Durban University of Technology

A hybrid simulation technique to predict the effects of human deterioration and learning in an industrial environment

2009

Graeme Kenneth Hay

A hybrid simulation technique to predict the effects of human deterioration and learning in an industrial environment

By

Graeme Kenneth Hay

B.Tech. Engineering (Industrial)

Supervisor: Prof. Kevin Jan Duffy

A thesis submitted in fulfilment of the academic requirements for the degree of M.Tech. in the Department of Industrial Engineering

Durban University of Technology

Durban, September 2009

Declaration

I, Graeme Kenneth Hay (Reg No. 19851563), hereby declare the contents of this thesis entitled **A hybrid simulation technique to predict the effects of human deterioration and learning in an industrial environment** is a true reflection of my own work, and that this thesis has not been submitted, in whole or part, for a degree to any other academic institution.

Graeme Kenneth Hay
September 2009

Abstract

Process simulation is an effective tool when used to simulate a system where a great deal of data exists for the process. This technique is however limited when it comes to simulating certain non-deterministic parts such as human behaviour and interaction, for which there may not be a great amount of data available.

This work creates a unique hybrid model through the combination of process simulation with agent based simulation that simulates the non-deterministic parts of the process, as well as the deterministic parts.

An actual industrial system forms the basis for the research, and the hybrid model is used to understand the effects that human deterioration has on the productivity of this system, as well exploring different scenarios that could lead to improved performance.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Introduction

Manufacturing in South Africa faces a major challenge because disease, a lack of appropriate skills, and economic conditions are preventing workers from matching the productivity levels of the developed Eastern and Western manufacturing countries. These conditions cause workers to experience deterioration in performance over time as their health worsens.

This deterioration poses a number of difficulties for manufacturing managers because they incur the same fixed costs whether the worker is healthy or sick, yet the difference in output between these two types can be markedly different.

Managers need a relatively simple tool to be able to predict the effect this deterioration has on productivity so that they can develop alternate solutions and contingencies.

The purpose of this dissertation is to develop such a tool by answering the following two questions:

1.1.1. What effect does employee deterioration have on system productivity?

1.1.2. Does defined management intervention on a deteriorating system have a positive influence on productivity?

For this work, “deterioration” is defined as a decrease in worker productivity over time, and “defined management intervention” is defined as a structured intervention plan that regulates a manager’s movement and interaction with the people under his / her control.

A system is generally understood to consist of a number of objects that are connected together through regular interaction, and exists to serve a common purpose.

1.2. Method

When conducting experiments to test a theory, it is usually best to use the real system (if it exists) to obtain the most accurate results because the many variables that may influence the outcome are present in that environment. This is not always possible in the manufacturing environment because many production facilities are not able to economically accommodate the experimentation. If this is so, then the next best option is to use simulation [8].

The application of simulation and the development of a model unique to this problem are the underlying themes of this dissertation. Broadly speaking, simulation is the mimicking of the behaviour of a system using a broad collection of methods and applications [1]. In many cases, computers and specialised software are used to facilitate the process.

The aim of the simulation modeller is to create a model of the system to be studied so that a number of experiments can be conducted. In general terms a model is a representation of a real world system that has been appropriately scaled according to the modeller’s requirements. In order to optimise the modelling process, the modeller identifies the key features of the system in question, and includes these in the model. This is commonly known as abstraction.

Simulation is an effective tool that provides decision makers with a platform to test the consequences of specific actions on a system, to aid in good decision making.

In order to create a platform that is able to answer the questions posed in this work, a simulation model was required that could perform the following tasks:

1.3. Simulate the manufacturing process

1.4. Simulate the deterioration of operators and the subsequent effects on the manufacturing process

1.5. Simulate the interaction of a manager with the operators, and the subsequent effects on the manufacturing process

Deterioration in the work place due to disease or economic hardships is a problem that is not unique to South Africa, and is experienced in Eastern Europe and South America, where similar conditions exist. Results of the questions posed are especially relevant to South Africa, and thus it was essential that real data from a South African manufacturing system be used for experimentation. A Durban based Logistics Company agreed to assist in this work by providing data from their system.

In the second chapter of this dissertation, the theory and methodology of computer simulation is explored and the three main approaches to simulation will be introduced and defined. These are:

- System Dynamics
- Discrete Event
- Agent Based

System Dynamics (SD) was developed at MIT in 1956 and involves the use of interlinked cause-effect relationships to simulation complex systems. SD is often used to simulate macro systems such as the economic or social models that have many complex inputs, but model development is time consuming and technically complex to achieve.

Discrete Event (DE) simulation is often used to simulate processes where entity state changes occur at specific points in time. Manufacturing and service processes are most often simulated using DE simulation techniques due to the ease of model construction; however a great deal of information about the process is required in order to develop an accurate model.

Agent Based (AB) simulation has traditionally been used by academics to simulate complex systems at an entity level (e.g. a virus infecting healthy cells). It is a technique that is receiving more attention due to improved software and greater awareness of its applications. The relative ease with which complex systems can be simulated offsets the specialised programming skills needed to develop AB simulation models.

It is however, the combination of the DE and AB simulation techniques to form a hybrid simulation model that are of interest in this work, and the concept is discussed in Chapter 3. The use of hybrid simulation techniques has been explored before, with Remondino [1] developing an Agent Based Process Simulation model that inspired the approach used in this work.

A hybrid simulation approach that simulates an actual South Africa production system is introduced in Chapter 3, as are the four main intelligent agents. Each agent is programmed at an entity with unique behaviours and is then allowed to interact within the simulation model. These agents are:

- Healthy operator agent
- Deteriorating operator agent
- New operator agent
- Manager agent

The hybrid model is comprised of three components, each an independent simulation model, that are combined to create a unique method of simulating the effect of employee productivity deterioration on a manufacturing process.

The first component is a Discrete Event simulation model that is called the Distribution Centre Discrete Event (DCDS) model, and it constitutes the foundation component of the hybrid model. Its conceptual development, construction, and testing is explained in Chapter 4.

The second component is called the Operator Agent Behaviour (OAB) model, and is combined with the DCDS to simulate the effects employee deterioration on the system. A key feature of this combination is the communication mechanism used to synchronise the two models together, and this is explained in Chapter 5 along with the OAB's conceptual development and experimentation. Four experiments were conducted using the model, each testing its sensitivity and robustness to the different conditions of employee deterioration.

The third component is based on the Agent Based simulation technique and is called the Manager Agent Interaction (MAI) model. It simulates the system "manager" entity's movement through the system and interaction with its employees. This is used to test the effect this interaction has on the individual employees' productivity, and whether it ultimately benefits the overall system productivity.

In Chapter 6 nine experiments are conducted with the hybrid model which comprised of all three components. The unique technique used to synchronise all three together is discussed, and a conclusion formed as to the hybrid model's effectiveness.

Chapter 7 concludes the results of the experiments conducted in this work by using a financial model to assess the viability of six different scenarios. Each scenario uses a selection of inputs from the fourteen experiments that were completed in Chapters 4, 5, and 6, and a financial comparison is made to assess which state is the most profitable.

Chapter 2

The simulation approach

2.1 Introduction

Over time human beings have used crystal balls, soothsayers, witch doctors, mystics, and sages to give them answers about the future. This is because the human brain uses the limited amount of information that is available to it to make decisions that guide future action. Social Psychologists such as Herbert Simon [28] [29] have defined this behaviour as “bounded rationality”.

The complexity of the modern manufactured and natural worlds (think multi-global enterprises and quantum mechanics respectively) makes it impossible for an individual human being to absorb all the available information and then transform it into a specific scenario for the future. Thus the need to understand complex systems and make predictions about their behaviour has resulted in the emergence of simulation as a tool.

When simulation is considered as a technique to be used in the development of a solution to a particular problem, it is important to consider that it is simply a tool. A typical computer simulation model usually consists of a software program that can contain a number of mathematical equations, or statistical equations, or logical operations. These equations and operations are developed according to rules of behaviour that define how the model will change its state through time [15].

If the dynamics of the real system are not understood, then the model cannot be built with a sufficient level of abstraction. This is important because when models are tested against reality, it is usual to make predictions about the future behaviour of the real system so that decisions or actions can be taken. A misrepresentation of reality could lead to the wrong assumptions being made, possibly leading to catastrophic results. The old adage; “Garbage in, garbage out” applies.

The popularity and flexibility of simulation of industrial and manufacturing systems can be demonstrated by the following samples of research and application:

- Hao and Shen [19] cite the usefulness of simulation when modelling complex manufacturing systems because they are able to test early behaviours in these systems and gain early insights to the mechanics of very complex systems.

- Baesler and Sepulveda [30] have used simulation in combination with the design of experiments (DOE), a technique that is currently popular with six sigma practioners to solve productivity problems in the sawmill industry

- Albino *et al.* [31] have attempted to model innovation at enterprise level through their work in industrial districts

The list of applications is exhaustive.

Simulation is a structured problem solving technique, and thus follows a process. For this work, the approach as defined in Banks *et al.* [12] and Labarthe *et al.* [27] is used, and consists of the following steps (refer to Figure 2.1):

2.1.1. Problem formulation – a problem statement is used to define what problem the simulation aims to solve.

2.1.2. Setting of objectives – a list of objectives detailing what questions are going to be answered by the simulation.

2.1.3. Data collection – key information about the system are collected to enhance model construction.

2.1.4. Model conceptualisation – the essential features of the real system are identified for modelling. Upon these building blocks the rest of the model is constructed allowing for it to be more representative of the real system.

2.1.5. Model translation – real system characteristics are translated into the virtual world with the assistance of relevant simulation software.

2.1.6. Model verification – involves determining whether the computer model is a sufficient copy of the real system. It is a continuous process throughout the construction of the model, and is an important part of model development.

2.1.7. Model validation – is the process whereby the model is tested for accuracy against the real system. It involves statistically testing the outputs of the model against that of the real system, using the same input parameters. If any discrepancies are experienced, then the model can be adjusted to be more representative of the real system. If the validation process is successful, then it can be confidently stated that the model is an accurate representation of the real system.

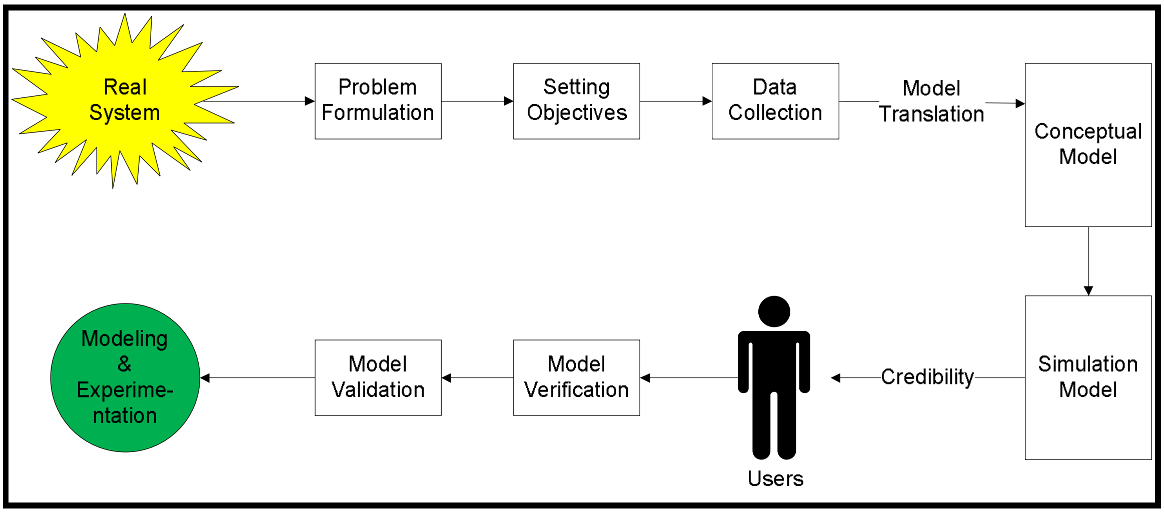

Figure 2.1: This simulation process (1)

Simulation has been used as a decision making and analytical tool in the production, logistics, planning, control, financial, and social sciences environments where complex systems are prevalent.

Hao and Shen [19] for example, have successfully used simulation in material handling environments because the unpredictability of this environment does not allow the problem to be solved through pure analytical or mathematical approaches. This has been substantiated in research by Labarthe *et al.* [27] in their research into the application of simulation to supply chains.

Furthermore, the application of simulation can be used either for descriptive or normative purposes. Descriptive applications involve using simulation to gain a better understanding of the system behaviour and performance, which can be used to improve management's knowledge and encourage better systems thinking from them.

Alternatively, normative applications aim to improve the performance of systems by identifying the optimum conditions within the dynamics and settings of the system. The recent improvement of computer graphics in simulation software has made the "physical picture" of the existing system being modelled, or a proposed system, easier for management to interpret and understand.

Computer simulation development is synonymous with the development of computation processing power, with the first simulations run from 1955 onwards. Since that time there have been huge strides in technology, especially in computer processing power, which has made simulation more accessible to professionals other than computer scientists.

According to the general literature, there are currently three main approaches to simulation, namely: System Dynamics, Discrete Event, and Agent Based.

2.2. System Dynamics (SD)

System Dynamics is the study of dynamic feedback systems using computer simulation usually using systems of ordinary difference or differential equations. The method is best suited to situations where most of the variables change continuously.

This approach has been under development at MIT since 1956, where the concept was pioneered by Jay W. Forrester [9]. It first began as a management discipline to understand how the policies of corporations produce successes and failures. It is used by practitioners as a method to better understand and learn about complex systems such as social systems.

Information-feedback characteristics of system activity are studied to determine how the structure, decision making dynamics, information flow, and social architecture interact to influence the behaviour of the system.

A typical SD model consists of a number of interlinked cause-effect relationships, as exist in real life. Time delays and non-linearity's that are typical of reality are often included in the models. Business researchers have in recent years attempted to model business systems and economies to gain a better understanding of their dynamics and to predict what they might do under various macro economic scenarios.

Social scientists use SD in a similar manner, but their focus is more on the humanities within the study environment. An example of this was when SD was applied to solve the problem of inner-city slums in Chicago, in the U.S.A. An assumption was made by city planners that more low cost housing was necessary to house the influx of workers from the rural areas of Illinois. An SD model was developed and showed that it was this increase in low cost housing that was encouraging workers to migrate.

SD is an appropriate tool when high levels of abstraction are required. An SD model can contain a large number of equations, or systems of equations, when simulating complex systems, and thus under these circumstances is better suited to the simulation of macro level relationships within a complex system.

The system being studied for this work is a complex system, but not on the scale that would require SD to be used. Thus SD is inappropriate for this application, and will not be considered further.

2.3. Discrete Event (DE)

The basic idea behind DE is to simulate the sequential flow of entities through a system of resources and to monitor the collective effective of this flow via critical measures such as resource utilisation and through-put time. Flow charts are often used to provide a “map” of entity flow through a system, which is then converted to a simulation model through a graphical user interface software package such as Arena or Simul8.

DE is usually applied to process modelling, where a number of entities are sent through a collection of resource blocks that process or delay them. It is essentially an entity processing algorithm that can be either deterministic or stochastic depending on the input variables [1].

The DE models used in this work are stochastic because certain parts of the model use statistical distributions as inputs, which will result in the outputs being stochastic as well [8]. The outputs of the models are analysed statistically.

The first process interaction simulation programming language, General Purpose Simulation System (GPSS), was developed at IBM by Geoffrey Gordon in 1961. Over the years, a number of individuals and organisations, including IBM, attempted to simplify simulation modelling through improvements in programming languages and interactivity with the modeller.

The relevantly recent growth in processing power of the Personal Computer (PC) has encouraged the development of software specifically for these machines, thus allowing ever greater access to the technology.

Generally process simulation is used when a lot of information is known about the system being measured. Information such as process times and distributions, entity arrival rate, resource utilisation and downtime, and entity flow have to be well documented in order to create a model using the various software packages available. The flow of the entities is usually deterministic, and can be sufficiently described using a flow chart or value stream map [2].

The usefulness of process simulation models can be misinterpreted by business managers who are not completely familiar with the tool. In the author’s experience, many managers simply see the tool as another way to calculate capacity instead of as an opportunity to experiment and optimise the system variables.

In fact, process simulation can be an excellent decision support tool for manufacturing managers because it allows for detailed analysis of modelled production systems. This allows for a number of different scenarios to be tested without disrupting the real system [8].

One major benefit of simulation is its usefulness in evaluating the effect of subsystems on the performance of a major system. Manufacturing environments are complex systems, which mean that the sum of the parts does not necessarily equal the whole [15]. There is a non-linear relationship between the components of a complex system and its resultant output.

A disadvantage with this type of simulation modelling is that a great deal of data gathering is required before it can be constructed [1]. The collection and interpretation of this data can be time consuming and expensive. The simulations themselves may require the use of expensive computer equipment and personnel, and may take weeks of processing before delivering an analysis [8].

Highly automated manufacturing environments are easier to model because the main variables are usually known, but environments that include humans have variables that are not easily described in a DE model. Humans are thus treated as facilitators of processes, and their individuality is substituted by statistical distributions. Any interaction with fellow humans and the subsequent effects on performance are ignored.

It is this lack of flexibility to model dynamic entities in a system that requires the agent-based simulation approach to be considered.

2.4. Agent Based (AB)

Franklin and Gaesser [16] define an agent thus; “An autonomous agent is a system situated within and a part of an environment that senses that environment and acts on it, over time, in pursuit of its own agenda and so as to effect what it senses in the future.”

Alternatively, Labarthe *et al.* [27] defines an agent thus; “an entity, virtual or physical, capable to act on itself and on the environment in which it evolves and to communicate with other agents. Its behaviour is the consequence of its observations, knowledge and interactions with other agents. An agent can be characterized by its role, its goals, its functionalities, its beliefs, its decisional abilities, its communicational capacities and its learning capabilities.”

An interesting example of how intelligent agents have been used to improve productivity is shown in the airline industry [24]. Up until the late 1980’s a large airline was staffed by 3 operators, namely: a pilot, a co-pilot, and an instrumentation officer. The introduction of intelligent agents to aid the decision making of the crew, by evaluating the on-board systems and then interfacing with crew to provide decision making support, allowed the crew to be reduced to two operators only. This technology is being further developed in Unmanned Aerial Vehicles (UAVs), where ironically there are no pilots.

A few examples of how broadly the agent based approach is applied to problem solving are:

2.4.1. Xiang and Lee [20] demonstrate how Ant Colony Intelligence (ACI), an approach that uses the collective intelligence of highly autonomous distributed systems, as evident in ant and other similar insect colonies, to solve a problem related to dynamic manufacturing scheduling.

2.4.2. Zhang *et al.* [21] have identified that manufacturing organisations face efficiency issues at the process, systems, and enterprise levels. The complexity of the problem at enterprise level, specifically with regards to reconfiguring manufacturing systems to rapidly respond to market demands have led them to use multi-agent simulation to assist management to develop dynamic supply and value chain prediction tools.

2.4.3. Pavon *et al.* [23] use a multi-agent model to analyse emergent behaviour patterns in social systems.

For this work, an agent may be defined as an autonomous entity whose state changes non-deterministically due to its interaction with other agents and the environment. What makes an agent unique is its ability to learn through interacting with the system within which it is found.

The concept of the agent was initially considered in the early fifties [1], with the first agent models being developed at MIT by J. McCarthy and O.G. Selfridge. Schmidt [13] lists the important attributes of agents as being autonomous and intelligent in behaviour, possessing an individual view of the external environment in which it exists, showing the capacity to communicate and cooperate with other intelligent agents in its environment, and being able to move within its spatial world.

