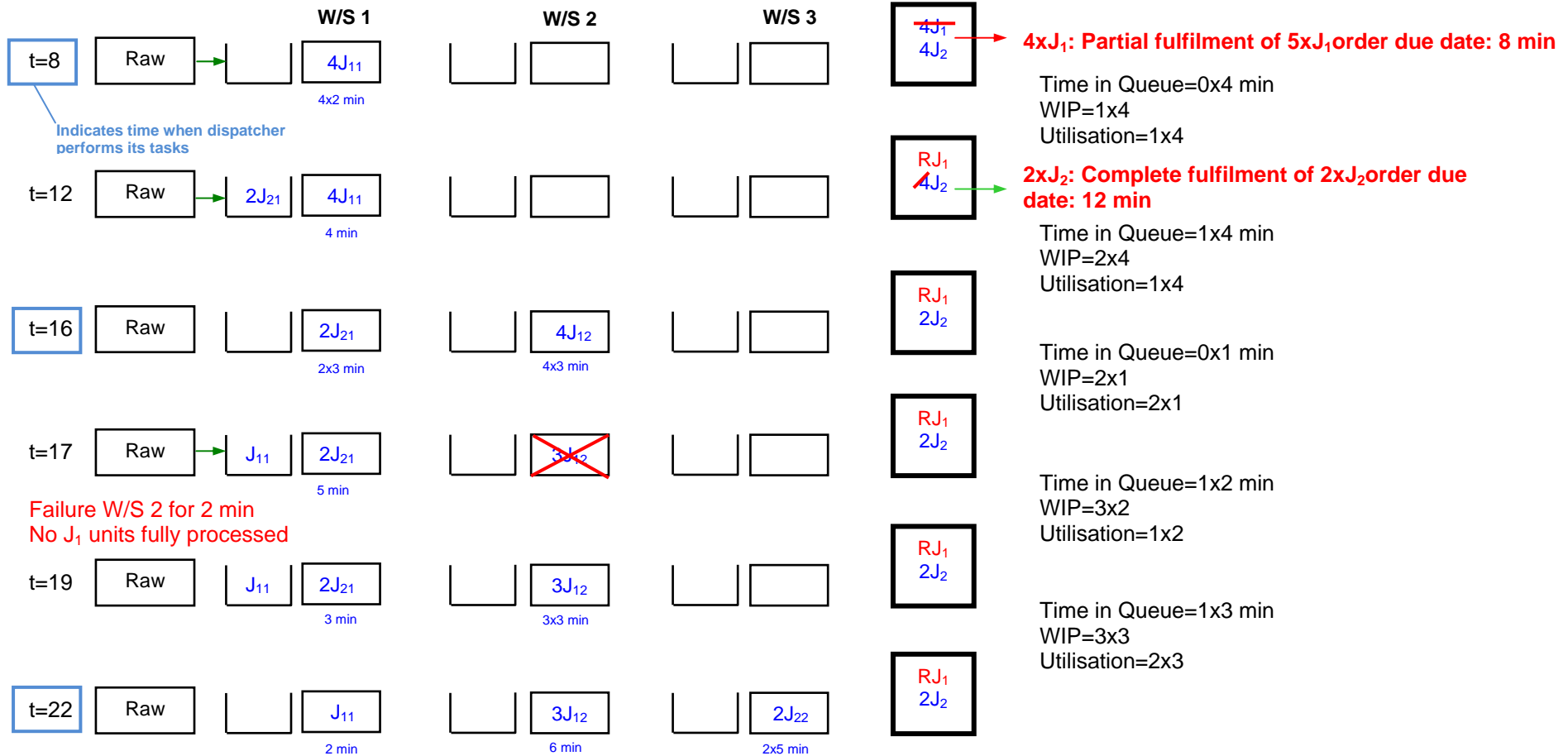
**APPENDIX F**  
**BASS verification using experiment BVE2**

**CONWIP (SIPT/Failure/Damage)**

Initial System Load: 4 of all job types

Machine Failure at t=17 min and t=23 min resulting in partial damage of batch in progress

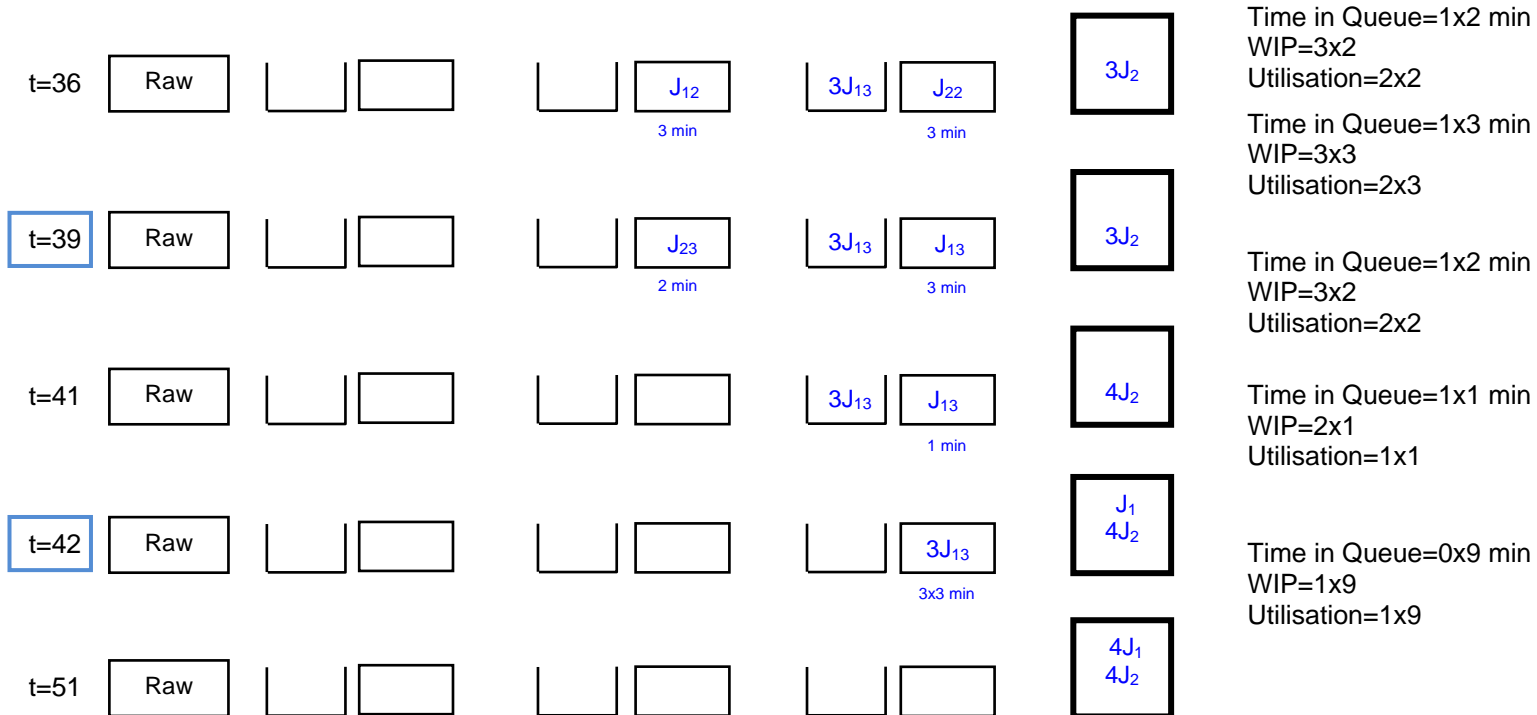
time	Job	Machine (Process time)			TPT	Due Date=arrival time+1xTPT
0	5xJ <sub>1</sub>	J <sub>11</sub> =1(2)	J <sub>12</sub> =2(3)	J <sub>13</sub> =3(3)	8	DD=0+1x8=8
2	2xJ <sub>2</sub>	J <sub>21</sub> =1(3)	J <sub>22</sub> =3(5)	J <sub>23</sub> =2(2)	10	DD=2+1x10=12







Replenishment



Performance Metric	Experimentation Results
Number of Tardy Jobs	= 1 job
Total Absolute Deviation of Earliness/Tardiness	=  (34-8)+(12-12) =26 min
Average Flow Time (Completion-Ready)/No. of Jobs	= [(34-0)+(12-2)]/2=22 min
Throughput (Completion-Start)	= (34-8)+(12-12)=26 min
Makespan (Completion Last-Start First)	= 34-8=26 min
Time in Queue to Fulfilment	= 0+4+0+2+3+0+1+2+1+2+6+4+2+2+3+2+1+0=27 min
Time in Queue including Replenishment	= 0+4+0+2+3+0+1+2+1+2+6+4+2+2+3+2+1+0=35 min
WIP to Fulfilment	= [(1x4)+(2x4)+(2x1)+(3x2)+(3x3)+(3x1)+(4x1)+(4x2)+(4x1)+(4x1)+(4x3)+(4x2)+(3x1)]/34=75/34=2.21 jobs
WIP including Replenishment	= [(1x4)+(2x4)+(2x1)+(3x2)+(3x3)+(3x1)+(4x1)+(4x2)+(4x1)+(4x1)+(4x3)+(4x2)+(3x1)+(3x2)+(3x3)+(3x2)+(2x1)+(1x9)]/51=107/51=2.10 jobs
Machine Utilisation to Fulfilment	= [(1x4)+(1x4)+(2x1)+(1x2)+(2x3)+(3x1)+(2x1)+(2x2)+(3x1)+(2x1)+(2x3)+(2x2)+(1x1)]/(3x34)=43/102=42.16%
Machine Utilisation including Replenishment	= [(1x4)+(1x4)+(2x1)+(1x2)+(2x3)+(3x1)+(2x1)+(2x2)+(3x1)+(2x1)+(2x3)+(2x2)+(1x1)+(2x2)+(2x3)+(2x2)+(1x1)+(1x9)]/(3x51)=67/153=43.79%
Set-up Time to Fulfilment	=0 min
Set-up Time including Replenishment	=0 min
Fill Rate	= 50.00%