Scientists, sociologists, and engineers are increasingly seeking new ways to accurately simulate complex systems, especially those where there is a lack of complete systems data [1]. Agent Based simulation is seen as a new paradigm in the development of complex models to analyse these systems, and consists of simulating a set of discrete objects which change their state over time in a potentially changing environment [13].

Further research has indicated that the traditional centralised models of complex systems are becoming difficult to solve and are insufficiently dynamic to remain relevant. This becomes an issue because the time and cost spent to build such a model is usually high, resulting in researchers such as Anosike and Zhang [22] using multi-agent models in their work with integrated manufacturing operations to overcome this problem.

The benefit of an agent based system is the flexibility in its design, where the results of the model are computed and hence don't need to be solved analytically [26]. This is especially useful for non-linear systems which can become "analytically intractable" [26], or difficult to algebraically manipulate.

Zhang *et al.* [21] have identified the need for an integrated decision platform that can test and evaluate multiple scenarios without needing system designers to restructure the models, as occurs with more traditional simulation tools such as Arena. They have used AB simulation in their work to achieve this.

Sanchez and Lucas [6] document that some of the varied applications of AB simulation are: traffic flow, financial market prediction, and battlefield scenarios.

AB simulation is unique because all the programming is done at an entity level, rather than at the systems level. The emergence of objected-oriented programming technology has facilitated this approach by reducing the coding burden on the developer [33], especially when multiple interactions and behaviours are required at agent level.

Models typically consist of multiple entities (agents) that sense and stochastically respond to conditions in their local environment. These multi-agent systems (MAS) are popular because the independent and flexible natures of the component agents are well suited to the construction of robust and dynamic complex system models.

A multi-agent system (MAS) can be defined as a collection of autonomous entities that interact with each other and the environment. MAS's are typically organised within a framework that defines the relationships between the agents, objectives of the individuals, and resources available to achieve the set goals.

The resulting behaviour of the model is non-deterministic, and often provides a good representation of the complex, large-scale system that it is attempting to simulate.

Remondino [1], and Brailsford and Schmidt [25] discuss two different types of agents namely: reactive, and deliberative.

Reactive agents draw on a finite set of prescribed behaviours, whilst deliberative agents exhibit more intelligent behaviour and are able to analyse a scenario and predict the outcomes of their actions. The agents used in this work are reactive because they are pre-set to change state under specific circumstances in the model, e.g. when a manager agent influences an operator agent's state in a predetermined way through agent interaction in the model.

At the multi-agent model level, three frameworks [22] can be identified: hierarchical, heterarchical, and hybrid.

Hierarchical frameworks are characterised by a master/slave relationship where control decisions are centralised at a high level and then sent down the structure to the lower levels of the system. Individual agents are given different classifications and objectives based on their status in the system.

Alternatively in heterarchical frameworks, the individual agents are given more freedom and thus are more likely to prioritise their local objective than the global one. An analogy to the "free-market" system employed by many western economies could be made here, where the system dynamics at an individual level are controlled by market forces in that environment.

Hybrid frameworks are a combination of the hierarchical and heterarchical frameworks that make use of the self-organising characteristics of the heterarchical, whilst incorporating the structure found in the hierarchical framework. The resulting model is highly flexible and robust, allowing for a more realistic model of a complex system to be constructed.

2.5. Deciding on the simulation approach

The industrial engineering community, of which the author is a member, has made extensive use of DE simulation, with the annual Winter Simulation Conferences in the U.S.A. being well attended by industrial engineers.

The emergence of new software programmes as well as the improvement of existing simulation methodologies and software, has left the industrial engineering simulation analyst spoilt for choice [8]. These packages include: VenSim, Arena, ProModel, FlexSim, Swarm, SimWalk, RePast, and Anylogic.

AB simulation by contrast has until recently been a purely academic topic. However, increasing demand by global business to optimise operations and supply chain, have created a need for some alternate tools [1].

It is in this environment that AB has emerged from universities and computer science laboratories to become a more accessible tool to the industrial engineer. Programs such as “Netlogo”, developed by North-Western University in the USA, are available for free on the internet (http://ccl.northwestern.edu/netlogo/) and can be used to model scenarios that would also be done with more conventional DE simulation techniques, such as the flow of traffic (as seen below in Fig 2.4.1)

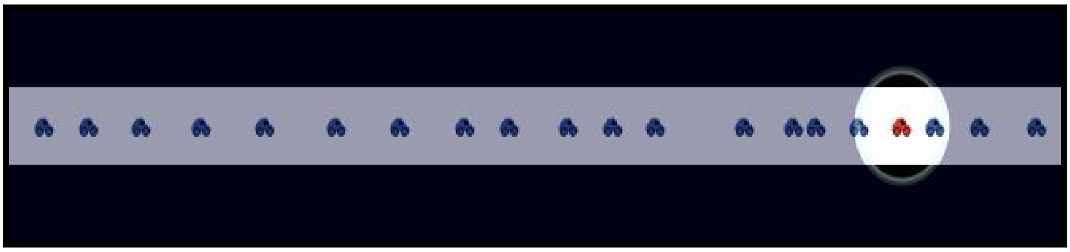

Figure 2.2: Screen shot of a Netlogo simulation example 1

As discussed in this chapter, the strength of AB simulation is its simplicity. Individual agents can be programmed with simple behavioural logic commands, yet the interaction of a significant number of the agents can mimic a very complex system.

The problem to be solved in this work has a foundation in DE simulation, but requires the flexibility of entity based programming that AB simulation enables to create a better representation of reality. Thus a hybrid model using both DE and AB simulation will be used, with the purpose of creating a representation of reality that can be experimented with to understand its dynamics.

Chapter 3

The Conceptual Hybrid Model

3.1. Introduction to the Hybrid Model

Hybrid simulation is generally deemed to be a combination of common simulation techniques that are used to develop models for highly complex systems such as multi-enterprise entities.

Remondino [1] and Hao and Shen [19] in their respective work in enterprise dynamics and manufacturing, have cited the benefits of using the aggregate behaviour of simply programmed agents in their work, as opposed to the highly detailed systems components that are used in regular process simulation. However, agent models become a more realistic representation of the system being modelled when combined with a structured framework.

The use of the hybrid paradigm to solve complex problems, specifically in the production, logistics, and planning environments [19] [21] [27] is gaining momentum, as is evident in the literature. Agent based technology is proving to be a promising paradigm for the design of next generation manufacturing and logistics systems, and systems engineers are seeking innovative approaches to model ever more complex value streams.

Traditional simulation tools are limited when simulating large-scale, complex, and highly dynamic heterogeneous systems [19]. Until recently, no one platform existed that combined discrete event and agent based simulation. This changed when AnyLogic was first presented at the 2000 Winter Simulation Conference.

AnyLogic, which has been developed by XJ Technologies, has recently emerged as the simulation tool of choice with practioners who are experimenting with the hybrid approach. It combines system dynamics, discrete event, and agent based on one platform, but requires programming knowledge to configure.

An objective of this work is to create a modelling technique that is simple and utilises commonly available software (ideally making use of Microsoft Office tools). This is because most industrial engineering and operations management personnel who could apply the knowledge gained in this research will not be experts in the development of complicated computer models.

Thus, software such as AnyLogic, which requires advanced programming knowledge, is deemed to be unsuitable and will not be used for model development. Instead an alternate modelling technique is developed.

3.2. Agent Based Communication and Bidding

In Chapter 2, reference was made to the different types of agent frameworks that can be employed. A key element of Agent Based simulation is the communication between agents within these frameworks. It is this interaction at the agent / entity level which is representative of what occurs in complex systems such as a multi-level manufacturing value streams that allows for dynamic models to be constructed.

A review of the literature reveals that a common method of communication used by multi-agents is “bidding” [22] [32]. This is used specifically in the manufacturing, logistics, and planning examples where optimisation of a system is desirable.

For example when optimising value streams at an enterprise level, a number of competing manufacturing multi-agent systems can “bid” for work from the central controlling system using a virtual currency that is linked to desired system performance (cost, quality, or delivery) (Refer to Figure 3.1). Multiple permutations of value flow can then be tested, with the optimum solution easily identified from the resulting experimentation.

This “bidding” mechanism is important to this work because it creates a relevant conduit for agents within the modelled system to communicate with each other on the same hierarchical level, and with the manager agent on a higher level. This is explored in more detail in Chapter 6.

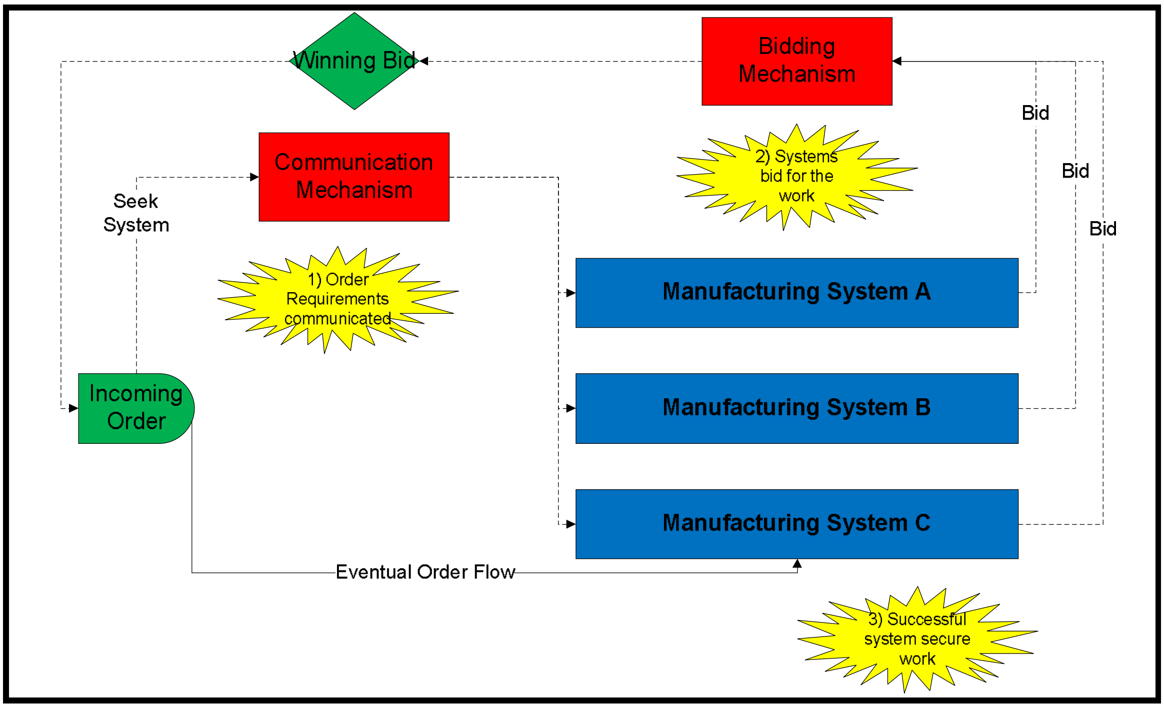

Figure 3.1: The Bidding Process

3.3. The Agent Based Process Simulation Approach

In a typical manufacturing environment, subsystems interact with the common aim of producing a desirable output, for example where a human interacts with a machine to manufacture a product that is to be sold. The human could be a machine operator, and the machine could be a press that manufactures a metal component. Each contributes to the output of the system in a unique way, and it would be impossible to produce the output without one or the other.

The machine's variables are generally well known and controllable, whilst the human's interaction with the machine are minimal, usually limited to inserting and removing the blank piece of metal to be pressed. Under these circumstances and given enough time, it is possible to collect sufficient data to be able to build an accurate Discrete Event simulation to model this particular interaction and make a confident prediction on its outputs [2].

Humans are usually more complex subsystems than machines, and thus the larger the role that they play in the interaction, the less deterministic the outputs, and thus the more difficult it is to build an accurate prediction model. Brailsford and Schmidt [25] have conducted some interesting research where they have attempted to model human behaviour within health care systems using a PECS (physical, emotional, cognitive, and social) model. The PECS model takes the view that human beings are "psychosomatic units with cognitive facilities embedded in a social environment", and gives evidence of the type of thinking that is being done to properly model humans within the simulation environment.

In the manufacturing environment that forms the basis of this work, humans interact with objects, machines, and other humans to produce system outputs. Under conditions of stasis, the interaction between humans and objects, and humans and machines is reasonably consistent, and can be sufficiently described by a statistical distribution.

However, the reality is that humans are variable beings that are sensitive to external inputs that affect their performance. Thus a pure DE simulation can only be developed for the stable parts of the process, but not for those components that are susceptible to human inconsistency. The simulation model must be sufficiently flexible to allow for these external inputs from the environment to be represented and adjusted.

A particular external influence that would result in process variability would be when a human interacts with another human, for example when a manager interacts with an operator. This interaction can be completely random in both the time of day that it occurs and the duration of the interaction. The difficulty in collecting sufficient process information to fit the process to a statistical distribution negates the use of DE simulation to model this process.

As discussed in Chapter 2, Agent Based simulation involves the programming of individual entities that interact within a specific environment. Each agent can be “told” how to react when contact is made with another agent, thus allowing for the autonomous behaviour that is found in human society. If the basic behaviours of humans in a specific manufacturing environment are understood, then an Agent Based model can be developed to model these behaviours, and their subsequent effects on the process.

In a manufacturing environment that contains a significant number of both humans and machines, such as the one used as the basis for this work, it is important to be able to develop a model that can simulate multiple interactions between humans, objects, and machines. This requires the development of a unique hybrid simulation technique that combines both Agent Based simulation and Discrete Event Process simulation.

Remondino [1] introduces a technique, called Agent Based Process Simulation that combines AB simulation and DE simulation to model a complex system such as an enterprise. This approach can be applied on a smaller, but no less complex basis, and it is the aim of this dissertation to demonstrate how it can be applied to a typical manufacturing environment, and in doing so answer the questions posed in the introduction.

3.4. The modelling challenge and approach

Anylogic, and other similarly orientated software that requires advanced programming skills are not applicable in this work. An alternate approach is to break the system into component sub-systems and investigate which commonly available software package are appropriate for simulating each.

Each sub-system model needs to complete simulation runs independent of the other sub-systems, but the overall process must be able to re-integrate these results to create the complete system. The results of each model’s run would then be consolidated into a single output report from which analysis can occur.

Zhang *et al.* [21] used a similar modular approach, which involved the disintegration of a complex system into sub-systems and the development of models for each which were then combined. In that work, as in this, each inter-system interaction was categorised as either deterministic or non-deterministic.

Deterministic interactions can usually be measured and described with a standard statistical distribution such as the normal or uniform distribution. These sub-system interactions can thus be modelled with a DE simulation software program. Alternatively, non-deterministic interactions can require an approach that uses AB simulation techniques and software.

The development of a working simulation model that satisfies the requirements explained in this text is the main outcome of this work. The conceptual model is explained in the following paragraphs, but it is important to note that the method used to interface the separate simulation models is unique to this work.

3.5. The conceptual hybrid model

The explanation of how the conceptual model works must begin with its smallest constituent the “activity”. An activity is a unit of work that moves through the system, and it is used in the productivity calculation that will be used to compare the different models to each other.

As mentioned in the introduction chapter, the model being developed is based on a real operational system (refer to Picture 3.1 below). It is an auto part warehousing and cross docking operation based in Durban, South Africa, and is manned by 20 operators who are supervised by a single manager.

It is a linear system through which activities flow, and is broken into 4 distinctive sections:

3.5.1. A receiving holding area
3.5.2. A receiving zone
3.5.3. A circular conveyor
3.5.4. A dispatch zone

Picture 3.1: The warehousing and cross docking operation

A summary of how activities flow through the system is shown in Figure 3.2 (refer to Chapter 4 for a detailed diagram of the activity flow). It is colour coded to display the following:

- White – displays where the activities enter and exit the system, and are thus measured for system productivity purposes
- Green – indicates where humans interact with the activities
- Yellow – indicates where the activities are queued on the conveyor

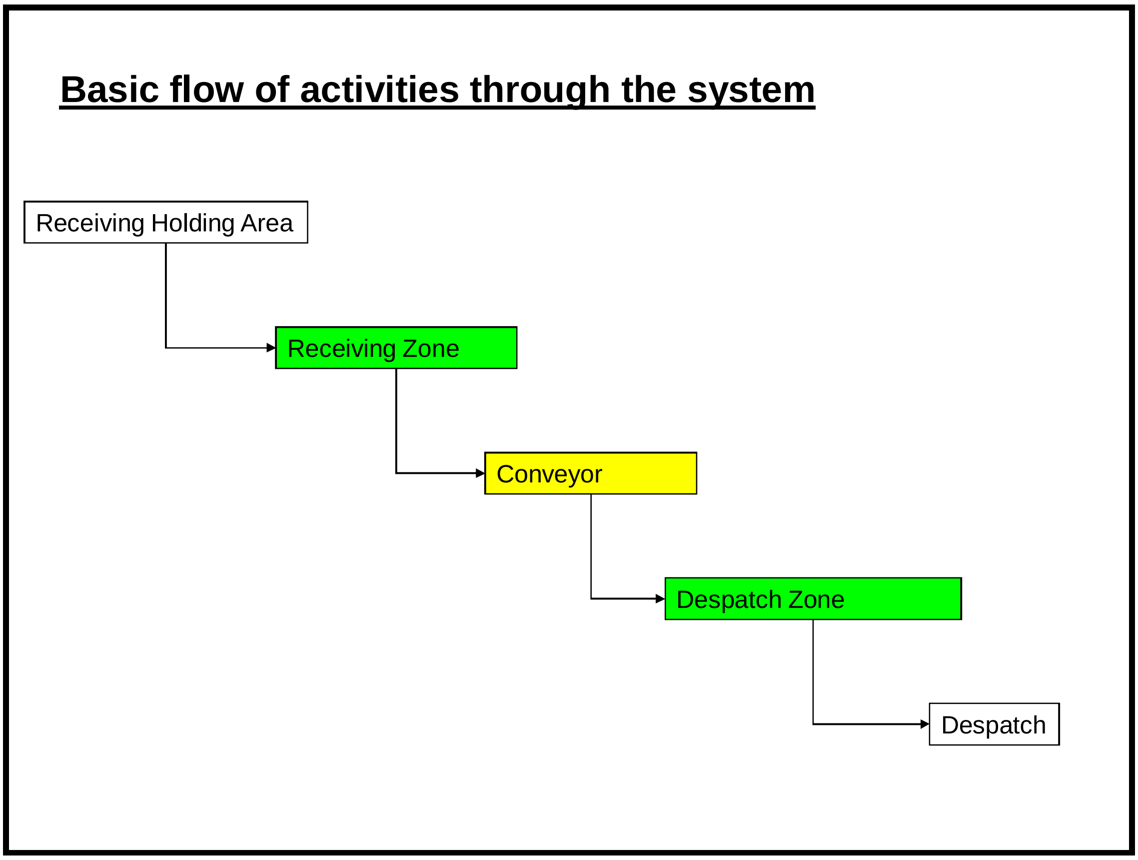

Fig 3.2: Basic flow of activities through the system

Eleven operators are present in the receiving process zone, and nine in the despatch processing zone. Receiving processing zone operators interact with each other and the manager, but not with the despatch zone operators, who have a similar relationship with their co-workers and the manager.

In the conceptual model each operator becomes an “operator agent”, whilst the manager becomes the “manager agent”. The agent’s position relative to the system layout and flow of activities can be seen in Figure 3.3. The red lines indicate the interaction between agents.

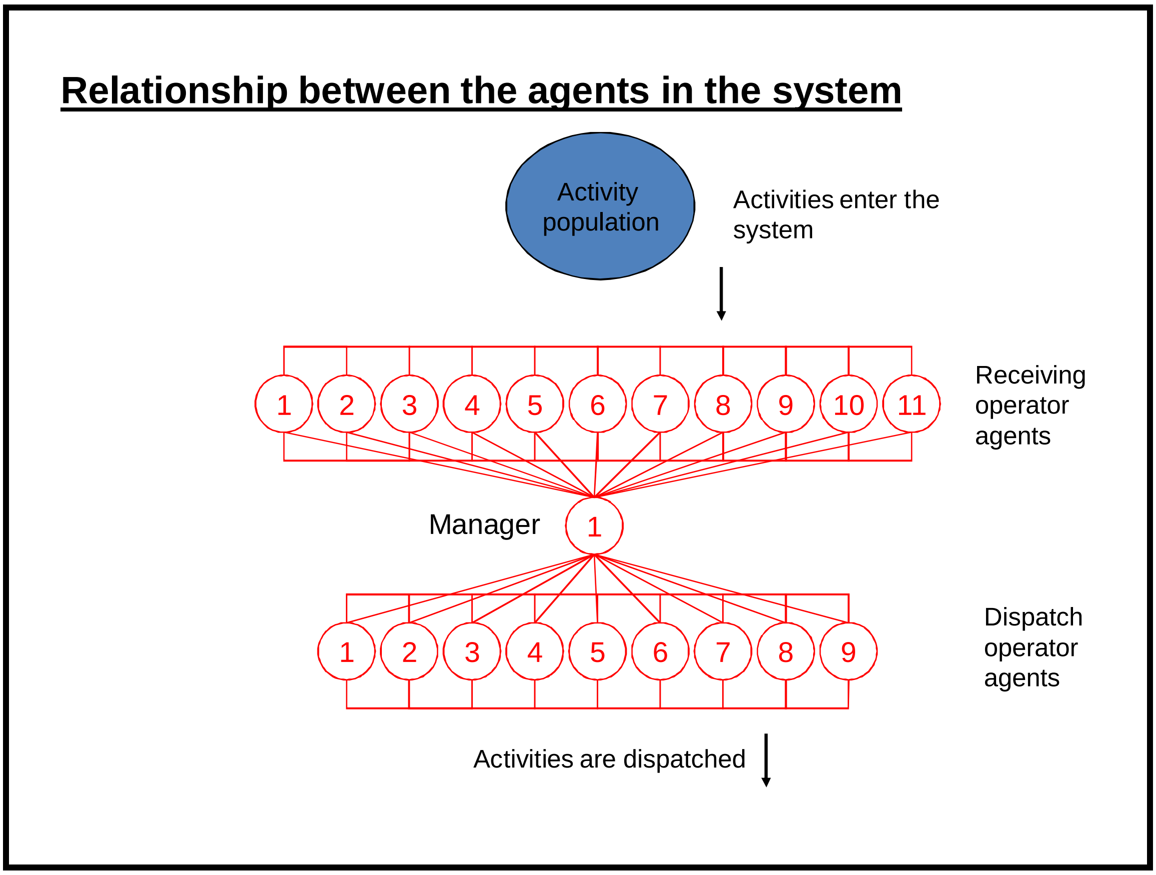

Figure 3.3: Activity flow relevant to agent placement

Within the modelling environment, each agent can be defined as one of following four entities:

- Manager agents (MA)
- Healthy operator agents (HA)
- Deteriorating operator agents (DA)
- New operator agents (NA)

Each agent has unique properties that influence the outcome of the model, and are based in reality.

Healthy operator agents (HA) work at an acceptable rate and are the benchmarks for performance in the system.

Deteriorating operator agents (DA) work at a decreasing rate until they are so inefficient that they need to be replaced by a new operator agent. This point is reached when their efficiency is half that of a healthy operator agent. The rate of decrease can be specified as one of the model parameters. DA's experience a temporary increase in productivity when they interact with a manager agent, but this improvement will lapse once the MA moves away.

New operator agents (NA) work at an increasing rate that is reliant on interaction with activities (i.e. the more work they do, the better they become). The NA's experience an improvement in productivity when they interact with a manager agent which is due to the mentoring effect that occurs during this interaction. Over time a NA will evolve into a HA and will exhibit the same productivity standards.

Manager agents (MA) raise the productivity of deteriorating operator agents through interaction with them because they either motivate them to work more productively or physically assist them to get the work done. Manager agents also improve the learning rate of new operator agents through mentoring them.

3.6. The hybrid model components

The real system can be represented by a simulation model that is broken into three components:

3.6.1. Component 1: The Distribution Center Discrete Simulation (DCDS) model

The Distribution Center Discrete Simulation (DCDS) model is a model of the linear flow of activities through the system. It is deterministic in nature and is simulated with a discrete event model that forms the foundation upon which the other components are built and referenced. It is the primary interface for the integration of the other two models, which enables experimentation to occur on the hybrid model.

Figure 3.4 shows a simple flow chart of the system, and the boundaries of the discrete event model. Note that the same colour coding as used in Figure 3.3 has been used to indicate human interaction with the activities flowing through the system. The model was constructed using an off-the-shelf software programme called Simul8, the mechanics of which are taught at most industrial engineering and operations management faculties.

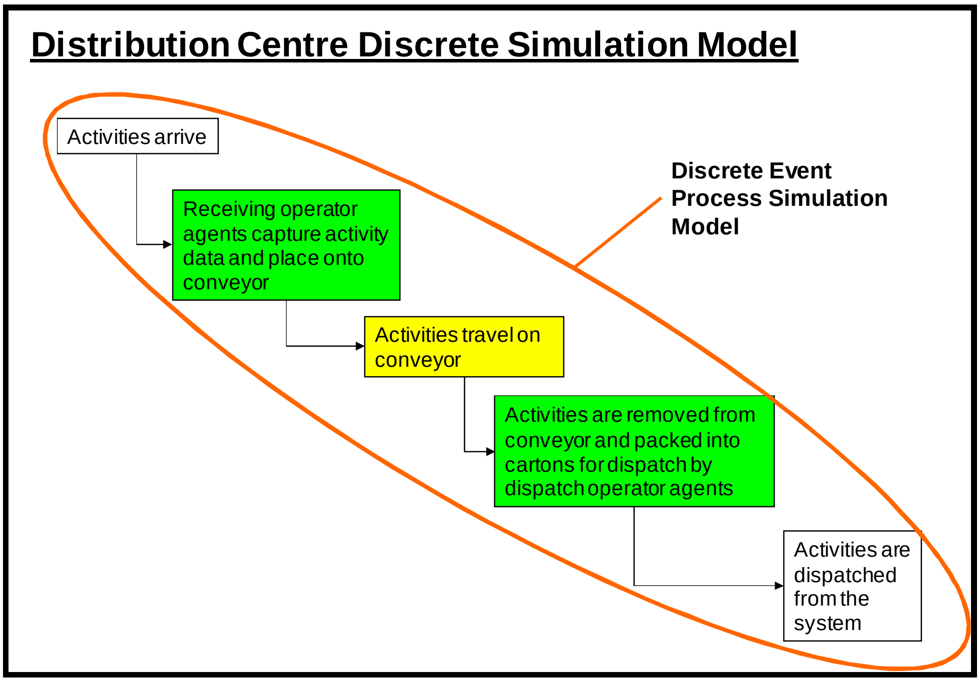

Figure 3.4: The DCDS model concept

3.6.2. Component 2: The Operator Agent Behaviour (OAB) model

The Operator Agent Behaviour (OAB) model is a spread sheet based model that creates an input to the discrete event model. Its mechanics are discussed in Chapter 5, and it simulates the reducing productivity of the deteriorating operator agents or increasing productivity of the new operator agents. It is deterministic in its singular form, but becomes complex when combined with the discrete event model. Figure 3.5 displays where the two models integrate with each other.

The conceptual logic of this model is that for simulation purposes an agent is classified as either in a state of deterioration or improvement (learning) over time. The agents' productivity, which is a measure of the number of activities it can complete within a specific time period, will decrease or increase depending on its state.

The model makes use of a deteriorating or learning "factor" which is applied within the spread sheet environment, and its output forms an input to the discrete event simulation model.

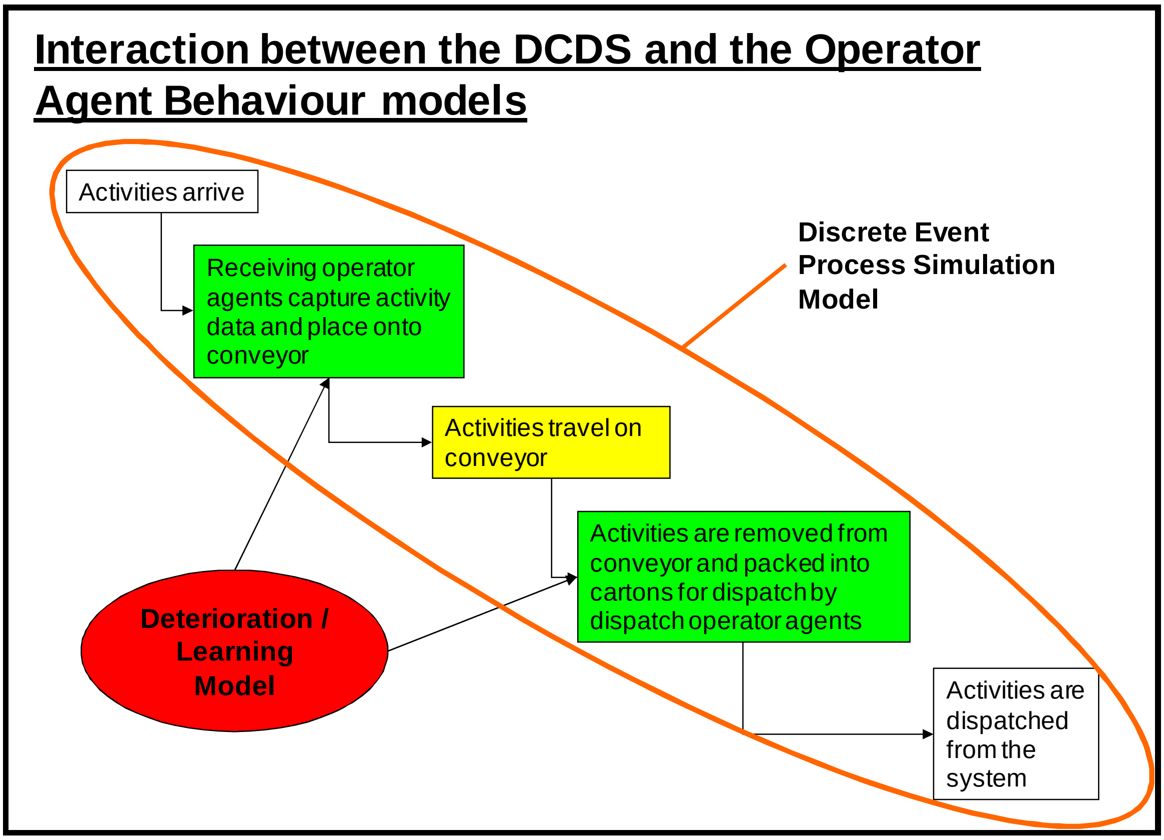

Figure 3.5: Interaction between the DCDS and OAB models

3.6.3. Component 3: The Manager Agent Interaction (MAI) model

The Manager Agent Interaction (MAI) model is an agent based model which simulates the manager agent's interaction with the operator agents. Observation of the real system indicated that the manager's daily tasks can be classified into categories, and the resultant behaviour modelled. The specific breakdown of a manager agent's day into these tasks is documented in Chapter 6 along with the emergent behaviour that occurs from its interaction with the operator agents.

The simulation logic dictates the manager agent's tasks in the modelling environment as well as which operator agents the MA interacts with. These interactions are controlled by the model, and are non-deterministic in nature. This creates an input to deteriorating / learning model, and affects its input to the discrete event model. Figure 3.6 provides an overview of how the three components integrate together to form the single simulation model used for experimentation.

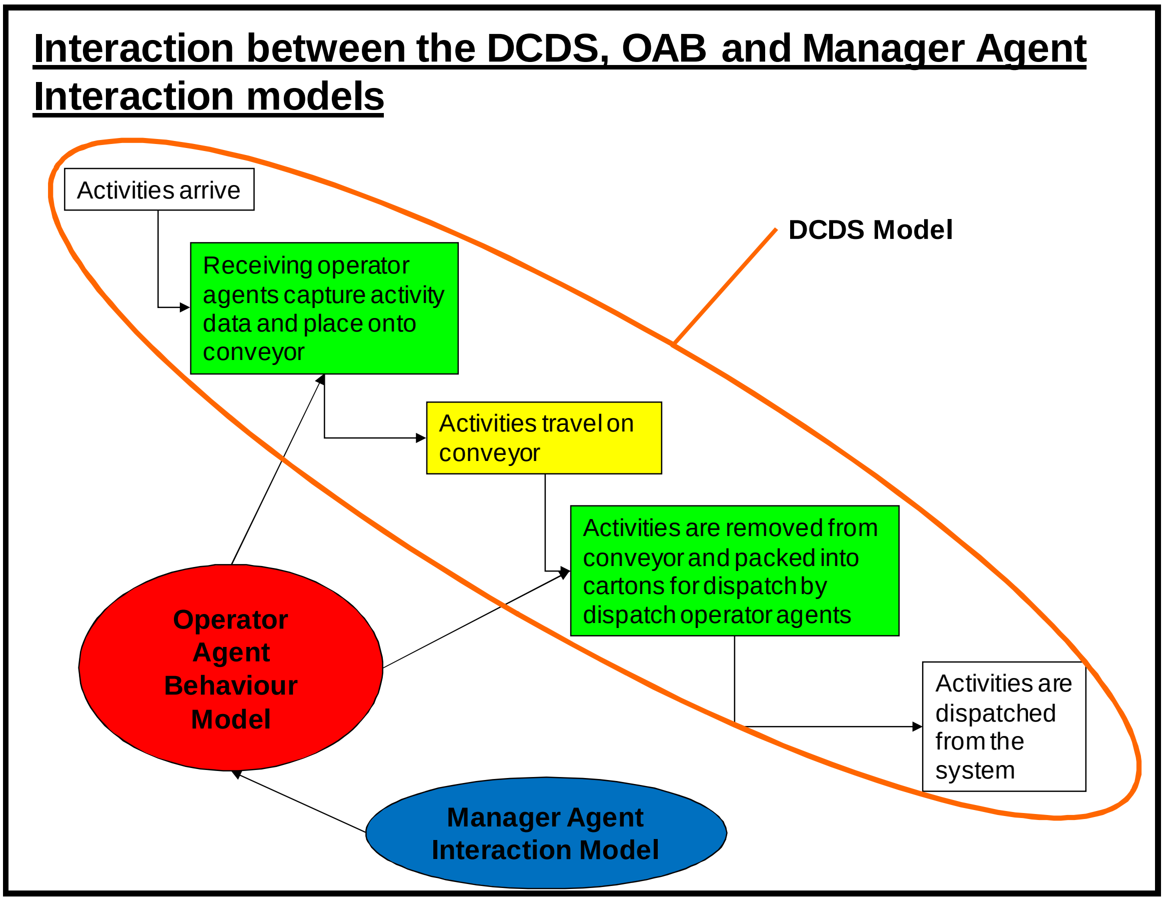

Figure 3.6: The hybrid simulation model components

3.7. The transformation from concept to simulation

The next three chapters provide a detailed description of how the conceptual model was transformed into a simulation model. Each chapter covers one of the models, and explains how the integration mechanism operates.

Chapter 4

The Discrete Event Model

4.1. Introduction

A discrete event simulation model was developed as the basis for the entire model system. It is an important component within the final hybrid model construction because it is the foundation and provides the platform with which the deteriorating / learning model and the Manager Agent model are integrated. This discrete event simulation model is the most conventional of the three approaches used.

As discussed in chapter 3, an off-the-shelf process simulation package was used to program the logic and develop the model structure.

4.2. Description of the system

The process under consideration is a cross-docking operation in a distribution centre that receives and routes motor vehicle spare parts to a network of dealerships around South Africa. The unit of measure for the system is activities, which are the units of work that flow through the system. An activity could be a single part (e.g. an oil filter), or a collection of parts with the same part number (e.g. bolts or nuts). A basic process flow chart (Figure 4.1) documents the flow of activities through the system.

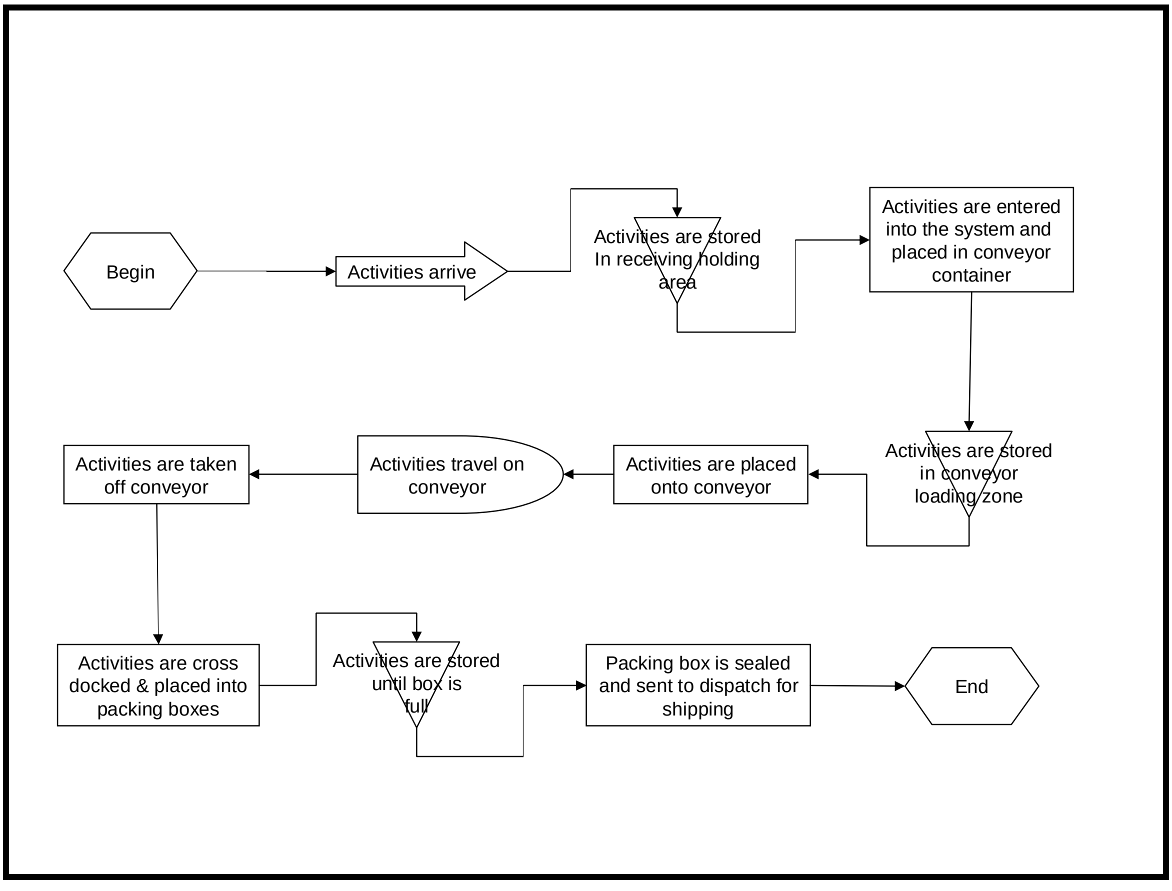

Figure 4.1 Process Flow Chart of Simulated Area

The system requires 20 operators, who are supervised by a single manager. The operators do not require any special skills, other than those learnt on-the-job, to perform the tasks necessary in the system. It was generally agreed by the facility's management that three days of on the job training is sufficient for a new operator to become fully skilled.