### JACK™ output

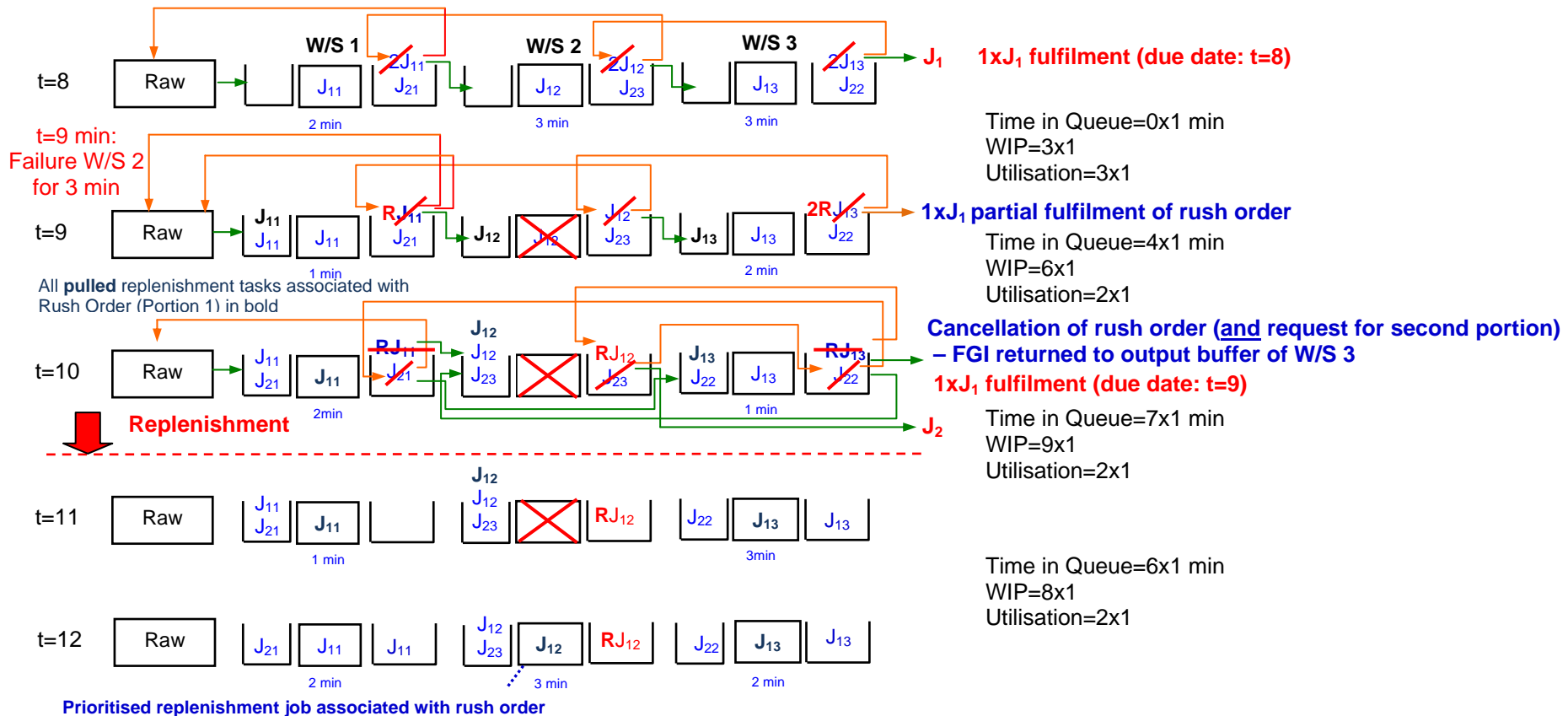
Total throughput time (mins): 26.00  
 Average flow time (mins): 22.00  
 Makespan (mins): 26.00  
 Earliness (mins): 0.00  
 Tardiness (mins): 26.00  
 Earliness/Tardiness (mins): 26.00  
 Number of tardy jobs: 1  
 Fill rate: 50.00  
 Average machine utilisation to fulfilment: 42.16  
 Average machine utilisation: 43.79  
 WIP level to fulfilment: 2.21  
 WIP level: 2.10  
 Time in queue to fulfilment (mins): 27.00  
 Total time in queue (mins): 35.00  
 Set-up time to fulfilment (mins): 0.00  
 Total set-up time (mins): 0.00  
 Number of jobs finished at termination: 2

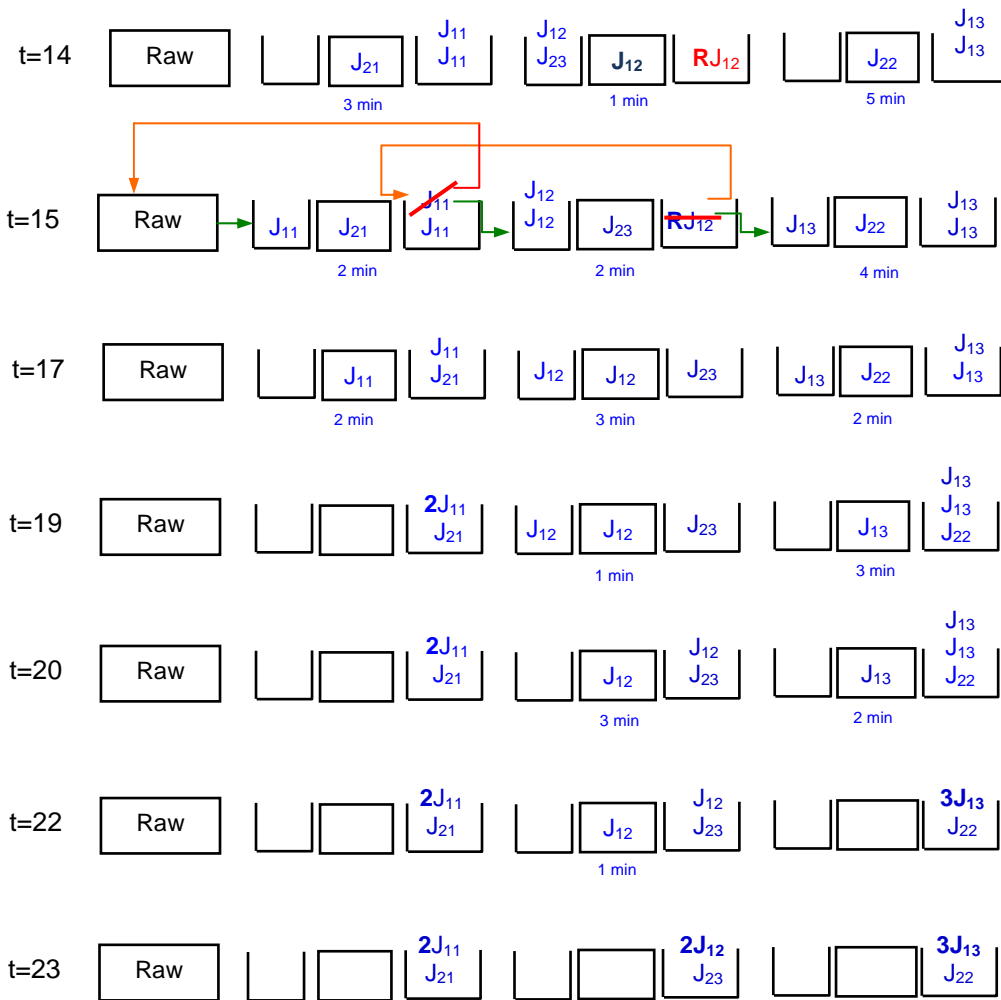
**APPENDIX G**  
**BASS verification using experiment BVE3**



**Kanban (SIPT/Failure/Damage/Rush Order/Cancellation of Rush Order at t=10, i.e. after its release and partial fulfilment)**

time	Job	Machine (Process time)			TPT	Due Date=arrival time+1xTPT
0	J <sub>1</sub>	J <sub>11</sub> =1(2)	J <sub>12</sub> =2(3)	J <sub>13</sub> =3(3)	8	DD=0+1x8=8
0	J <sub>2</sub>	J <sub>21</sub> =1(3)	J <sub>22</sub> =3(5)	J <sub>23</sub> =2(2)	10	DD=0+1x10=10
1	J <sub>1</sub>	J <sub>11</sub> =1(2)	J <sub>12</sub> =2(3)	J <sub>13</sub> =3(3)	8	DD=1+1x8=9
9	2xJ <sub>1</sub> Rush Order	J <sub>11</sub> =1(2)	J <sub>12</sub> =2(3)	J <sub>13</sub> =3(3)	8	DD=9+1x8=17





Time in Queue=4x2 min  
 WIP=7x2  
 Utilisation=3x2

Time in Queue=2x1 min  
 WIP=5x1  
 Utilisation=3x1

Time in Queue=4x2 min  
 WIP=7x2  
 Utilisation=3x2

Time in Queue=2x2 min  
 WIP=5x2  
 Utilisation=3x2

Time in Queue=1x1 min  
 WIP=3x1  
 Utilisation=2x1

Time in Queue=0x2 min  
 WIP=2x2  
 Utilisation=2x2

Time in Queue=0x1 min  
 WIP=1x1  
 Utilisation=1x1

Performance Metric	Experimentation Results
Number of Tardy Jobs	= 1 job
Total Absolute Deviation of Earliness/Tardiness	=  (8-8)+(10-10)+(10-9) =1.00 min
Average Flow Time (Completion-Ready)/No. of Jobs	= [(8-0)+(10-0)+(10-1)]/3=9.00 min
Throughput (Completion-Start)	= (8-8)+(10-10)+(10-9)=1.00 min
Makespan (Completion Last-Start First)	= 10-8=2.00 min
Time in Queue to Fulfilment	= 0+4=4.00 min
Time in Queue including Replenishment	= 0+4+7+6+8+2+8+4+1+0+0=40.00min
WIP to Fulfilment	= (3x1)+(6x1)]/10=9/10=0.90 jobs
WIP including Replenishment	= [(3x1)+(6x1)+(9x1)+(8x1)+(7x2)+(5x1)+(7x2)+(5x2)+(3x1)+(2x2)+(1x1)]/23=77/23=3.35 jobs
Machine Utilisation to Fulfilment	= [(3x1)+(2x1)]/(3x10)=5/30=16.67%
Machine Utilisation including Replenishment	= [(3x1)+(2x1)+(2x1)+(2x1)+(3x2)+(3x1)+(3x2)+(3x2)+(2x1)+(2x2)+(1x1)]/(3x23)=37/69=53.62%
Fill Rate	=66.67%

### JACK™ output

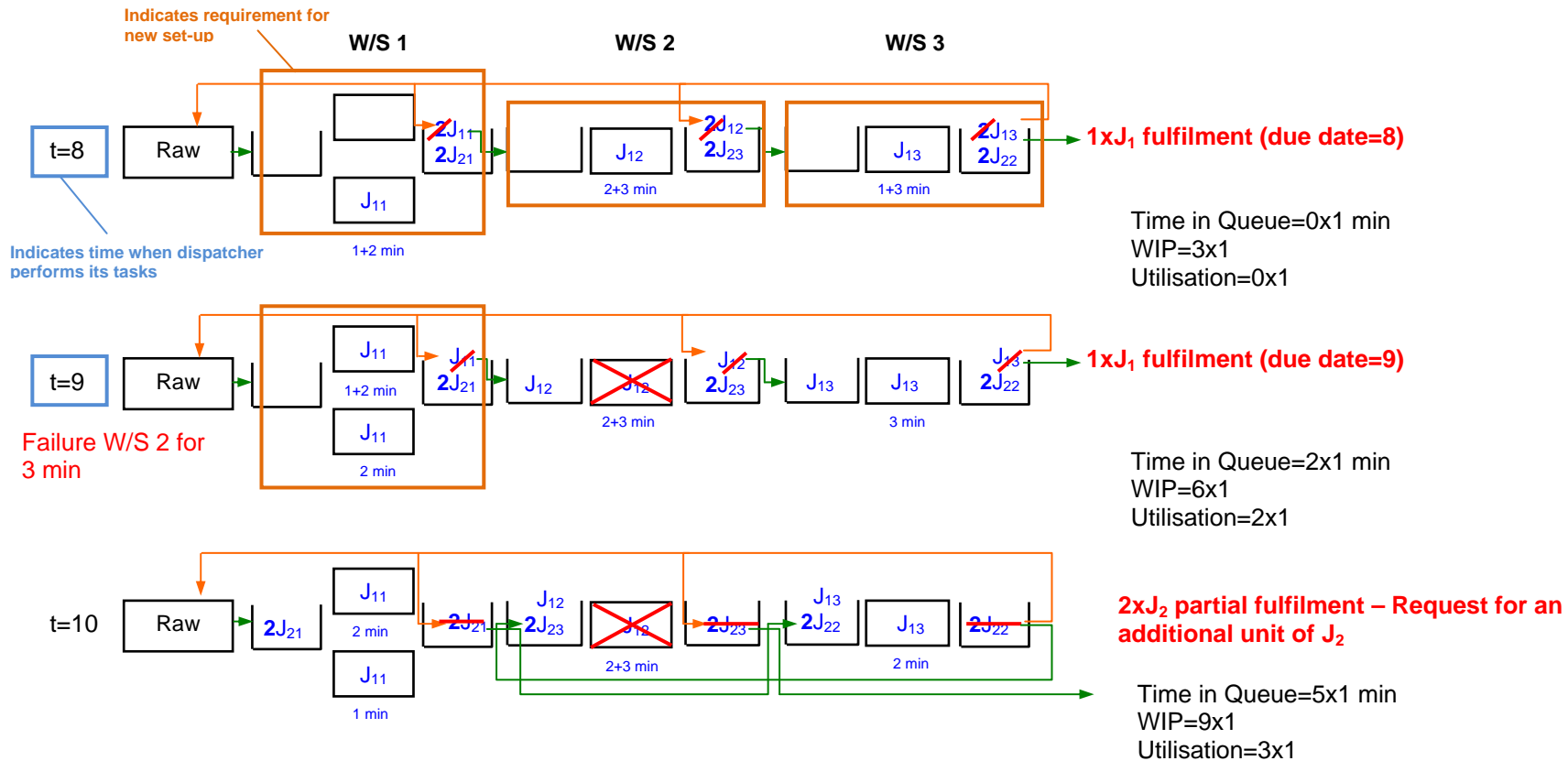
Total throughput time (mins): 1.00  
 Average flow time (mins): 9.00  
 Makespan (mins): 2.00  
 Earliness (mins): 0.00  
 Tardiness (mins): 1.00  
 Earliness/Tardiness (mins): 1.00  
 Number of tardy jobs: 1  
 Fill rate: 66.67  
 Average machine utilisation to fulfilment: 16.67  
 Average machine utilisation: 53.62  
 WIP level to fulfilment: 0.90  
 WIP level: 3.35  
 Time in queue to fulfilment (mins): 4.00  
 Total time in queue (mins): 40.00  
 Set-up time to fulfilment (mins): 0.00  
 Total set-up time (mins): 0.00  
 Number of jobs finished at termination: 3