4.3. Model conceptualisation

The deterministic nature of the activity flow through the system facilitates the linear construction of the model. Analysis of the real system indicated that there were four main modules through which activities pass (refer to Figure 3.2), these were:

- A receiving holding area where activities are held until an operator is available to work on them
- A receiving zone, where activities are logged into the stock control system and are processed
- A conveyor, which transports activities from the receiving zone to the dispatch zone
- A despatch zone, where activities are logged out of the stock control system, and packed into boxes in preparation for shipping.

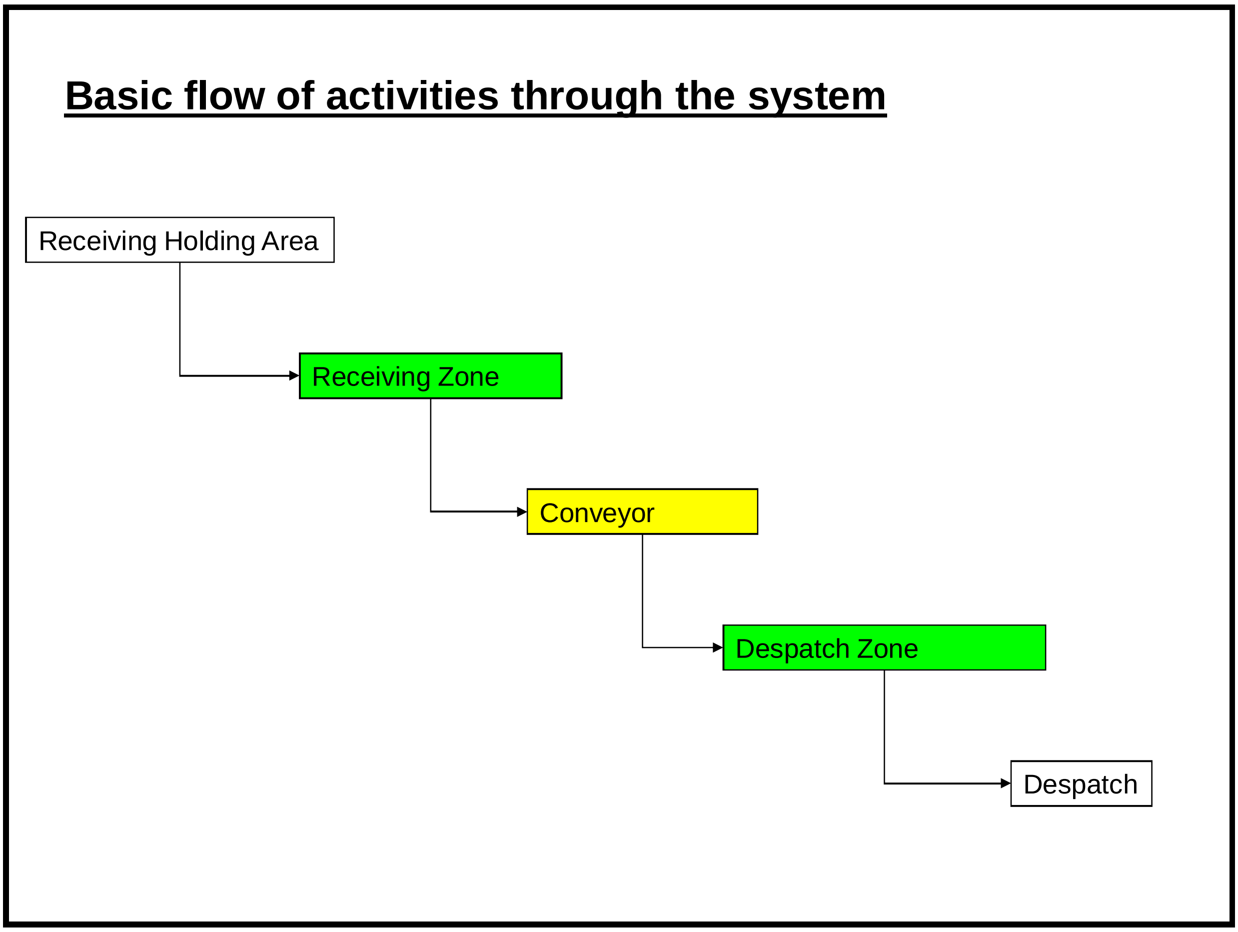

Figure 4.2: Basic flow of activities through the system

The process flow map (Figure 4.1) was used to identify measurement points within the real system, from which information about the system was collected. The measurement points were:

- Activity arrival rate at the receiving dock
- Activity storage capacity in the receiving dock
- Activity transport distance from receiving dock to receiving zone
- Activity storage capacity in the receiving zone
- Cycle time to process activities into the system
- Activity storage capacity in the conveyor storage zone
- Cycle time to load activity onto conveyor
- Conveyor speed
- Cycle time to unload activity from conveyor to despatch zone
- Cycle time to process and prepare activities for dispatch

All activity volume data was analysed, as were the process times collected from the relevant activity processing sections, and the results used to create a basic quantitative picture of the real system. Figure 4.3 below shows the basic conceptual discrete event model.

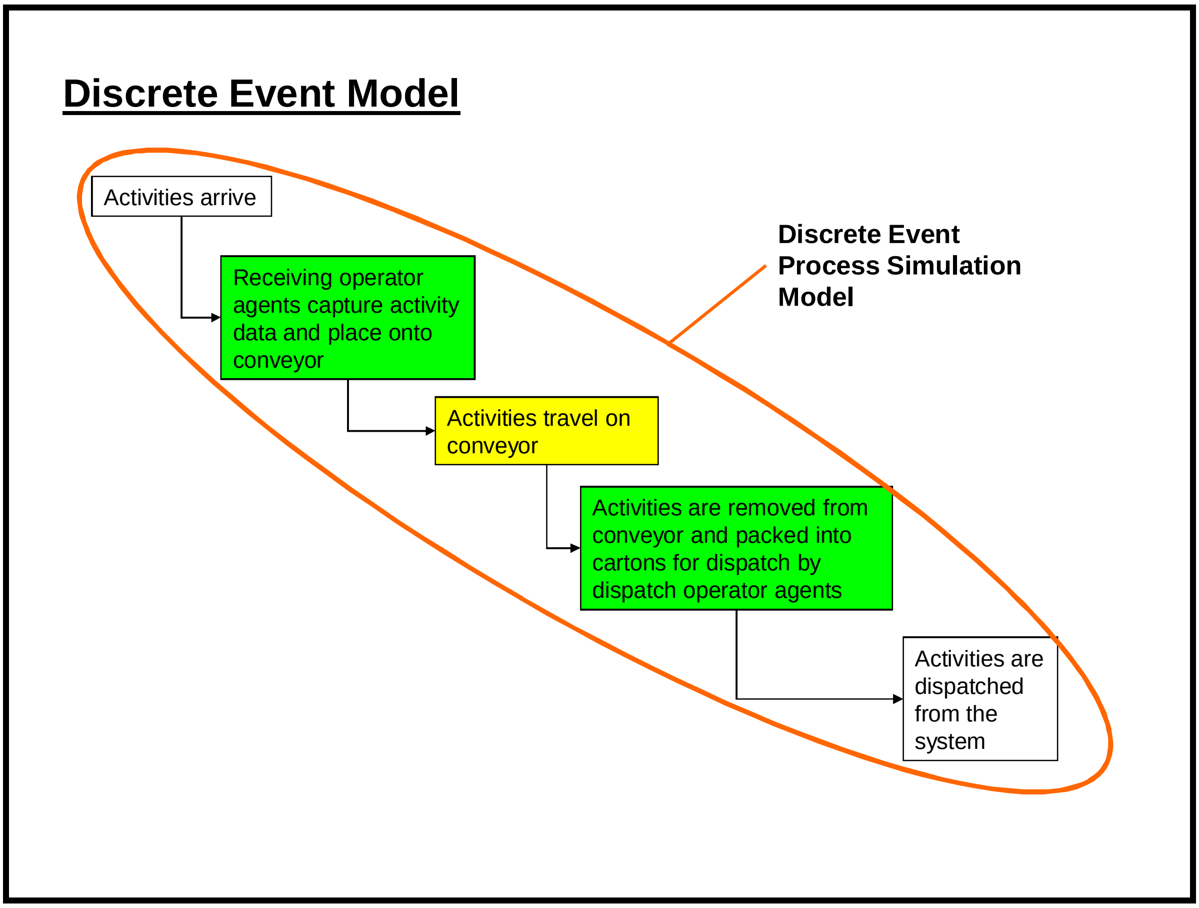

Figure 4.3: The discrete event simulation model concept

4.4. Model development

The prototype model was originally done in Arena (Figure 4.4). Arena was initially chosen because the author had prior experience with it and demonstration software was easy to access.

It was developed using standard discrete event modelling techniques [3] whereby entities that have been programmed with specific attributes, enter a simulation environment and experience the following:

- Queuing
- Being captured by a work centre / resource
- Get released to another queue or work centre / resource

The prototype was developed to be run over a short simulation period, and a production day of 445 minutes was chosen. The Arena model was able to portray a simple version of the system, and was used to gain an understanding of the dynamics of the process. The model was shown to the manager of the system, who agreed that the basic logic and flow was correct. Figure 4.4 shows a screen shot of the arena model

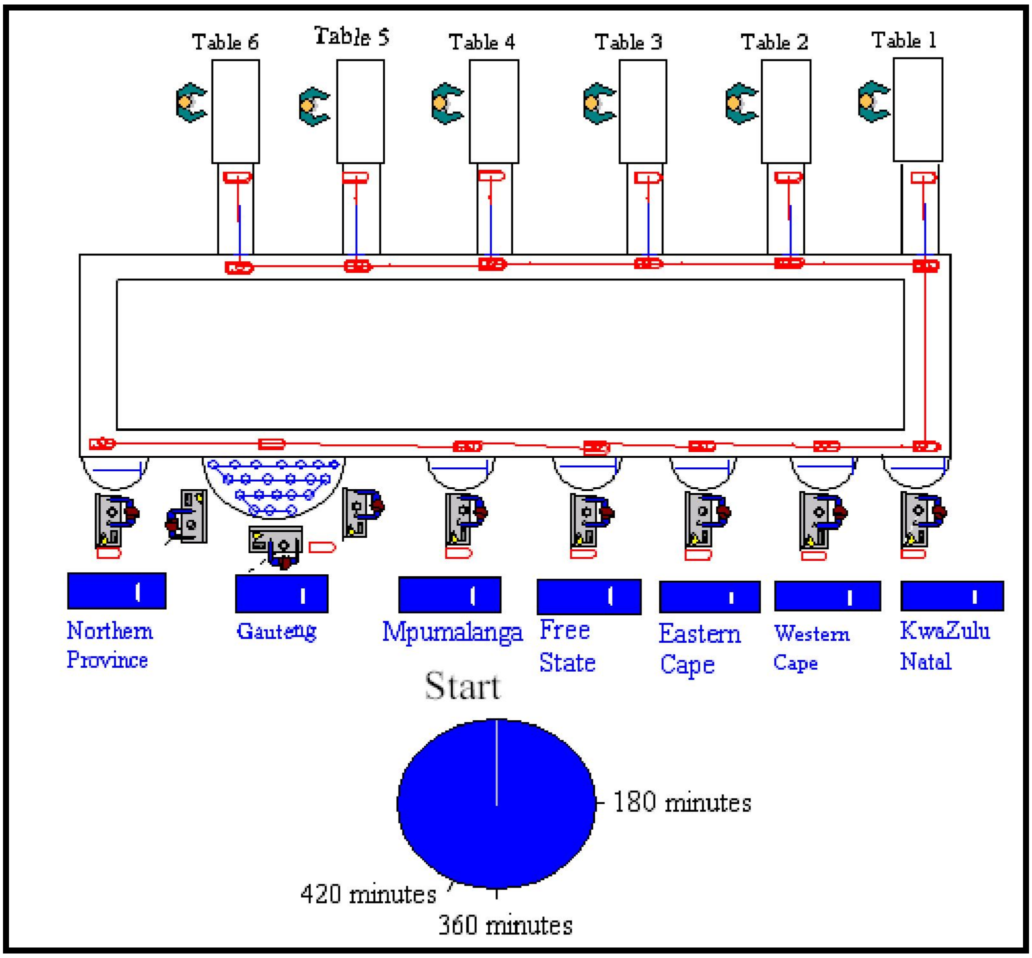

Figure 4.4 Screen Shot of Arena Model

Arena is an excellent simulation package, but is expensive to purchase, and the demonstration version that was used for the initial model did not have sufficient capacity for further model development. Thus it was necessary to identify another software package that could meet the technical requirements of the model without exceeding the established budget. Simul8 from the Simul8 Corporation was identified as the appropriate package to use.

The Simul8 model was developed around the four main modules, with each having their own rules and logic. Once each of these individual components had been constructed they were connected together in the Simul8 model space. The first module modelled the receiving dock, followed by the receiving zone module. The conveyor and despatch zone were the last to be constructed. The completed model will here forth be referred to as the Distribution Centre Discrete Simulation (DCDS) model.

4.5. The DCDS model components

4.5.1. The receiving dock

This model area's primary function is to allow activities to enter the system in a controlled fashion. Once the activities have entered the system, they are temporarily stored, and then equally distributed amongst the 11 receiving zones.

All activities entering the area are given a label by the resource that captured them, and determines which receiving zone receives the particular activity. Figure 4.5 shows a screenshot of the receiving dock from the Simul8 model.

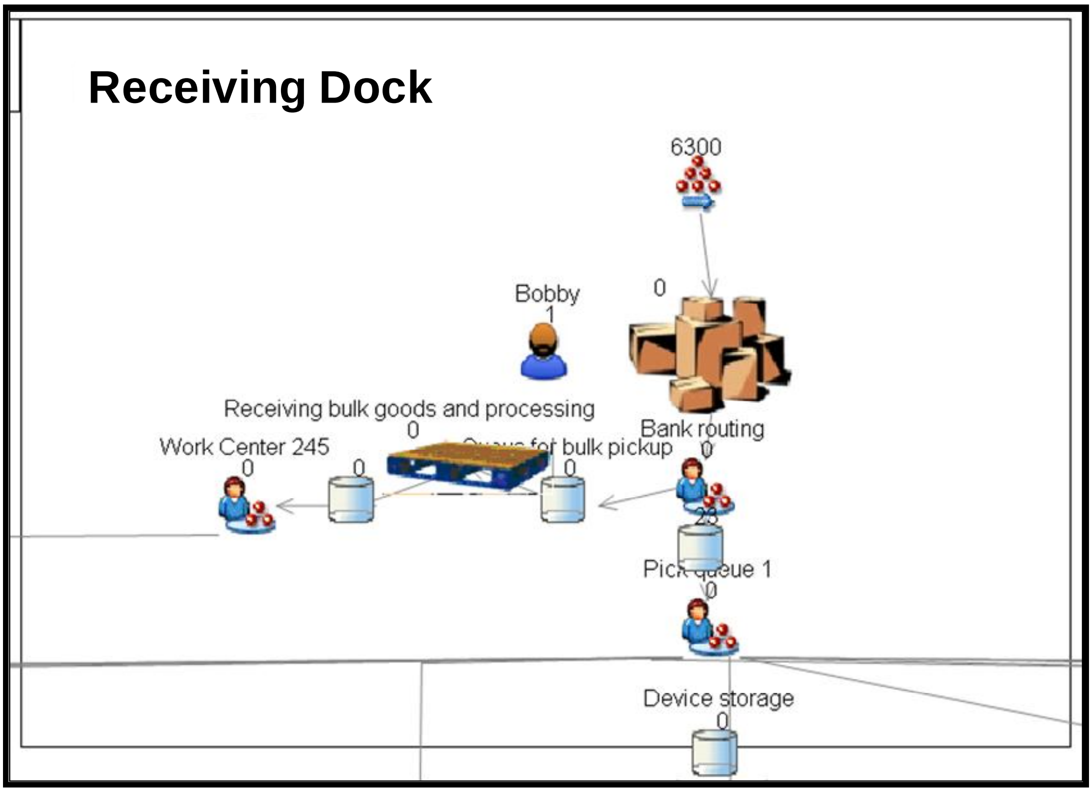

Figure 4.5: Screen shot of the Simul8 model receiving dock

4.5.2. The receiving zone

Once an activity has arrived at this module, it is placed in a queue until a resource is available. The activity is then captured by the resource and held there for a length of time according to the input standard. It is then merged with a container and released to the conveyor.

Another data label is given to each activity here, and this determines which despatch zone is to receive the activity. This is important because it gives the activity a pathway to exit the conveyor, and thus move onto the next module.

Figure 4.6 shows a screen shot of a receiving table, which is one of six subsections that make up the entire receiving zone.

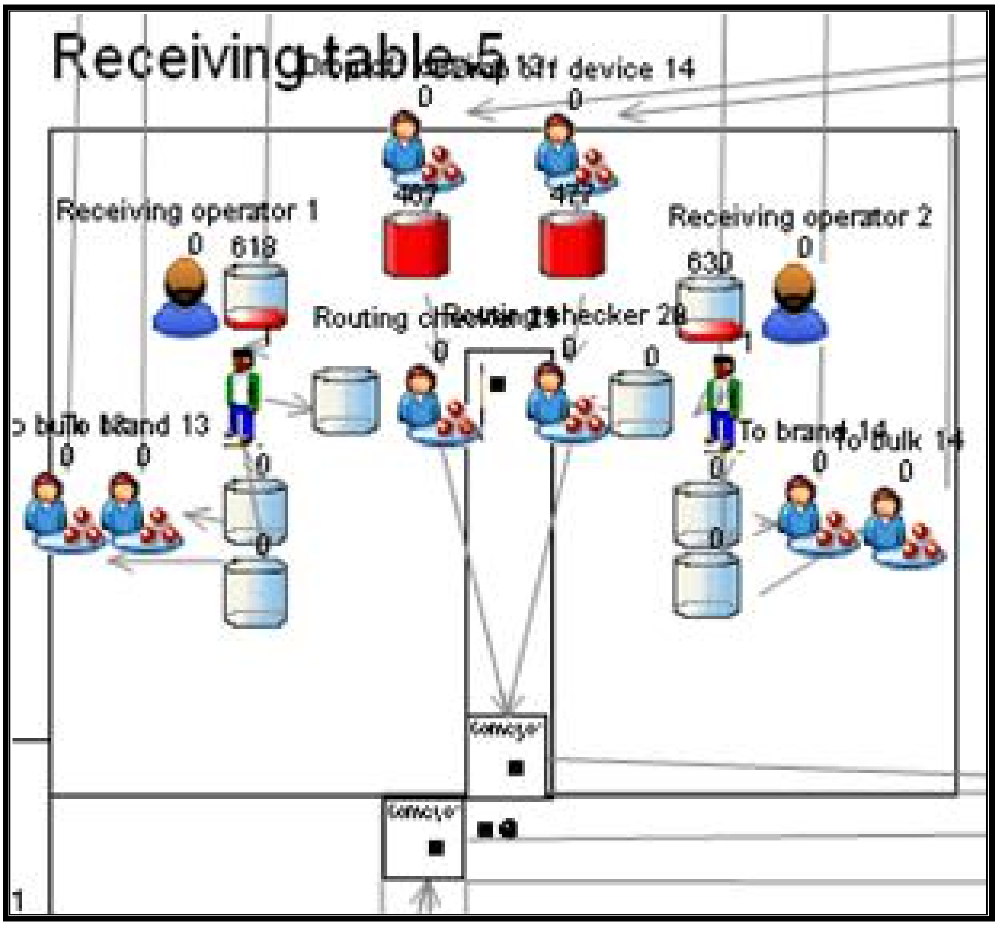

Figure 4.6: Screen shot of one the Simul8 model receiving Zones, showing the entities used to construct the logic

4.5.3. The conveyor

The conveyor consisted of thirteen segments that make up a closed loop (refer to Figure 4.7 for an example). Activities are transported on this until captured by an available and designated despatch zone work centre.