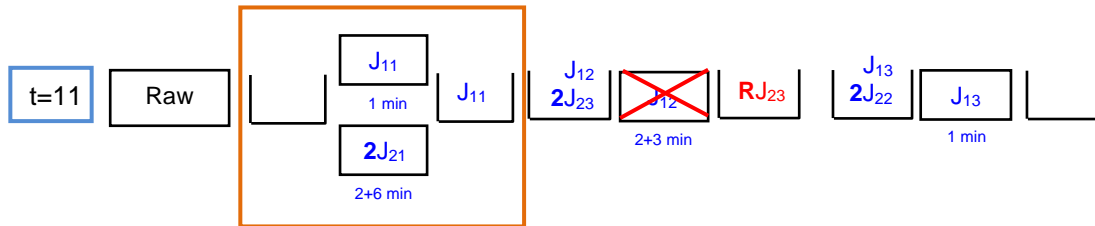
**APPENDIX H**  
**BASS verification using experiment BVE4**

**Base stock (SIPT/Failure/No Damage/Set-up times)**

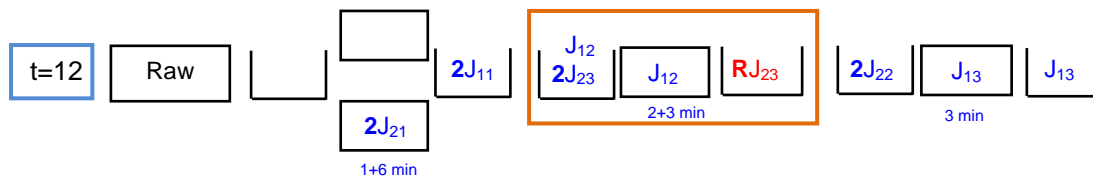
Inventory for  $J_1$  and  $J_2$  set to 2. At time 0 a new order for  $J_2$  batch size=3 is received. Due date remains  $t=10$ . Portions of jobs are processed as batches.

time	Job	Machine/Process time/Set-up time	TPT	Due Date=arrival time+1xTPT
0	$J_1$	$J_{11}=1/2/1$ $J_{12}=2/3/2$ $J_{13}=3/3/1$	8	$DD=0+1 \times 8=8$
0	$J_2$	$J_{21}=1/3/2$ $J_{22}=3/5/3$ $J_{23}=2/2/1$	10	$DD=0+1 \times 10=10$
1	$J_1$	$J_{11}=1/2/1$ $J_{12}=2/3/2$ $J_{13}=3/3/1$	8	$DD=1+1 \times 8=9$

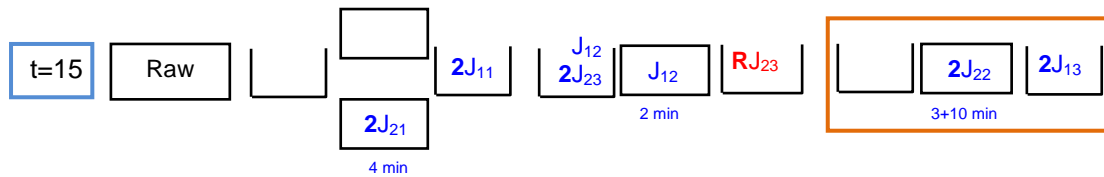




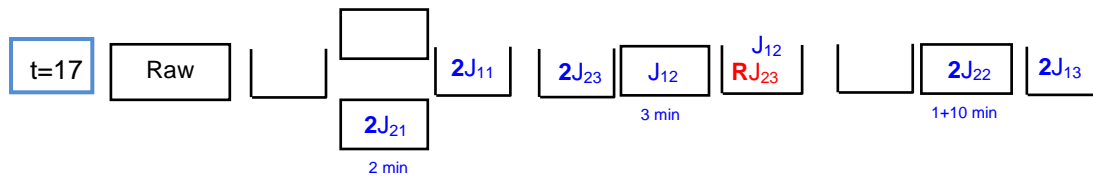
Time in Queue=4x1 min  
 WIP=8x1  
 Utilisation=2x1