If an activity reaches its destination despatch zone, but is not able to exit the conveyor because the zone is unavailable, then it is immediately placed back onto the conveyor and re-circulated.

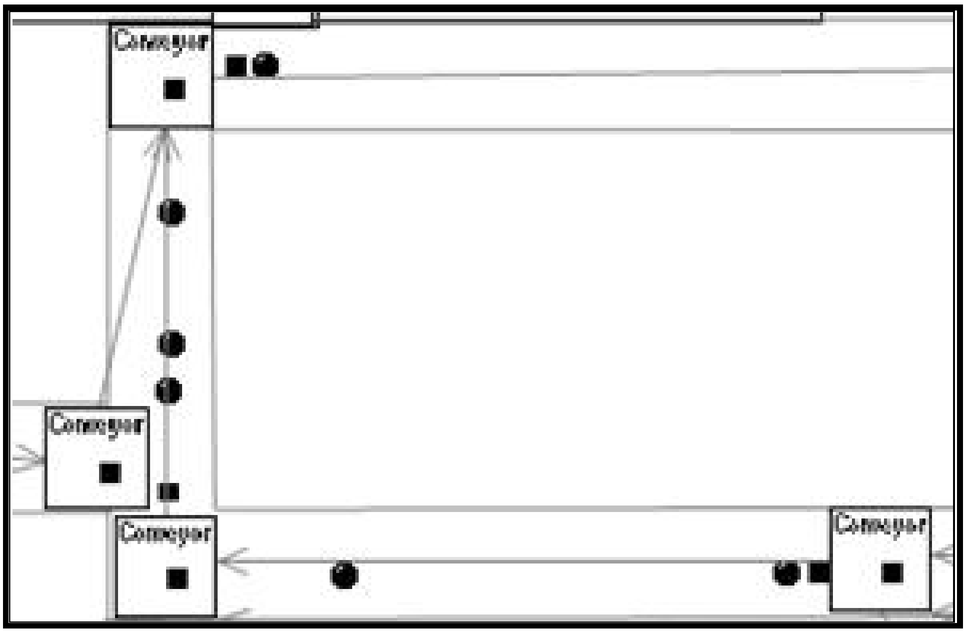

Figure 4.7: Screen shot of four of the thirteen Simul8 model conveyor sections

4.5.4. The despatch zone

This is the most complex module in the model because most of the sorting and processing of the activities are done here. Activities are removed from the conveyor and then processed by the relevant resource in the despatch zone. Activities are then separated from their containers and sent to one of the boxing stations for packing and preparation.

This is the main output measurement point in the model. It is at this point that the through-put of activities coming out of the model can be measured and compared.

Figure 4.8 shows a screen shot of one of the five sub sections that make up the despatch zone.

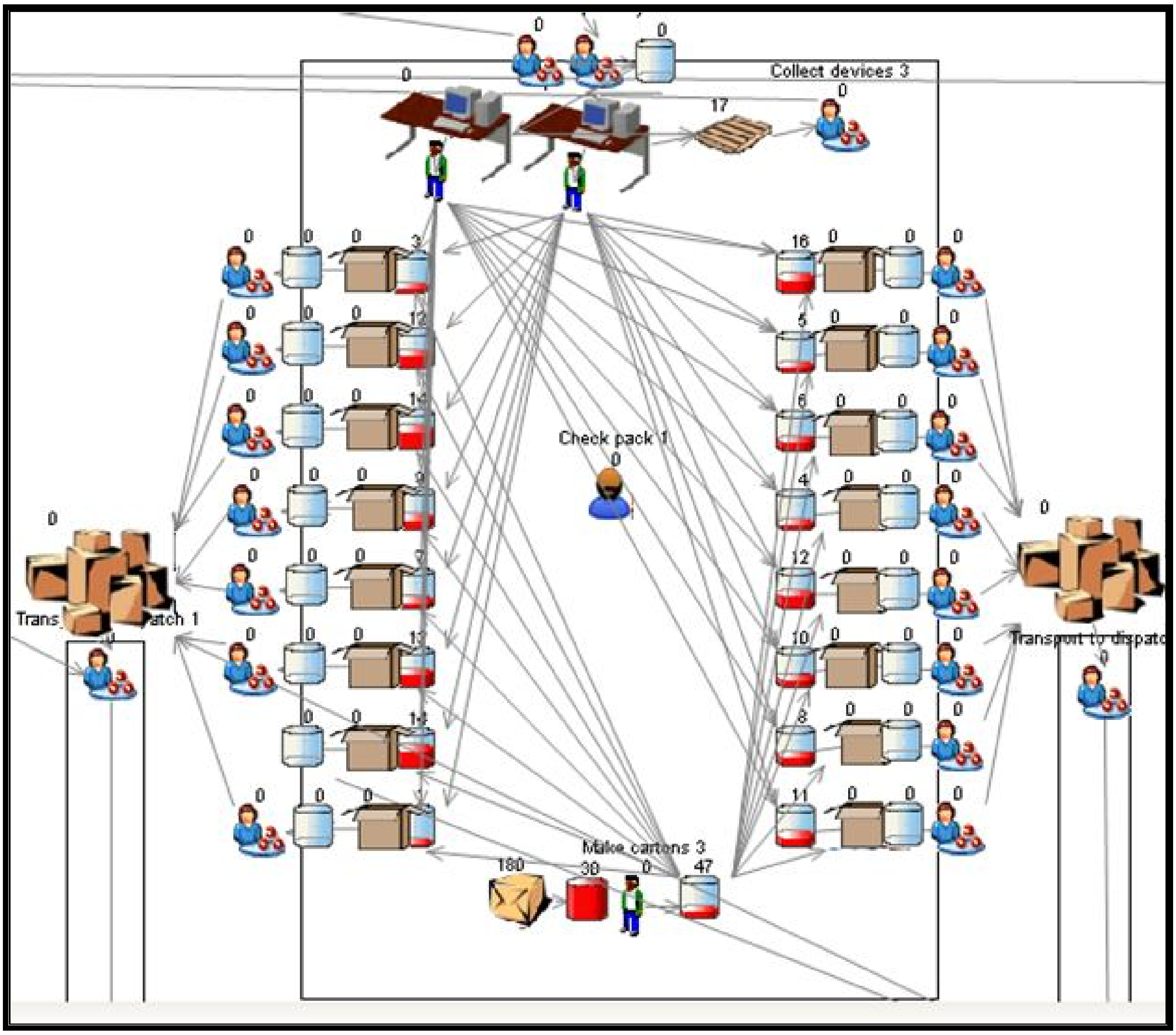

Figure 4.8: Screen shot of one of the Simul8 model despatch zone sub-sections

4.5.5. Supporting components

There are a number of supporting components to the model that facilitate its smooth operation:

- There are four forklifts, two each in the receiving and despatch zones respectively.
- There is a material handler that collects empty containers from the despatch zone and returns them to the receiving zone.

A complete representation of the DCDS model in the Simul8 format is shown in Figure 4.9 below.

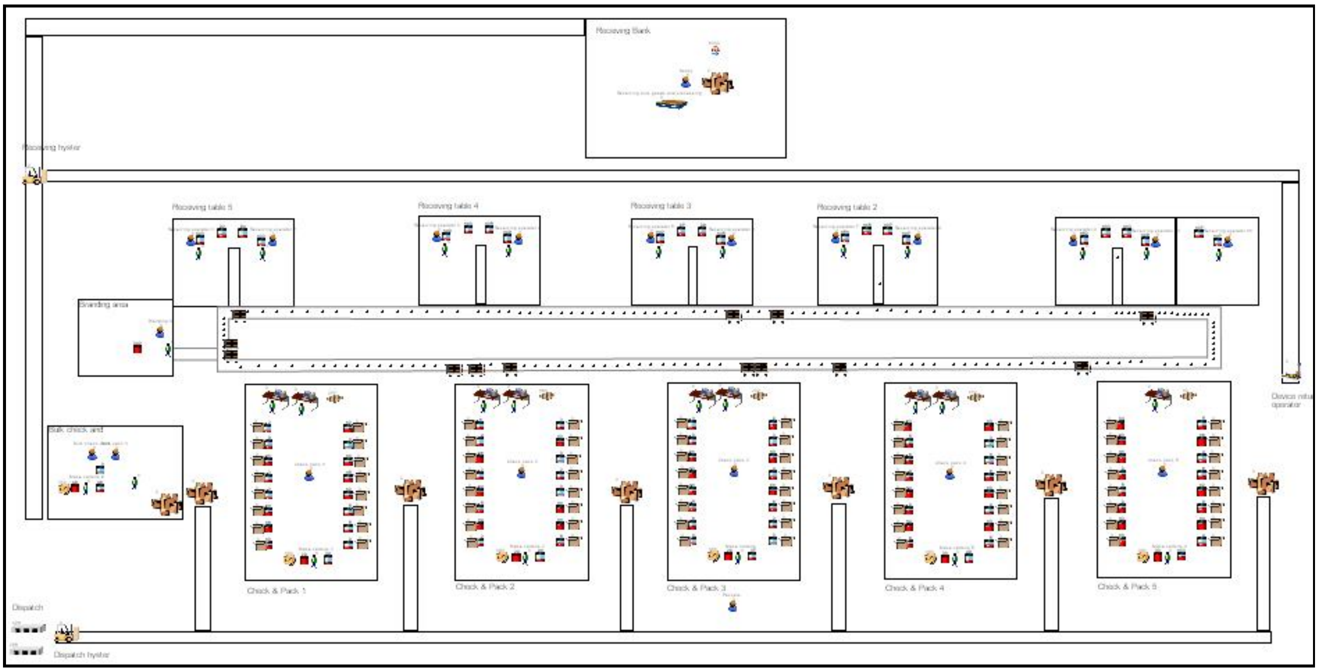

Figure 4.9 Screen Shot of DCDS Model

4.6. Model Verification

During the model construction, the management team of the system being simulated were consulted to ensure that the logical structure and input parameters were correct. The model development did not progress unless they verified each component.

4.7. Model Validation

The DCDS model forms the foundation of the hybrid model, and it was important that it be an accurate representation of the real system. The validation stage was vital in proving this, and allowing for future experimentation to occur. The model was validated by comparing the output data of the simulation runs to throughput information extracted from the production tracking database that is used to monitor the real system.

It was only possible to collect five points of output during the data collection window period (Table 4.1). This was because there were only five days when the operation operated under optimum conditions. The rest of the time, breakdowns, absenteeism, and material supply shortages caused stoppages and delays which caused bias in the output data.

No	Date	Activities Completed Per Day
1	20/04/2005	2397
2	10/05/2005	2392
3	11/05/2005	2476
4	12/05/2005	2417
5	16/05/2005	2335

Table 4.1 Output of Real System Data

One hundred replications of the DCDS model were then run using the input parameters that were consistent with the running conditions of the real system, of which the following were the most important:

- Volume of incoming activities (6300 per 445 minutes of operational time)
- Conveyor speed (40 meters per minute)

The resulting output was recorded and can be seen in Table 4.2. The data is summarised in Table 4.3 under the "Simul8 Output Data" column.

Run No	Activities Completed Per Day	Run No	Activities Completed Per Day	Run No	Activities Completed Per Day	Run No	Activities Completed Per Day	Run No	Activities Completed Per Day
1	2403	21	**2415**	**41**	**2365**	**61**	**2414**	**81**	**2432**
2	2415	22	**2432**	**42**	**2420**	**62**	**2458**	**82**	**2419**
3	2432	23	**2409**	**43**	**2417**	**63**	**2436**	**83**	**2416**
4	2436	24	**2415**	**44**	**2450**	**64**	**2401**	**84**	**2402**
5	2421	25	**2418**	**45**	**2417**	**65**	**2431**	**85**	**2404**
6	2419	26	**2404**	**46**	**2393**	**66**	**2404**	**86**	**2441**
7	2412	27	**2441**	**47**	**2432**	**67**	**2422**	**87**	**2422**
8	2415	28	**2408**	**48**	**2407**	**68**	**2410**	**88**	**2374**
9	2415	29	**2401**	**49**	**2418**	**69**	**2438**	**89**	**2426**
10	2418	30	**2374**	**50**	**2444**	**70**	**2426**	**90**	**2416**
11	2418	31	**2426**	**51**	**2435**	**71**	**2423**	**91**	**2455**
12	2401	32	**2426**	**52**	**2386**	**72**	**2415**	**92**	**2423**
13	2459	33	**2446**	**53**	**2425**	**73**	**2425**	**93**	**2430**
14	2445	34	**2415**	**54**	**2425**	**74**	**2416**	**94**	**2413**
15	2413	35	**2430**	**55**	**2431**	**75**	**2415**	**95**	**2392**
16	2427	36	**2425**	**56**	**2435**	**76**	**2406**	**96**	**2426**
17	2393	37	**2392**	**57**	**2430**	**77**	**2407**	**97**	**2406**
18	2401	38	**2440**	**58**	**2444**	**78**	**2421**	**98**	**2426**
19	2364	39	**2393**	**59**	**2416**	**79**	**2426**	**99**	**2409**
20	2388	40	**2458**	**60**	**2399**	**80**	**2402**	**100**	**2387**

Table 4.2 Results of the DCDS model runs for validation purposes

Measure	Real System Output Data	Simul8 Output Data
Average	2403	2414
Standard Deviation	50.76	20.70

Table 4.3 Comparison of real system and DCDS data

The average results for the real system data and the simulation data are given in Table 4.3, and then compared statistically. These data are assumed to be random samplings of independent measurements of each system. The two data sets are normally distributed (normality tests yield p-values of 0.542 and 0.366). Levene's test [35] yields a p-value of 0.069 thus the two data sets have equal variances. These conditions allow the use of a two sample t-test which when applied to the data yields a p-value of 0.431 and as this is significantly greater than 0.05 there is no statistical difference between the results in Table 4.3 (The same methodological approach is used for all the statistical tests used throughout this work).

The standard deviations of the two data sets are however different from each other, and this can be attributed to the naturally occurring variation in the real system, which is slightly higher than that from the controlled experimental environment. Slight differences in standard deviation between the data sets are expected due to the complex nature of the model and real system.

It appears reasonable to conclude that the DCDS model does adequately simulate the real system and there is sufficient validity for it to be used for further experimentation.

4.8. Model Experimentation Results

The first objective was to establish a foundation of throughput data for comparison to results from the hybrid model. In its current form, the model can confidently be used as a simulation of the real system, and thus it is assumed that the resulting volume of activity throughput is an accurate reflection of what the system can produce in a steady state.

The model simulation run period used was equivalent to 5 days of production time, each day consisting of 445 minutes. Thus the total simulated time per replication was 2225 minutes. The model was replicated 20 times so that an equivalent of 100 days of simulation runs, and an average of 12495 activities were completed during the experiment run. As discussed earlier in the chapter, the main output measure is activities completed, and a summary of the output results can be seen in Table 4.4. Figure 4.10 shows a box plot of the results.

Measure	**DCDS Output Data**
Average	12495
Standard Deviation	47.20

Table 4.4: Activity output summary for DCDS experiment run

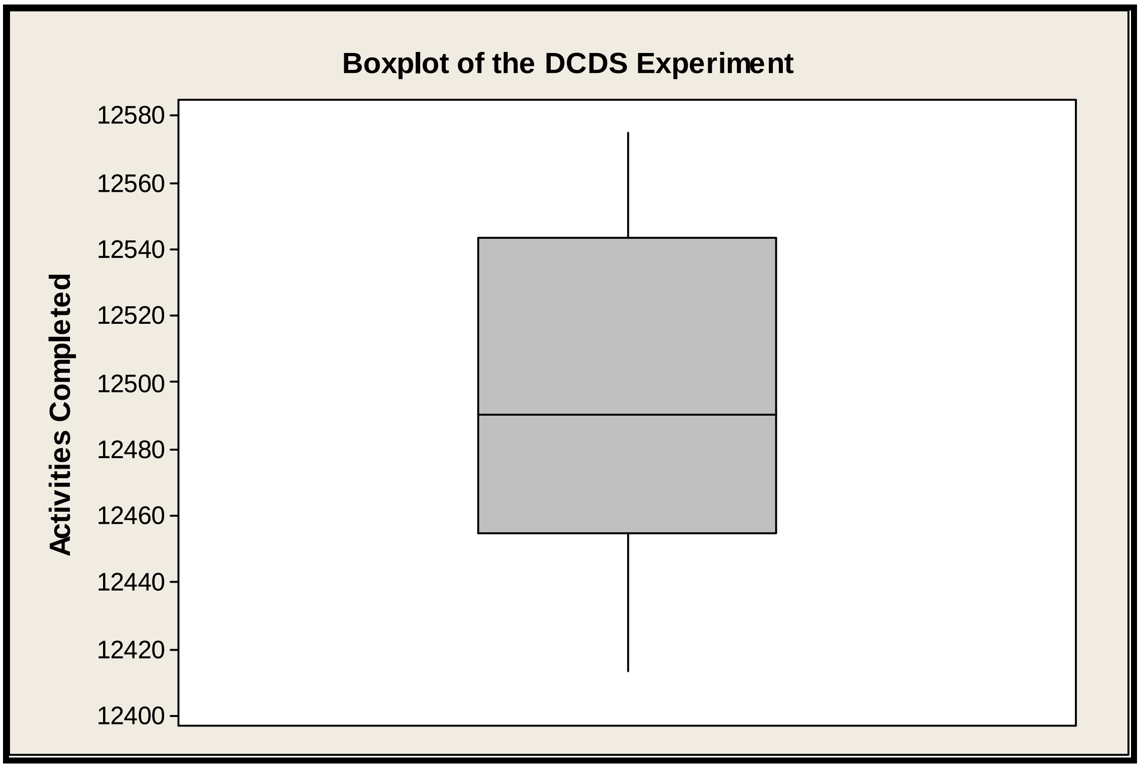

Figure 4.10: Box plot showing the output results of the DCDS experiment run

Whilst “activities” completed is the primary output measure, an important secondary measure is resource utilisation, which gives an indication of the intensity of resource activity at the two main processing zones (the receiving and despatch zones).

This is calculated by categorising a resource in the model as being in either a “busy” or “idle” state, and recording the amount of time that it remains in that state. The amount of time that a resource occupies each state is measured and displayed as a percentage of total time. Utilisation percentage greater than 75% is assumed to be a satisfactory indicator that a resource has been appropriately occupied with work.

The receiving zone section is relatively static with an average utilisation level of 99%, which indicates that this area of the model was constantly fed with sufficient work to keep the operator agents busy for nearly the entire simulation time. Figure 4.11 shows a box plot that summarises the receiving zone operator agent utilisation. Of interest is the lack of data spread, which indicates the consistency of the utilisation over the twenty runs completed.

The despatch zone section of the model is of the most interest because it is here that each activity must pass in order to exit the system. Analysis of the DCDS data indicates that on average the despatch zone resources were utilised in excess of 85% of the run time, which satisfies the utilisation requirement. Figure 4.12 shows a box plot which summarise the utilisation results, and shows the variation in the utilisation over the twenty experiment runs.

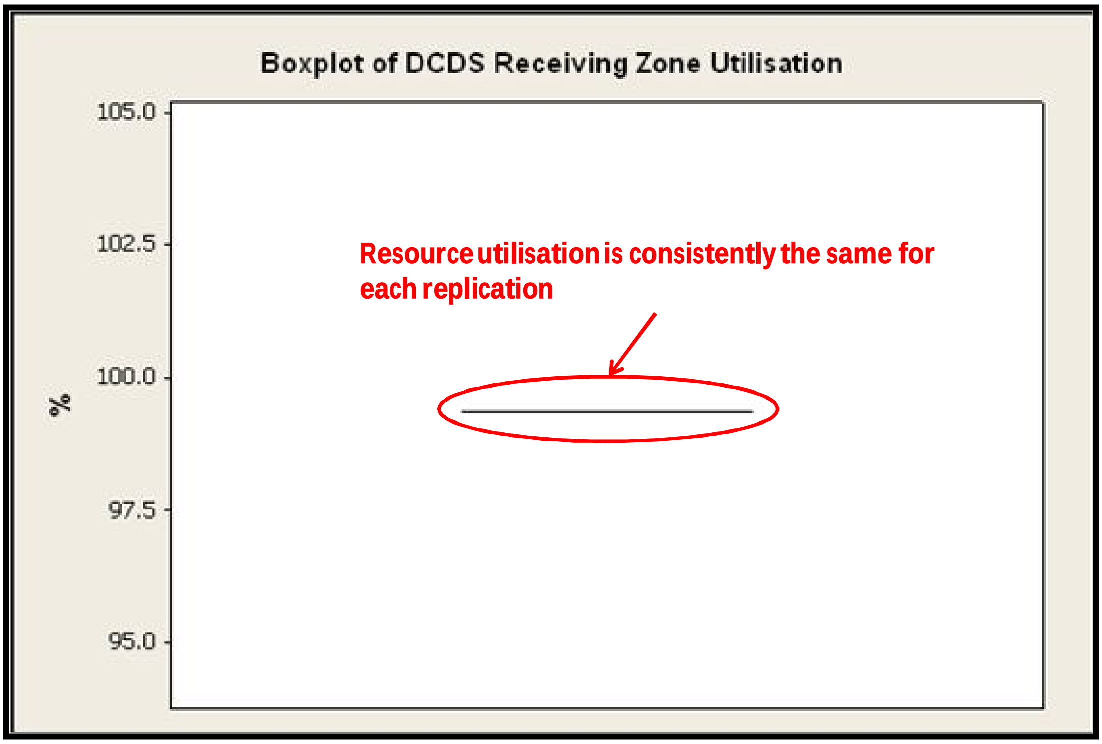

Figure 4.11: Box plot showing receiving zone resource utilisation

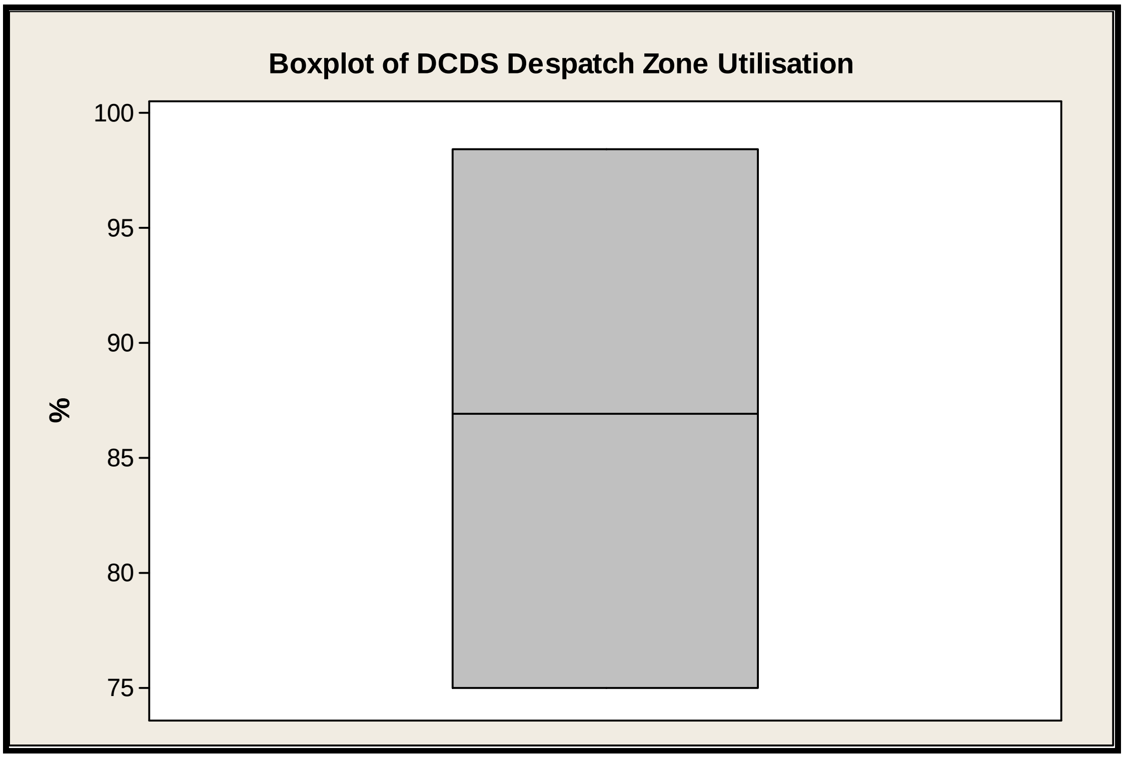

Figure 4.12: Box plot showing despatch zone resource utilisation

4.9. Determining the optimum number of replications

Due to the complexity of the simulation model and the limited computer processing resources available, the run time for one replication of this model is between 10 and 12 hours. Usually, simulation exercises like this might be run many times but due to these limitations a compromise was needed in order to get through all the experimental runs required.

An exercise was completed where the output data was analysed to determine the minimum number of replications required for accurate model interpretation. The method of batch means was used [17], which involved subdividing the output data from the main experiment model run into batches of 5 measurements (refer to Table 4.5), and then comparing the average output for each.

Measure	**Batch 1**	**Batch 2**	**Batch 3**	**Batch 4**
1	12500	12471	12575	12555
2	12453	12438	12459	12546
3	12551	12442	12520	12477
4	12413	12536	12493	12464
5	12450	12562	12491	12489
Mean	**12473.4**	**12489.8**	**12507.6**	**12506.2**

Table 4.5: Output data from the DCDS model divided into 4 batches of 5 samples

The resulting means were statistically compared using the test for equal means (ANOVA). The p-value from the ANOVA test was 0.663 and therefore there is significant difference between the batch means. Thus 5 replications should be sufficient for the simulation experiments of this thesis. In order to ensure accuracy, all simulation experiments conducted in chapters 5 and 6 were replicated 10 times.

4.10. Model Conclusion

Comparisons with the real system's output data give confidence that the DCDS model is an accurate representation of the reality and could be used as a foundation for the hybrid model. Thus, it is a robust platform for experimentation, and is used in the development of the Operator Agent Behaviour (OAB) model in Chapter 5.

Chapter 5

The Operator Agent Behaviour Model

5.1. Introduction

The second model creates a platform for the simulation of the deterioration of a human operator at work. Also this operator can subsequently be replaced with another operator with the capability to learn through experience (time on the job). The aim is to understand the difference in productivity when a system experiences these characteristic human behaviours, and to try to predict how to sustain an acceptable level of productivity under these circumstances.

5.2. Model conceptualisation

The software Simul8 allows a user to input data to a model from an external source (e.g. Microsoft Excel), which allows for great flexibility when experimenting with various simulation scenarios. The software's logic is such that a choice can be made between using the software's internal random number generator, or to "read" from another form of number input. This compatibility was necessary because of the complexity of the input distributions that were needed to simulate operator deterioration and learning behaviour.

The operator agent behaviour model was developed to allow for deterioration of the operators that had fixed work functionality in the discrete model of chapter 4 (in fact operators are not explicit in that model because their inputs are fixed). Deterioration, in the context of this work, occurs when an operator agent experiences a decrease in performance over a period of time.

This deterioration is not linear, but fluctuates with a downward trend. This type of deterioration is used because it reflects observed behaviour in the real system where operators typically will experience short term improvements in behaviour after resting between processing batches of activities.

Learning occurs when a new operator agent replaces a deteriorating operator agent who has reached the end of the deterioration cycle. The new agent experiences a learning curve that follows a randomly increasing pattern, as was observed in the real system when operator's performance did not increase in a linear pattern thus indicating a nonlinear learning cycle.

There are three factors that influence the learning cycle:

- The amount of time spent working on the job
- The number of activities completed
- The amount of time spent with a manager who provides training and guidance

The model's basic mechanics are to create an input list of process times (time taken by an operator agent to complete one activity of throughput) in Microsoft Excel that can be manipulated and then integrated with the DCDS model. The DCDS model could then be run with different input spreadsheets and the results analysed from data supplied by the Simul8 software.

The input spreadsheets facilitate sufficient flexibility to allow for experimentation with the model. When combined with the DCDS, the outcome is a model that simulates a complex system of process flow, and the effect of erratic and dynamic behaviour that is characteristic of human beings in the work place. This model will henceforth be known as the Operator Agent Behaviour (OAB) model, and its interaction with the DCDS model can be seen in Figure 5.1.

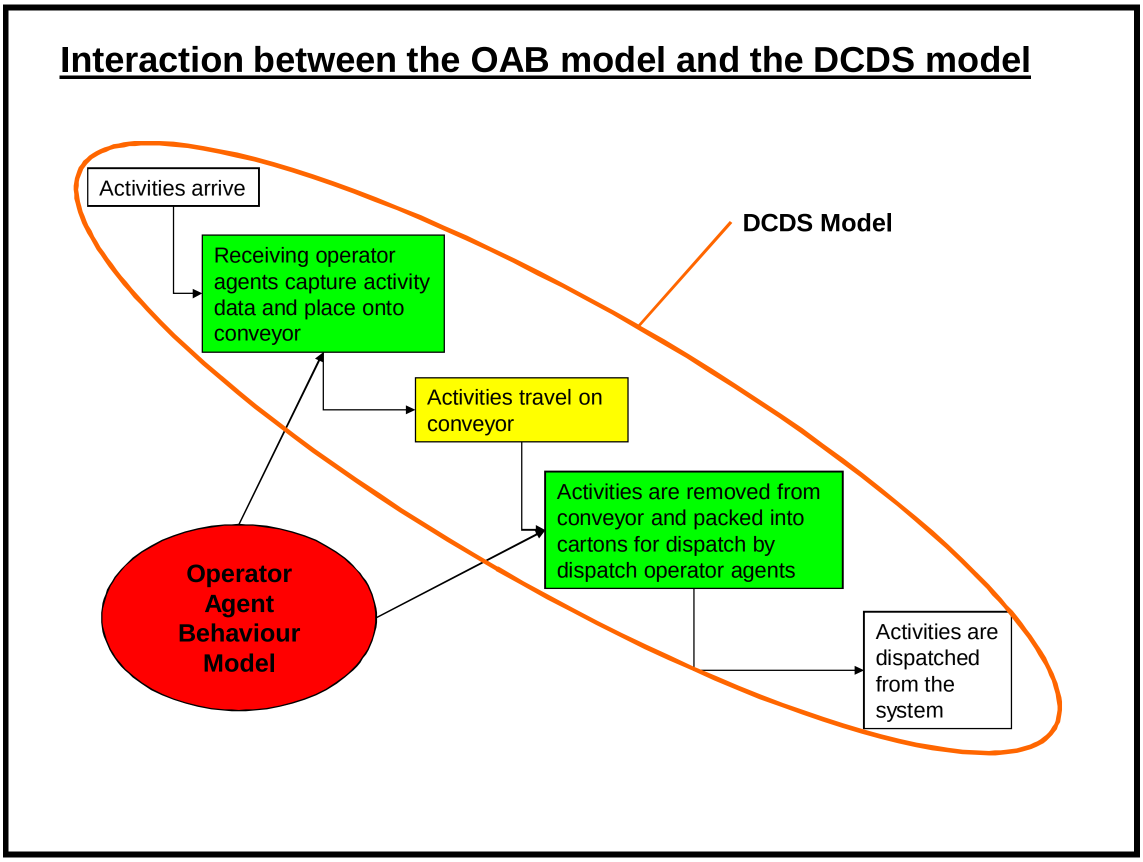

Figure 5.1: Interaction between the DCDS and OAB models

Figure 5.2 shows a flow chart of the OAB model logic, the red components occur within the OAB model whilst the green blocks are the parts of the DCDS model with which it interacts (as per the colours in Figure 5.1). Of particular interest are the deteriorating and learning factors that are applied to the activity process times which create new input standards. These factors are explained in more detail later in the chapter.

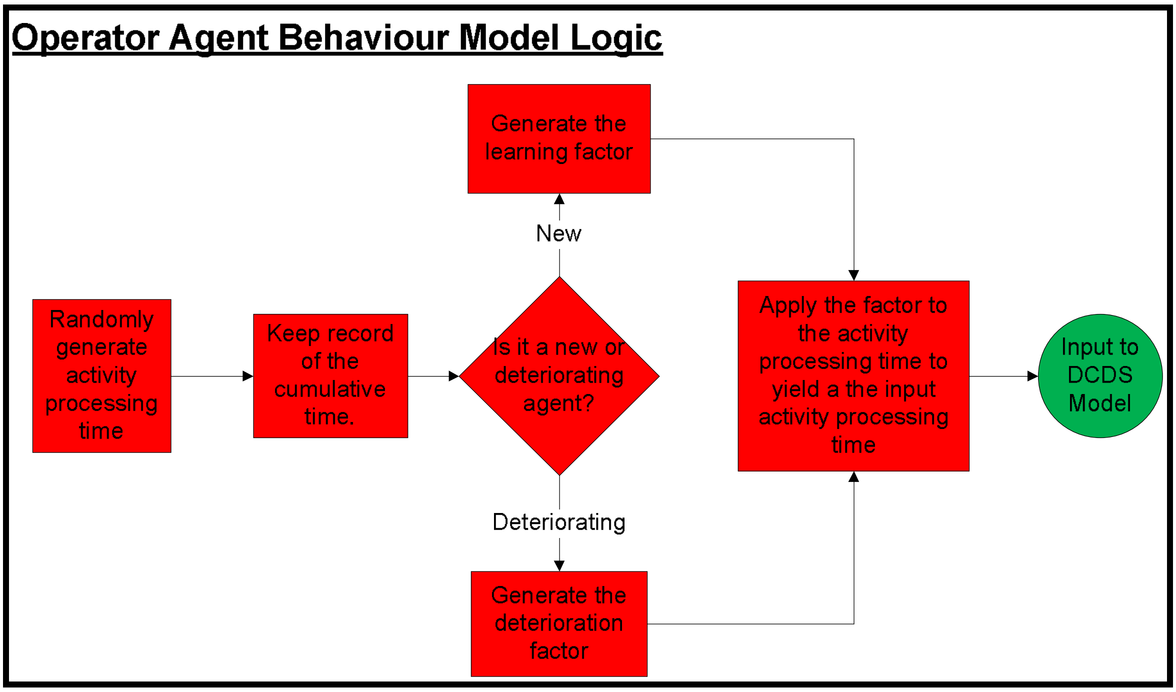

Figure 5.2: OAB model logic

5.3. Model development

The OAB model consisted of a Microsoft Excel spreadsheet used as the external interface with Simul8 because Microsoft Excel is an easily accessible and a robust platform. There are 20 operator agents in the DCDS model, each of which required separate input “sheets” within the spreadsheet. Thus the Microsoft Excel document comprises 20 “sheets”, each labelled according to the operator agent that it is inputting to.

These 20 operator agents can be further divided up into 11 receiving and 9 dispatching agents, each team doing different tasks and thus having different process times.

Each “sheet” comprises six columns (see Table 5.1 below) labelled as follows:

- Day number (A)

  As the model runs it accumulates time, which needs to be accumulated and accounted for. In Simul8, as in all other discrete models, an internal simulation clock is used [3]. This column acts as the simulation clock and is used to identify and categorise the passage of time, usually in days.

- Cumulative total (B)

  Each production period being modelled consists of a specifiable amount of time which is made available for work. Each operator agent cannot work for more than this time and this column is used to track the amount of time that has transpired by cumulatively adding up completed activity process times.

- Process time from Minitab (C)

  A list of random numbers using the triangular and exponential distributions is generated by two external random number generators; Minitab, a statistical software analysis program, and Crystal Ball, a statistical software package that works with Microsoft Excel.

- Agent specific ratio (D)

  The deterioration or learning factor is inserted into this column, which along with the management influenced ratio column, is the most important factor in the model due to influence that they wield on the final input time.

- Management influence factor (E)

  The agent based model interfaces with the discrete model through this column. The method used will be explained later in Chapter 6.

- Input variable (F)

  This is the final input variable that is used by the DCDS model. It is the product of the process time, the agent specific ratio, and the management influence ratio.

A	B	C	D	E	F
Day Number	**Cumulative Total**	**Process Time**	**Deterioration / Learning Ratio**	**Mgt Influence Factor**	**Input Variable**
	428.6351	1.22288	1.896613	1	2.31933
	429.8284	1.19336	1.787674	1	2.133339
	431.0159	1.18746	1.665011	1	1.977134
	432.1873	1.1714	1.79512	1	2.102803
	433.3897	1.20245	1.822524	1	2.191494
1	434.5566	1.16689	1.839388	1	2.146364

Table 5.1 Example of an agent input sheet to the DCDS model

5.4. OAB model components

5.4.1. Process time

As mentioned earlier in this chapter, Crystal Ball and Minitab software were used to generate the process time inputs.

Minitab was developed by Pennsylvania State University in 1972 and it performs basic and advanced statistical functions. Although more commonly used by Six Sigma practitioners for statistical analysis, it has a powerful random number generator engine which proved to be useful in this work. Minitab was chosen due to its ease of use, and because the holding company, Minitab Inc, had made an academic version of the software available to the author.

5.4.2. Calculation of the input variables

Operator agents in the receiving zone are involved in processing and logging an activity into the information control system. The system prints out a barcode label which is attached to the activity item. The operator agent places the activity into a container, which is put onto the conveyor. The measured average time to complete one activity as per the above sequence is 1.9 minutes with a standard deviation of 0.138 minutes.

In the despatch zone, operator agents collect the relevant activities for the conveyor, discard the container, and then prepare the activity for dispatch. This involves entering some data into the SAP stock control system, placing the activity into a carton, sealing the carton and affixing the postal label. The measured average time to complete one activity is 1.2 minutes with a standard deviation of 0.026 minutes.

5.4.3. Deterioration / Learning factor

The deterioration / learning factor is one of the cornerstones of this research. Whilst the concept of applying a factor to a standard time to change its value is not new [36] (the science of Work Study applies a rest allowance to measured process times in order to represent operator fatigue or job difficulty), the method used in this work is novel.