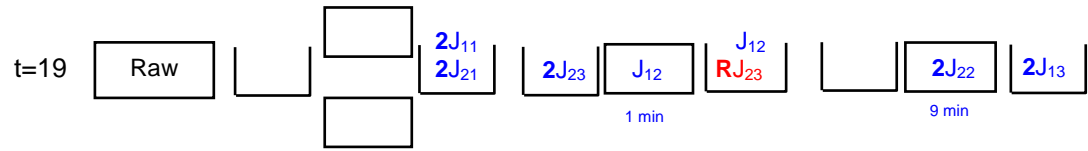
Time in Queue=3x3 min  
 WIP=6x3  
 Utilisation=1x2+1x1+1x3=6x1



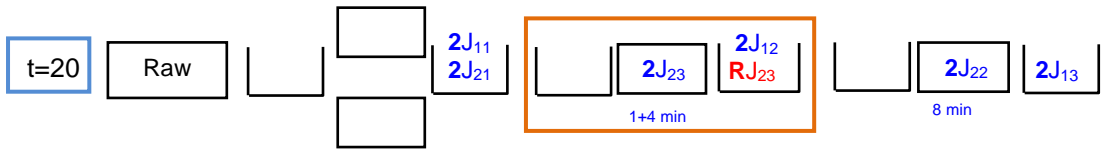
Time in Queue=2x2 min  
 WIP=5x2  
 Utilisation=2x2



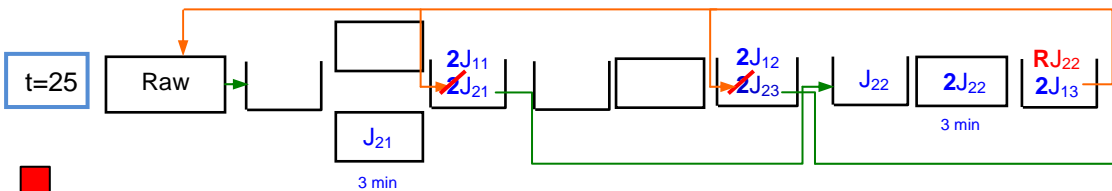
Time in Queue=1x2 min  
 WIP=4x2  
 Utilisation=2x2+1x1=5x1



Time in Queue=1x1 min  
 WIP=3x1  
 Utilisation=2x1



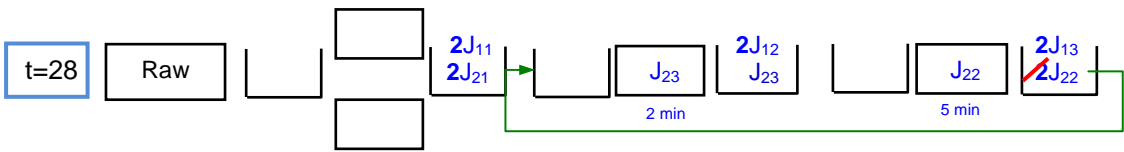
Time in Queue=0x5 min  
 WIP=2x5  
 Utilisation=1x4+1x5=9x1