Due to the lack of empirical data about deterioration and learning in the area being modelled, a number of assumptions were made from discussions with the manager of the system in order to construct this model:

- A production week of 5 working days consisting of 8 hours each is an optimal simulation period for experimentation.
- Within the 5 day period, an operator agent will deteriorate from a factor of 1(no deterioration) to a factor of 2 (fully deteriorated for an experienced operator agent) over a two day period. Thus when the factor is applied to the input time, the operator agent becomes half as productive at the end of day 2 as it was at the beginning of the first day.
- When a new operator agent replaces a fully deteriorated operator agent, its input variable is the result of the process time multiplied by a factor of 2.5 (completely unskilled for the task to be completed). This factor was derived from the assumption that a new operator entering the real system would be even less productive than a deteriorating operator due to inexperience. Over a period of three days, this factor is reduced until it is equal to 1, which indicates that the new agent has now reached full proficiency for the process being performed.

Even though the operator agent experienced linear deterioration and learning over time, there was variation within the linear time movement. This was due to the operator in the real system experiencing phases of different productivity during the working day, which needed to be reflected in the model (i.e. more productive at the start of the shift than at the end). It is this variability that is difficult to model using a single standard and thus necessitated the multiplication of the input standard by the deterioration / learning factor.

Kelton [3] suggests that the triangular distribution is "commonly used in situations in which the exact form of the distribution is not known, but estimates (or guesses) for the minimum, maximum, and most likely values are available". Thus the triangular distribution was chosen as the most suitable statistical distribution to model the variation within the deterioration and learning models.

The version of Minitab used was unable to generate random numbers for theTriangular Distribution required for this factor, and thus Crystal Ball was used. Crystal ball is a risk analysis, simulation, and optimisation software program developed by Dicisioneering Inc, a Denver, Colorado based software development company. It is an effective tool used to develop Monte Carlo simulation models in MS Excel, and has a powerful random number generator, along with a wide range of statistical distributions.

The parameters for deterioration and learning were inputted to Crystal Ball and a random number generated. An example of the variability in the operator agent's deterioration and learning can be seen in Figure 5.3.

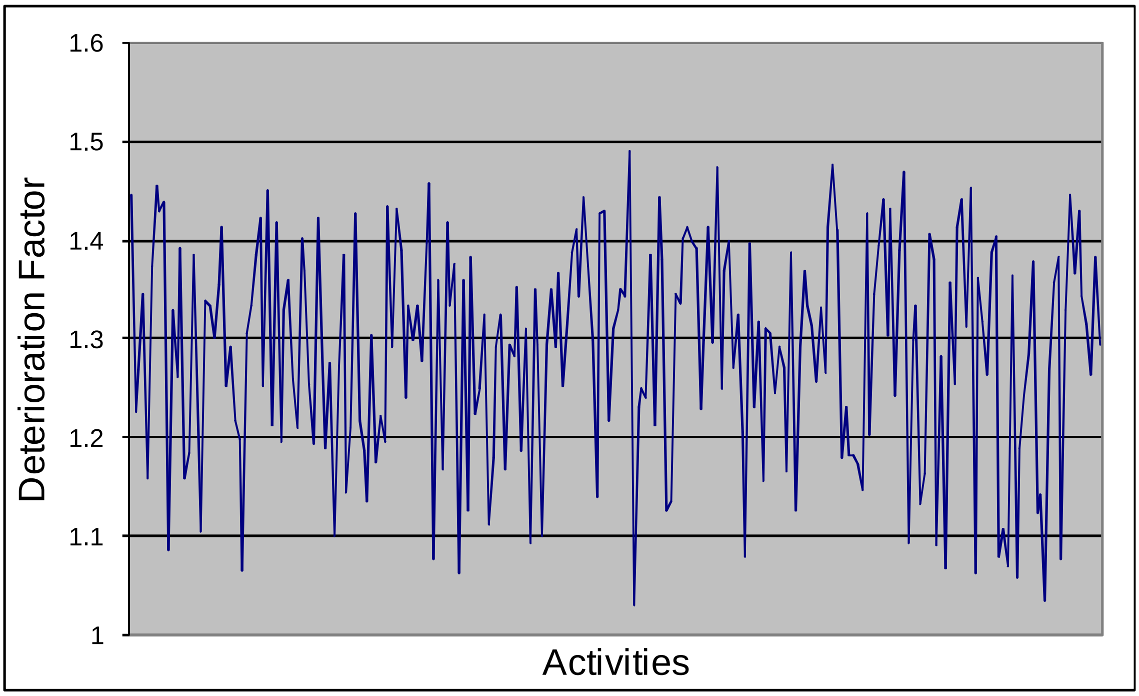

Figure 5.3 Example of the variation within operator agent deterioration or learning

5.5. Model verification and validation

Once the complete spreadsheets had been built and all the input variables prepared, the new input model was tested to determine if the results made logical sense. This was a difficult task because no real system data existed to validate the model. Instead, the results of the model runs were shown to the management team of the cross-docking organisation and they were asked to evaluate and validate the model. All team members agreed that the results were realistic and consistent with their experience and personal opinions.

5.6. Model experiment results

5.6.1. Objective of the experiments

The model is capable of simulating human deterioration in a production environment, and the experiments are structured to determine the difference between an optimum system in steady state and one that has fluctuating operator performance.

Since there are two main areas where operator agents interact with activities, the experiments are orientated towards testing the limits of these areas.

The following three experiments were run:

5.6.1.1. Receiving zone experienced deterioration and learning only

5.6.1.2. Despatch zone experienced deterioration and learning only

5.6.1.3. Receiving and Despatch zones experienced deterioration and learning

5.6.2. Simulation experiment setup

As per the experimental setup discussed in this chapter, five working days of production is simulated. Each production day consists of 445 minutes of time available for work, with a total of 2225 minutes of simulated time per replication.

Each operator agent nominated for deterioration was programmed to experience productivity deterioration over a two day (890 minutes) period at the rates defined earlier in this chapter. After this full period an operator agent becomes too inefficient to remain at the work station and is replaced by a new operator agent.

The new operator agent experiences learning over a 2.5 day period (1112 minutes) at the "learning" rate discussed in section 4.3. After this period the new operator agent becomes a healthy operator agent and performs at an optimum productivity level. Table 5.2 shows the state change of an operator agent over a single replication of a model run.

Day	**Agent Type**	**Operator agent state**
Day 1	Deteriorating Operator Agent	Deterioration
Day 2	Deteriorating Operator Agent	Deterioration
Day 3	New Operator Agent	Learning
Day 4	New Operator Agent	Learning
Day 5	New Operator Agent	Learning (Operator Agent becomes a Healthy Operator Agent)

Table 5.2: State changes of an operator agent in the OAB model

5.6.3. Simulation Experiment 1: Receiving zone operator agents experience deterioration and learning only

In this experiment, all the receiving zone operator agents experience deterioration, and after two production days are all replaced by new operator agents. The despatch zone operators maintain a steady state as healthy operator agents.

Table 5.3 shows a summary of the experiment results, and the average system throughput of this experiment was 7972 activities, which is significantly lower than the steady state experiment completed in Chapter 4 (the DCDS experiment). A two sample t-test comparing the outputs of the DCDS experiment and Experiment 1 yielded a p-value less than 0.05, thus indicating that they are significantly different from each other.

5.6.4. Experiment 2: despatch zone operator agents experience deterioration and learning only

In this experiment, all the despatch zone operator agents experience deterioration, and after two production days are all replaced by new operator agents. The receiving zone operators maintain a steady state as healthy operator agents.

Figure 5.4 shows a box plot of the results of 10 replications, with an average of 8913 activities completed on average (refer to Table 5.5). Compared to the steady state experiment completed in Chapter 4 (the DCDS model runs), there is a significant difference in productivity, with experiment 2 yielding a lower system throughput count of activities.

Again statistical analysis of the output data was conducted and it was concluded that the two models are significantly different from each other.

Figure 5.5 shows the box plot of Experiment 2 compared to the other experiments, and shows that the average system throughput or 8913 is significantly lower than the steady state experiment completed in Chapter 4 (the DCDS experiment). A two sample t-test comparing the outputs of the DCDS experiment and Experiment 2 yielded a p-value less than 0.05, thus indicating that they are significantly different from each other.

5.6.5. Experiment 3: receiving and despatch zone operator agents experience deterioration and learning only

In this experiment, all the receiving and despatch zone operator agents experience deterioration, and after two production days are all replaced by new operator agents.

The average system throughput of the experiment was 7908 activities (refer to Table 5.3), which is significantly lower than the steady state experiment completed in Chapter 4 (the DCDS experiment). A two sample t-test comparing the outputs of the DCDS experiment and Experiment 3 yielded a p-value less than 0.05, thus indicating that they are significantly different from each other.

Measure	Output Data of Steady State DCDS	Output Data of Experiment 1	Output Data of Experiment 2	Output Data of Experiment 3
Average	12495	7972	8913	7608
Standard Deviation	47.20	9.65	10.66	6.99

Table 5.3: Summary of output results of OAB - Experiment 1 compared to DCDS run

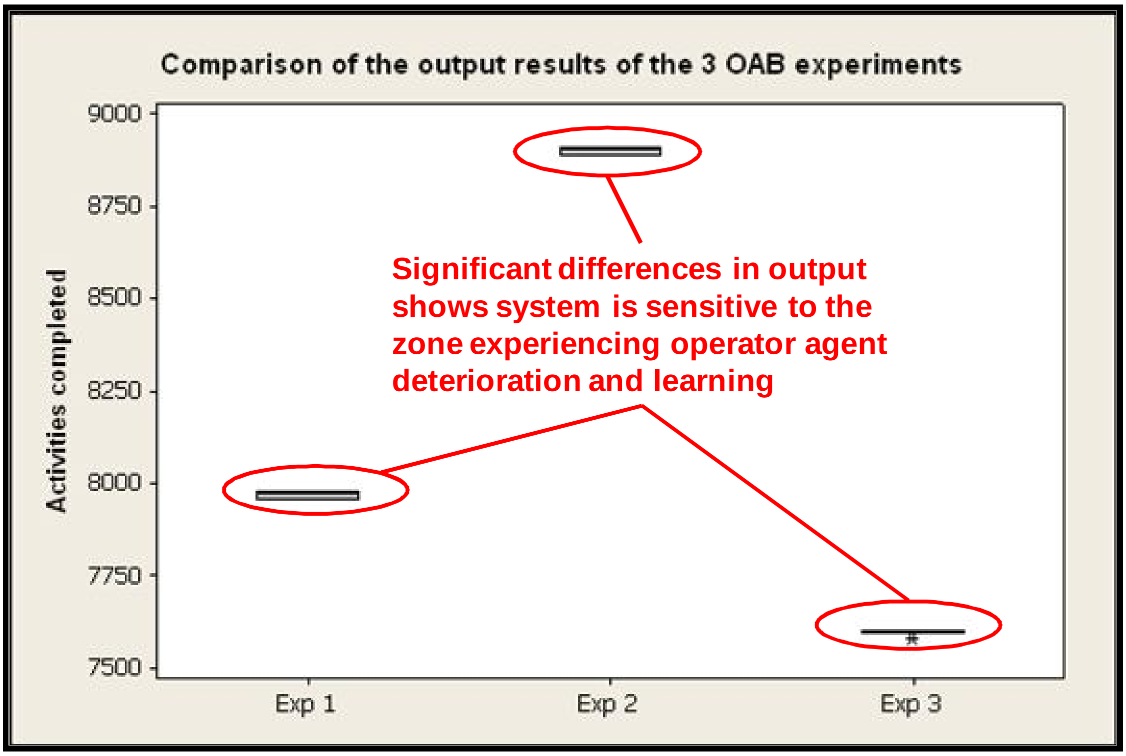

Figure 5.4: Box plot showing a comparison of the output of the three OAB experiments

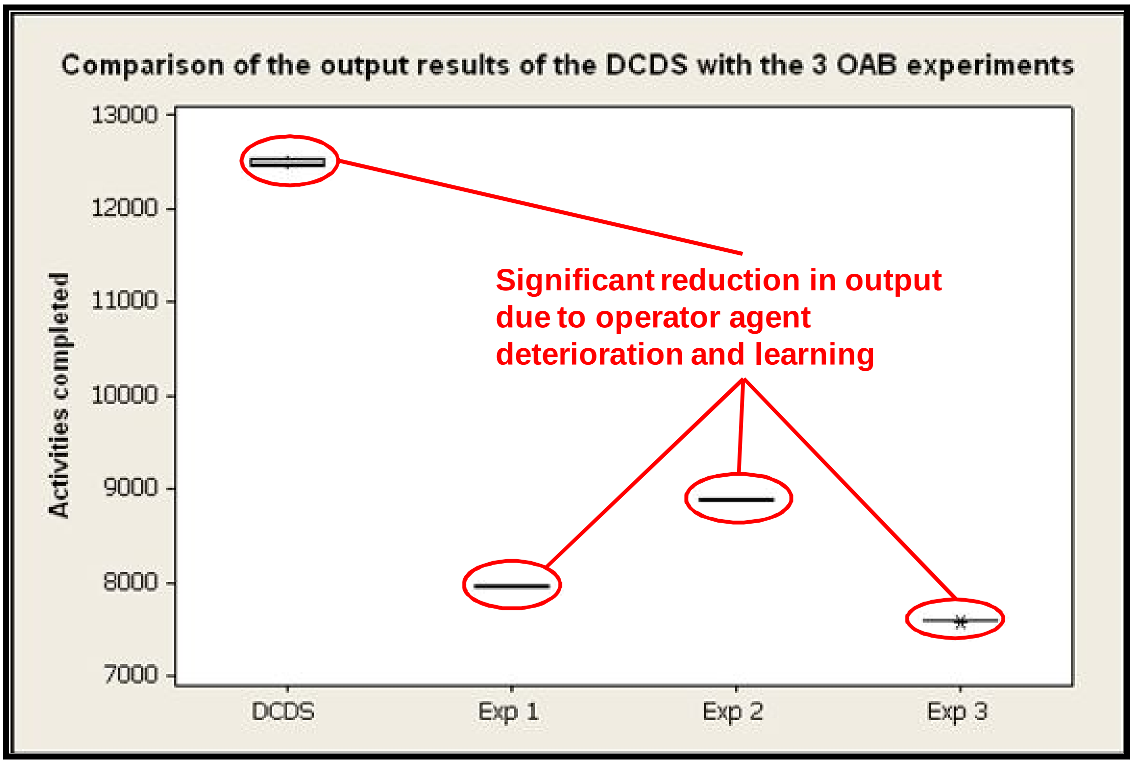

Figure 5.5: Boxplot showing a comparison of DCDS output to the three OAB experiments

5.7. Summary of experimental results

An Analysis of Variance statistical test conducted on the three experiments yielded a p-value less than 0.05 which indicated that they are significantly different from each other.

Thus it can be generally concluded that the deterioration of existing operator agents and their replacement with new operator agents has a significant negative effect on the system output. What was surprising was the model's sensitivity to the specific department experiencing deterioration / learning, with the system being less productive whenever the receiving zone was experiencing deterioration / learning. Figure 5.4 indicates the magnitude of the difference between the three experiments.

A review of the resource utilisation for experiment 1 shows that the receiving zone operator agents were constantly busy for 99% of the simulation time, which indicates that they were constantly supplied with work (and full queues) for the duration of the model runs. Figure 5.6 shows a comparison of the receiving zone operator agent utilisation for the three OAB experiments.

This despatch zone operator agent utilisation shows different results to the receiving zone (Figure 5.7). In Experiment 1, the average despatch zone operator agent utilisation was 55% which indicates that these resources were idle for 45% of the simulation duration. It can be concluded that the reduced productivity at the receiving zone due to deterioration / learning has resulted in a lack of activities being pushed through the system to the dispatch zone, and thus lower levels of output.

For Experiment 2, the average despatch zone operator agent utilisation is 91%, which indicates that a steady flow of work is entering the system and is keeping the despatch zone operator agents busy for most of the simulation period. It can be concluded that when the receiving zone is working at full productivity, it is possible to obtain higher levels of output despite the despatch zone operating at lower productivity levels due to deterioration and learning.

Experiment 3 shows that the average despatch operator agent utilisation is 69%, although there is significant fluctuation between the maximum utilisation and the minimum, which can be attributed to the variable flow of work moving through the system from the receiving zone. It can again be concluded that the lower productivity of the receiving zone operator agents is causing a starvation of work for the entire system, resulting in lower levels of output and underutilisation of despatch zone resources.

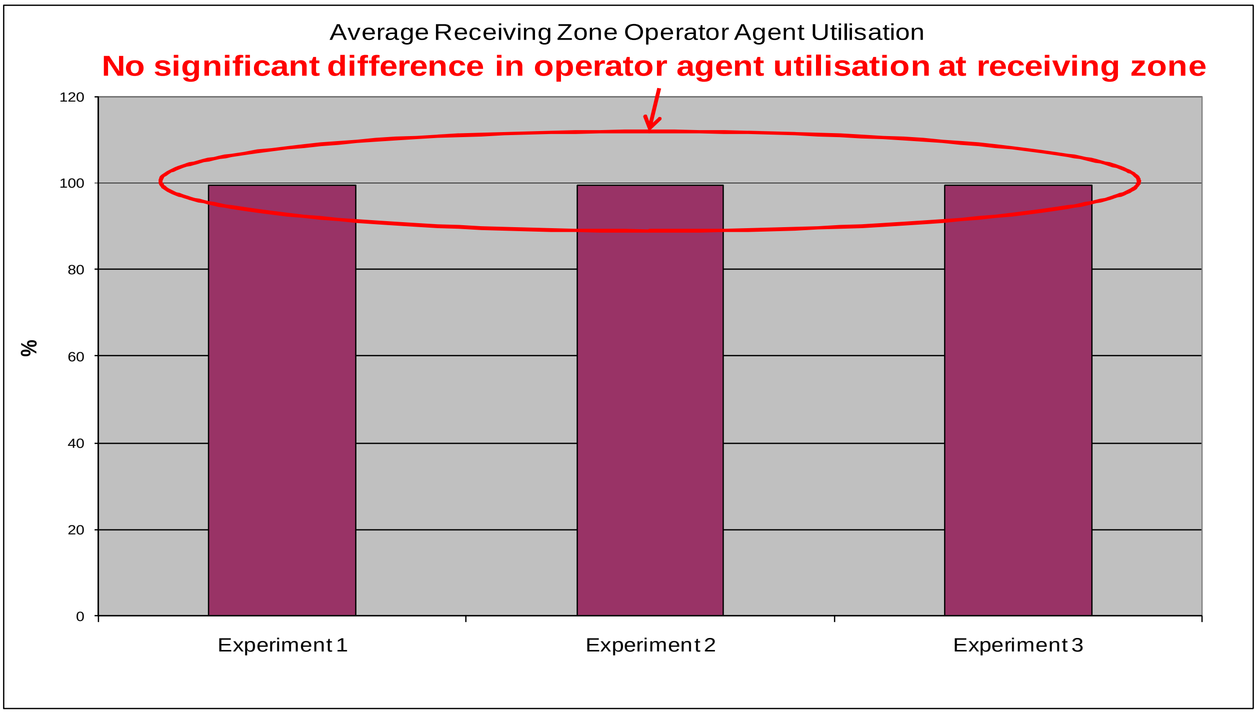

Figure 5.6: Comparison of the receiving zone operator agent utilisation for the three OAB experiments