**J<sub>2</sub> Complete fulfilment of batch order**

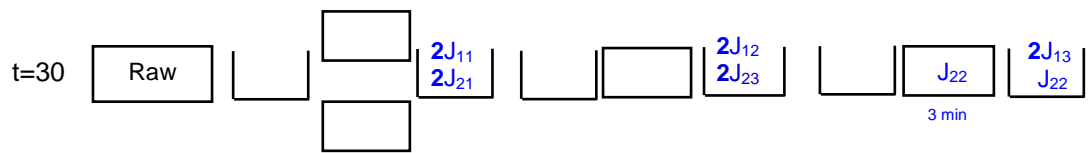


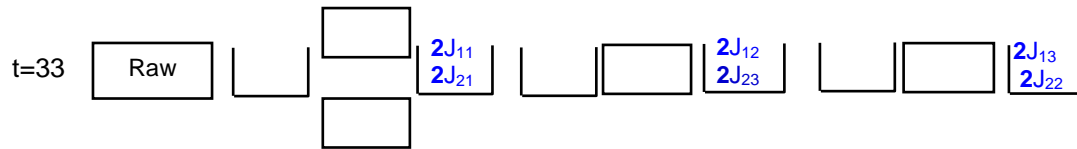
**Replenishment**



Time in Queue=1x3 min  
 WIP=3x3  
 Utilisation=2x3

Time in Queue=0x2 min  
 WIP=2x2  
 Utilisation=2x2





Time in Queue=0x3 min  
 WIP=1x3  
 Utilisation=1x3

Performance Metric	Experimentation Results
Number of Tardy Jobs	= 1 job
Total Absolute Deviation of Earliness/Tardiness	=  (8-8)+(9-9)+(25-10) =15 min
Average Flow Time (Completion-Ready)/No.of Jobs	= [(8-0)+(9-1)+(25-0)]/3=13.67 min
Throughput (Completion-Start)	= (8-8)+(9-9)+(25-10)=15 min
Makespan (Completion Last-Start First)	= 25-8=17 min
Time in Queue to Fulfilment	= 0+2+5+4+9+4+2+1+0=27 min
Time in Queue including Replenishment	= 0+2+5+4+9+4+2+1+0+3+0+0=30 min
WIP to Fulfilment	= [(3x1)+(6x1)+(9x1)+(8x1)+(6x3)+(5x2)+(4x2)+(3x1)+(2x5)]/25=75/25=3.00
WIP including Replenishment	= [(3x1)+(6x1)+(9x1)+(8x1)+(6x3)+(5x2)+(4x2)+(3x1)+(2x5)+(3x3)+(2x2)+(1x3)]/33=91/33=2.76 jobs
Machine Utilisation to Fulfilment	= [(0x1)+(2x1)+(3x1)+(2x1)+(6x1)+(2x2)+(5x1)+(2x1)+(9x1)]/(4x25)=33/100=33%
Machine Utilisation including Replenishment	= [(0x1)+(2x1)+(3x1)+(2x1)+(6x1)+(2x2)+(5x1)+(2x1)+(9x1)+(2x3)+(2x2)+(1x3)]/(4x33)=46/132=34.85%
Set-up Time to Fulfilment	=(1+2+1+1+2+2+3+1)-1=12 min
Set-up Time including Replenishment	=(1+2+1+1+2+2+3+1)-1=12 min
Fill Rate	= 66.67%

**JACK™ output**

Total throughput time (mins): 15,00  
 Average flow time (mins): 13,67  
 Makespan (mins): 17,00  
 Earliness (mins): 0,00  
 Tardiness (mins): 15,00  
 Earliness/Tardiness (mins): 15,00  
 Number of tardy jobs: 1  
 Fill rate: 66,67  
 Average machine utilisation to fulfilment: 33,00



Average machine utilisation: 34,85  
WIP level to fulfilment: 3,00  
WIP level: 2,76  
Time in queue to fulfilment (mins): 27,00  
Total time in queue (mins): 30,00  
Set-up time to fulfilment (mins): 12,00  
Total set-up time (mins): 12,00  
Number of jobs finished at termination: 3

**APPENDIX I**  
**Stochastic data used in experiment LA40**

**LA40 (15x15): 25 Instances**

**Dynamic Arrivals**

$$\text{Expo Mean (Arrivals)} = \frac{\mu_p \times \mu_g}{U \times m} = \frac{49 \times 15}{0.80 \times 15} = 62$$

$\mu_p = 49$  (sec)

$\mu_g = 15$

U = 80%

m = 15

**Machine Breakdowns**

(affecting 30% of machines)

$$\text{Expo MTTR} = 1.5 \times \text{ePT} = 1.5 \times 49 = 74$$

ePT = 49 (sec)

**Rush orders**

Arrivals: 5 times less frequent.

Proportion: 20%

Order Number	Job type Uniform [1,15]	Job Arrival Time Expo (62)	Rounded BatchSizes Tria [50,150,250]	Machine Affected			Job type Uniform [1,15]	Job Arrival Time Expo (310)	Rounded BatchSizes Tria [50,150,250]
				Uniform [0,14]	MTBF Expo (1000)	MTTR Expo (74)			
1	10	71	210	13	1007	8	15	113	60
2	2	128	120	4	3042	37	6	142	220
3	11	212	120	3	3914	73	6	294	140
4	6	216	210	13	6289	113	7	596	200
5	2	388	120	4	6652	40	10	1065	200
6	4	482	150				5	1232	110
7	6	539	110						
8	9	543	180						
9	3	691	230						
10	2	768	200						
11	7	914	170						
12	13	1067	130						
13	2	1099	180						
14	8	1121	140						
15	11	1327	150						
16	3	1391	190						
17	14	1518	130						
18	10	1573	110						
19	10	1721	140						
20	12	1744	200						
21	6	1751	190						
22	4	1782	150						
23	4	1843	160						
24	14	1938	90						
25	9	1972	170						

**APPENDIX J**  
**Stochastic data used in experiment ABZ7**

**ABZ7 (20x15): 30 Instances**

**Dynamic Arrivals**

$$\text{Expo Mean (Arrivals)} = \frac{\mu_p \times \mu_g}{U \times m} = \frac{24 \times 15}{0.80 \times 15} = 30$$

$\mu_p = 24$  (sec)  
 $\mu_g = 15$   
 $U = 80\%$   
 $m = 15$

Order Number	Job type Uniform [1,20]	Job Arrival Time Expo (30)	Rounded
			BatchSizes Tria [50,150,250]
1	13	35	100
2	8	63	130
3	15	104	140
4	9	107	140
5	10	190	160
6	16	237	120
7	7	265	70
8	4	267	220
9	9	339	120
10	17	377	90
11	16	448	110
12	10	523	120
13	12	539	170
14	3	550	140
15	13	650	150
16	16	681	220
17	13	743	160
18	6	770	90
19	9	843	210
20	19	855	130
21	17	858	70
22	9	874	90
23	2	904	90
24	12	950	90
25	11	967	80
26	16	1010	160
27	4	1013	90
28	15	1034	170
29	3	1043	140
30	11	1048	80

**Machine Breakdowns**  
 (affecting 30% of machines)

$$\text{Expo MTTR} = 1.5 \times \text{ePT} = 1.5 \times 24 = 36$$

ePT=24 (sec)

Machine Affected Uniform [0,14]	MTBF Expo (1000)	MTTR
		Expo (36)
7	1029	26
3	1854	85
11	2085	9
5	3908	48
12	4060	16

**Rush orders**  
 Arrivals: 5 times less frequent.

Proportion: 20%

Job type Uniform [1,20]	Job Arrival Time Expo (150)	Rounded
		BatchSizes Tria [50,150,250]
1	208	130
19	221	120
18	323	170
2	363	150
4	383	60
5	538	170
14	663	180
9	698	80

**APPENDIX K**  
**Stochastic data used in experiment ABZ9**

**ABZ9 (20x15): 30 Instances**  
**Dynamic Arrivals**

$$\text{Expo Mean (Arrivals)} = \frac{\mu_p \times \mu_g}{U \times m} = \frac{24 \times 15}{0.80 \times 15} = 30$$

$\mu_p = 24$  (sec)  
 $\mu_g = 15$   
 $U = 80\%$   
 $m = 15$

Order Number	Job type Uniform [1,20]	Job Arrival Time Expo (30)	Rounded BatchSizes Tria [50,150,250]
1	3	31	160
2	7	93	130
3	8	120	100
4	9	192	180
5	13	204	110
6	6	208	130
7	1	224	130
8	19	254	120
9	5	300	150
10	2	317	160
11	4	359	170
12	5	363	130
13	14	384	90
14	9	393	160
15	10	398	130
16	19	429	200
17	13	455	190
18	3	463	130
19	18	519	100
20	7	524	140
21	1	546	60
22	3	618	200
23	3	626	130
24	3	666	70
25	2	668	140
26	13	693	220
27	3	710	140
28	15	718	200
29	8	729	200
30	2	751	110

**Machine Breakdowns**  
(affecting 30% of machines)

$$\text{Expo MTTR} = 1.5 \times \text{ePT} = 1.5 \times 24 = 36$$

ePT=24 (sec)

Machine Affected Uniform [0,14]	MTBF Expo (1000)	MTTR Expo (36)
12	1802	43
6	4046	24
1	4643	15
4	6662	84
8	8248	42

**Rush orders**  
Arrivals: 5 times less frequent.

Proportion: 20%

Job type Uniform [1,20]	Job Arrival Time Expo (150)	Rounded BatchSizes Tria [50,150,250]
10	274	70
19	298	170
13	405	50
16	758	160
18	793	70
7	989	90
1	998	140
17	1120	60

**APPENDIX L**  
**Stochastic data used in experiment YN1**



**YN1 (20x20): 30 Instances**

**Dynamic Arrivals**

$$\text{ExpoMean(Arrivals)} = \frac{\mu_p \times \mu_g}{U \times m} = \frac{29 \times 20}{0.80 \times 20} = 36$$

$$\mu_p = 29 \text{ (sec)}$$

$$\mu_g = 20$$

$$U = 80\%$$

$$m = 20$$

**Machine Breakdowns**

(affecting 30% of machines)

$$\text{ExpoMTTR} = 1.5 \times \text{ePT} = 1.5 \times 29 = 44$$

$$\text{ePT} = 29 \text{ (sec)}$$

**Rush orders**

Arrivals: 5 times less frequent.

Proportion: 20%

Order Number	Job type Uniform [1,20]	Job Arrival Time Expo (36)	Rounded BatchSizes		Machine Affected Uniform [0,19]	MTBF Expo (1000)	MTTR Expo (44)	Job type Uniform [1,20]	Job Arrival Time Expo (180)	Rounded BatchSizes	
			Tria [50,150,250]	Tria [50,150,250]						Tria [50,150,250]	Tria [50,150,250]
1	3	37	160	160	16	3560	48	11	94	130	130
2	7	111	130	130	14	3624	17	4	139	80	80
3	8	144	100	100	6	3759	22	2	198	70	70
4	9	230	180	180	2	5648	111	13	324	70	70
5	13	244	110	110	7	6929	19	16	513	160	160
6	6	248	130	130	0	7576	66	15	559	170	170
7	1	267	130	130	17	7958	38	8	755	80	80
8	19	303	120	120				12	1016	120	120
9	5	358	150	150							
10	2	378	160	160							
11	4	429	170	170							
12	5	433	130	130							
13	14	458	90	90							
14	9	468	160	160							
15	10	474	130	130							
16	19	512	200	200							
17	13	543	190	190							
18	3	552	130	130							
19	18	619	100	100							
20	7	625	140	140							
21	1	652	60	60							
22	3	737	200	200							
23	3	746	130	130							
24	3	794	70	70							
25	2	797	140	140							
26	13	827	220	220							
27	3	847	140	140							
28	15	856	200	200							
29	8	869	200	200							
30	2	895	110	110							

**APPENDIX M**  
**Stochastic data used in experiment YN4**

**YN4 (20x20): 30 Instances**

**Dynamic Arrivals**

$$\text{Expo Mean(Arrivals)} = \frac{\mu_p \times \mu_g}{U \times m} = \frac{29 \times 20}{0.80 \times 20} = 36$$

$\mu_p = 29$  (sec)  
 $\mu_g = 20$   
 $U = 80\%$   
 $m = 20$

Order Number	Job type Uniform [1,20]	Job Arrival Time Expo (36)	Rounded BatchSizes Tria [50,150,250]
1	3	42	170
2	7	76	130
3	8	124	180
4	9	127	140
5	13	228	150
6	6	283	190
7	1	316	130
8	19	319	110
9	5	405	140
10	2	451	200
11	4	536	190
12	5	625	150
13	14	644	160
14	9	657	90
15	10	777	170
16	19	815	190
17	13	889	160
18	3	921	120
19	18	1008	140
20	7	1022	220
21	1	1026	190
22	3	1044	140
23	3	1080	80
24	3	1136	160
25	2	1156	150
26	13	1207	180
27	3	1211	110
28	15	1236	180
29	8	1246	100
30	2	1252	150

**Machine Breakdowns**  
 (affecting 30% of machines)

$$\text{ExpoMTTR} = 1.5 \times \text{ePT} = 1.5 \times 29 = 44$$

$\text{ePT} = 29$  (sec)

Machine Affected Uniform [0,19]	MTBF Expo (1000)	MTTR Expo (44)
11	1549	11
13	1840	79
1	3846	66
4	4367	47
17	7255	64
1	7746	87
2	7940	13

**Rush orders**

Arrivals: 5 times less frequent.

Proportion: 20%

Job type Uniform [1,20]	Job Arrival Time Expo (180)	Rounded BatchSizes Tria [50,150,250]
4	105	90
8	298	80
12	382	80
3	409	60
11	629	120
9	810	120
17	832	180
20	1473	150