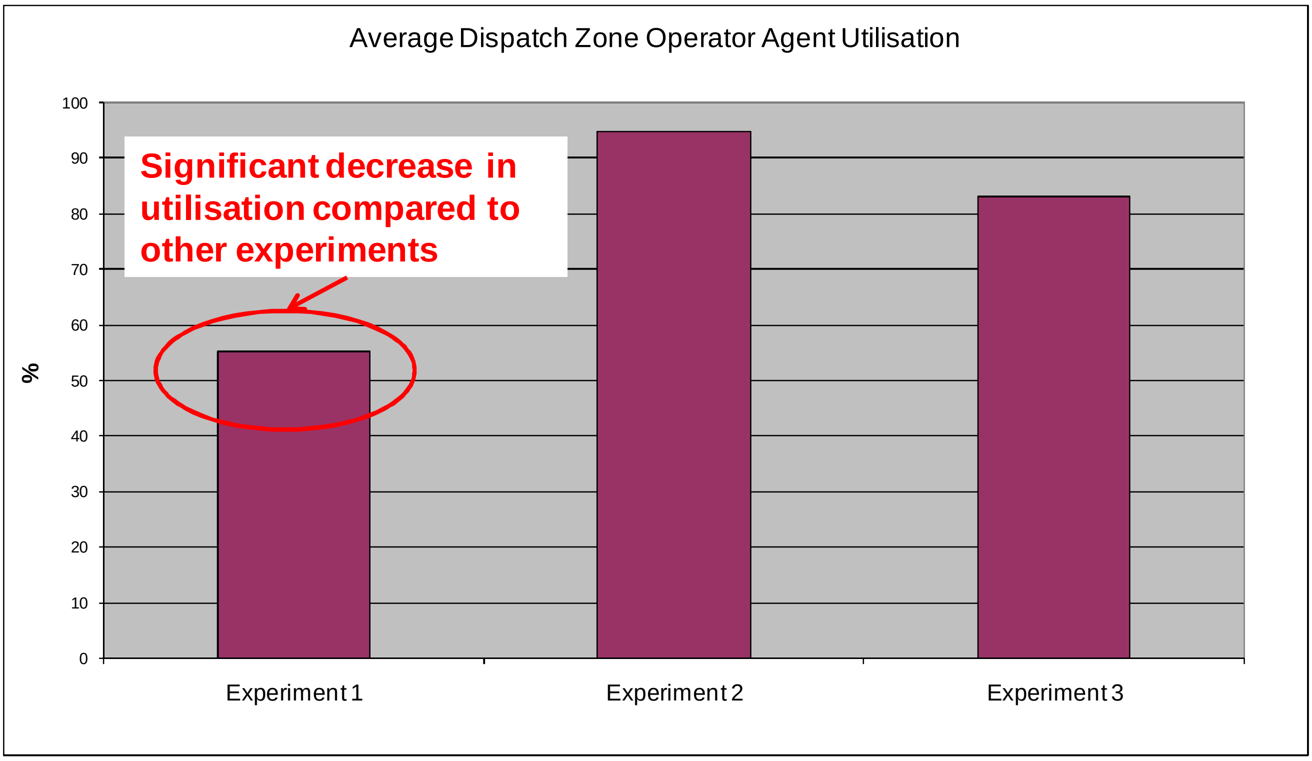

Figure 5.7: Comparison of the dispatch zone operator agent utilisation for the three OAB experiments

5.8. Further experimentation

A fourth experiment was run to test the robustness of the model. The model was configured so that both the receiving zone and the dispatch zone experienced deterioration for only one day (instead of two), whilst the learning parameters remained the same. This resulted in the last day being one where all operator agents were healthy, and thus productivity was at an optimum.

The average throughput of the experiment, as seen in Table 5.4, was 8848 activities which is significantly lower than the DCDS experiment completed in Chapter 4 (the DCDS model runs). As with the other experiments, a statistical test was conducted and the results indicated that output results of Experiment 4 is statistically different from the DCDS experiment . Figure 5.9 shows a comparison of the Experiment 4 and DCDS box plots, indicating this difference in system output.

This is interesting because it shows that the system is sensitive to deterioration, even if it occurs over a shorter period of time. Figure 5.8 compares the box plot graphs of all four of the experiments, and it indicates that Experiment 4 yield higher productivity levels than Experiment 1 and 3.

Figure 5.10 shows a comparison of the average despatch zone utilisation. Experiment 4 had an average utilisation level of 84% which indicates that the system is also sensitive to deterioration that occurs at the beginning of the simulation run because the downstream parts of the model are starved of work and are thus less productive

Measure	Output Data of Experiment 4	Output Data of Steady State DCDS
Average	8848	12495
Standard Deviation	17.64	47.20

Table 5.4 Summary of output results of OAB - Experiment 4 compared to DCDS run

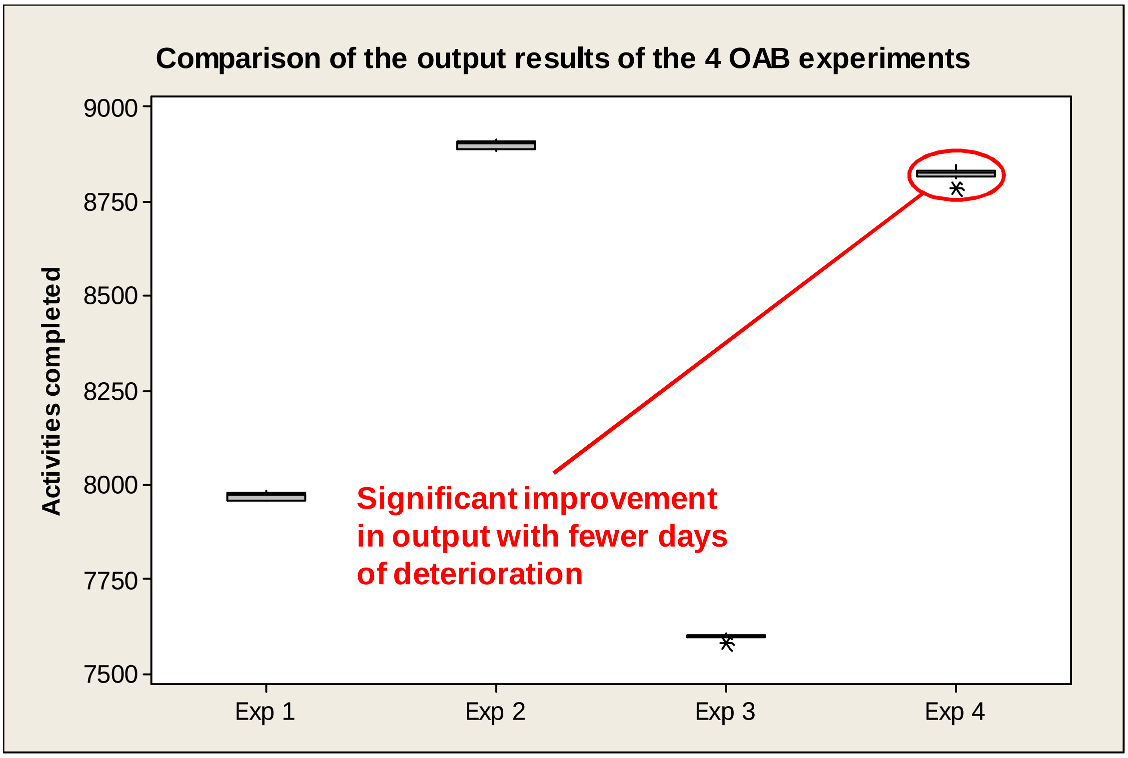

Figure 5.8: Box plot showing comparison of the 4 OAB experiments

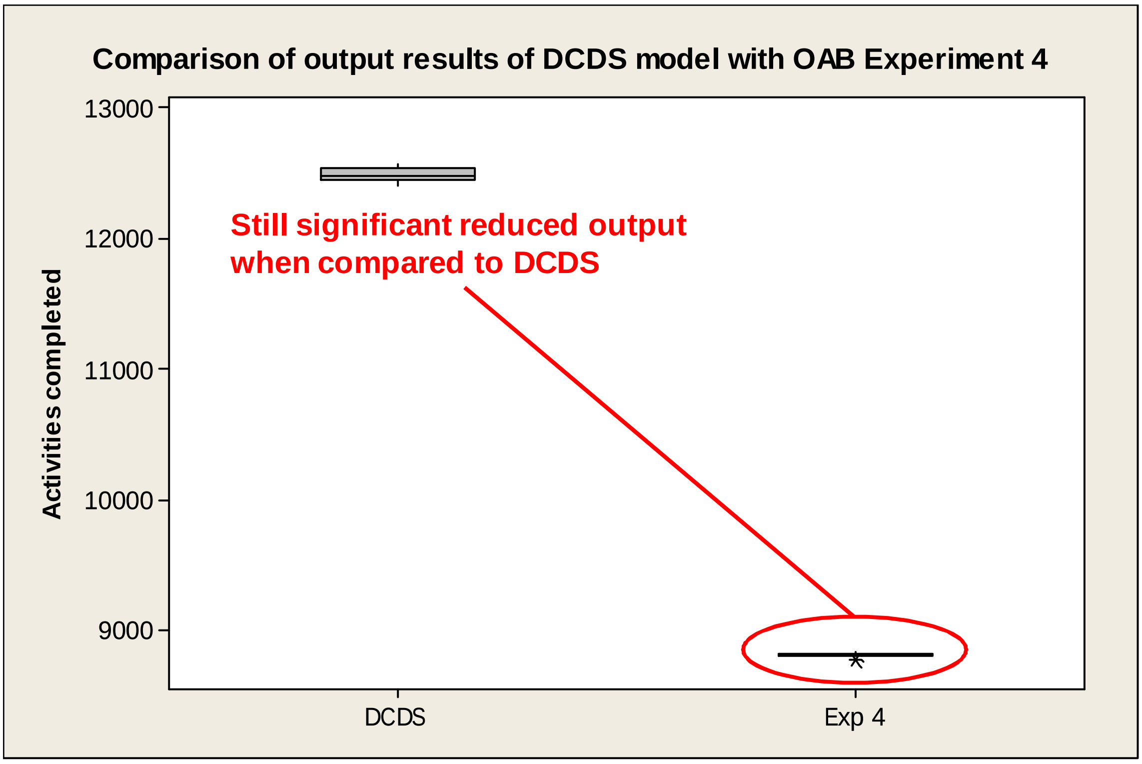

Figure 5.9: Box plot showing a comparison of output results between the DCDS and OAB - Experiment 4

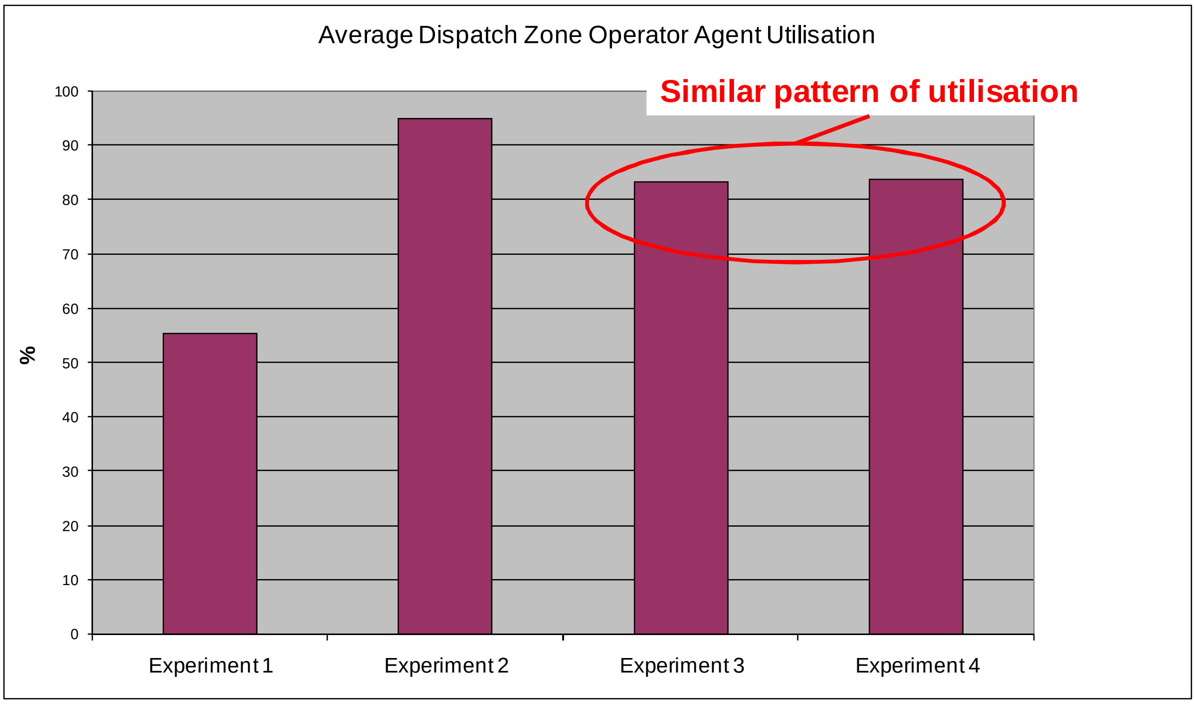

Figure 5.10: Comparison of the despatch zone operator agent utilisation for the four OAB experiments

5.9. Model Conclusion

The question posed at the beginning of this work was; "What effect does employee deterioration have on system productivity?"

These experiments show how deterioration and learning cycles typically experienced in a real system due to employee turnover can affect system productivity. The magnitude of this effect for this system is related to:

- The number of operator agents experiencing deterioration / learning,
- The length of the deterioration / learning cycle, and
- Where in the system it occurs.

In extreme cases where all employees experience this cycle, productivity can be affected by up to 36%.

It is important to note that the focus of this work is to develop a model to test the effects of deterioration and learning, but not necessarily to demonstrate its effect under various scenarios of employee turnover. The model that has been created and explained in this chapter provides an acceptable platform to perform these tasks.

The final question that requires attention is the issue of whether "management interaction" with the deteriorating / learning entities will significantly affect system productivity. The final component of the hybrid model is covered in Chapter 6.

Chapter 6

The Manager Agent Interaction Model

6.1. Introduction

The final component of the hybrid model allows for one or more manager agents to enter the simulation environment. These agents then interact with the operator agents and influence their behaviour. The goal is to assess whether one or more manager agents interacting in the simulation environment has an effect on the systems productivity.

6.2. Model conceptualisation

Since the main purpose of this model was to simulate the manager agent's roaming and interaction within the simulation environment over time, the key conceptual issues were:

- Manager agent decision making – which operator agent to interact with
- Interaction effect – what reaction or stimuli would be realised from the interaction
- Model synchronisation – how will this model synchronise with the rest of the hybrid model

Figure 6.1 shows how three the models "fit" together to form the hybrid model.

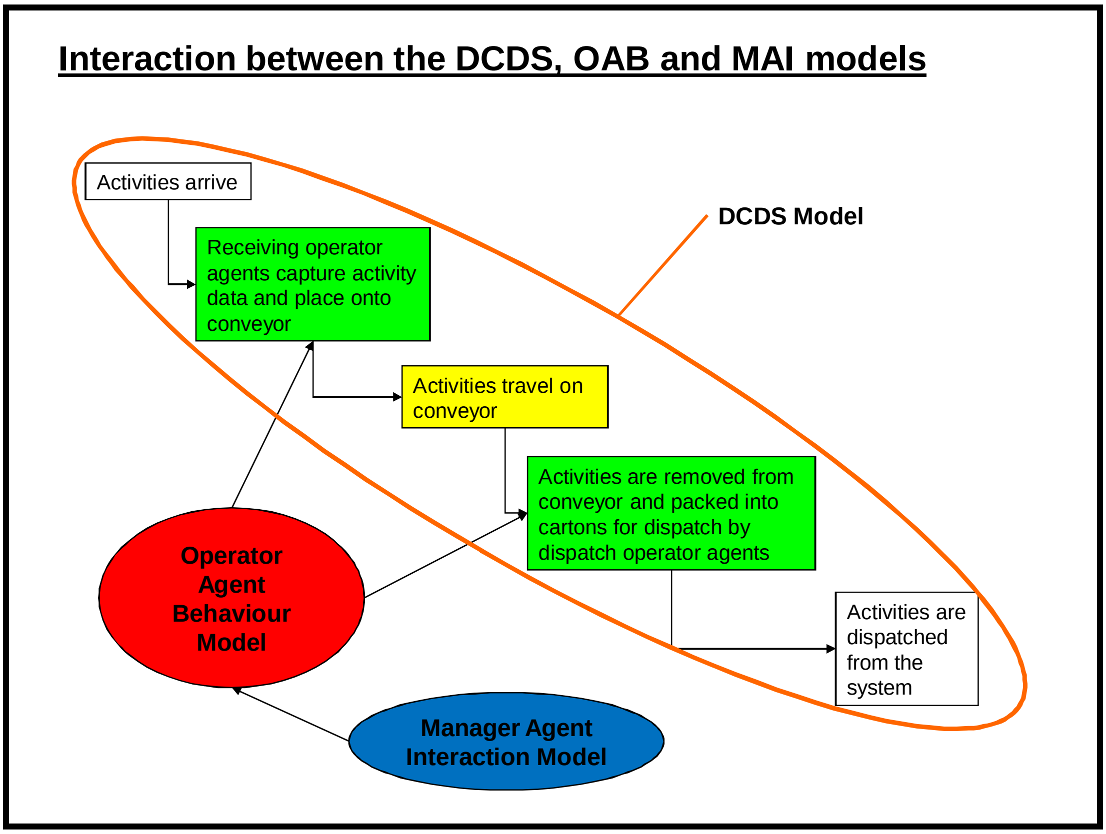

Figure 6.1: Interaction between the DCDS, OAB, and MAI models

6.3. Agent decision making

A key component of the model is the logic that is used to move the manager agent through the simulation environment. In Chapter 3, the concept of "bidding" was introduced. Agent bidding is a communication mechanism used by specific agents in an agent based simulation to attract other agents.

The initial concept with this model was to get the operator agents to use a bidding mechanism to attract the manager agent to interact with them. Initially, the criteria were that operator agents within the OAB that were experiencing the most deterioration would be the first to interact with the manager agent. This draws similarity with the real world where underperforming people receive more attention from management than those working at an acceptable level.

This method did not sufficiently translate the complete reality into the simulation model because within the real system the manager's movement between operators is random. Thus a system needed to be developed whereby the manager's movement could be both randomised, and then controlled for experimental purposes.

For information gathering purposes, the manager from the real system was interviewed and his cycle of on-the-job activities observed. It was noted that the manager is occupied with one of three different tasks at any one time, these being:

6.3.1. Meetings

- Morning meeting with other managers
- Crisis meetings
- Disciplinary meetings
- Problem solving meetings with senior management
- Meeting with service providers and vendors
- Meeting with customers

6.3.2. Supervising

- Motivating workers
- Physically assisting workers to complete their duties

6.3.3. Administration

- Reconciling production targets
- Preparing monthly reports
- Preparing performance measurement figures
- Capturing data into the performance measurement system

He indicated that he spent between 15 and 45 minutes conducting meetings, supervising, or completing administrative tasks per event, and that he completed about 14 tasks on average per day.

His typical work day is unstructured due to the reactive nature of the work environment and he is expected to deal with operational problems that occur without prior warning. The only fixed parts of his day are in the morning when he attends a management meeting, and in the afternoon when he tallies the production scores for the day and inputs them into a performance measurement software system. Figure 6.2 shows a summary of the logic flow that governs the operator agent's state change per activity step, with the colour of the logic step linked to the relevant model in Figure 6.1.

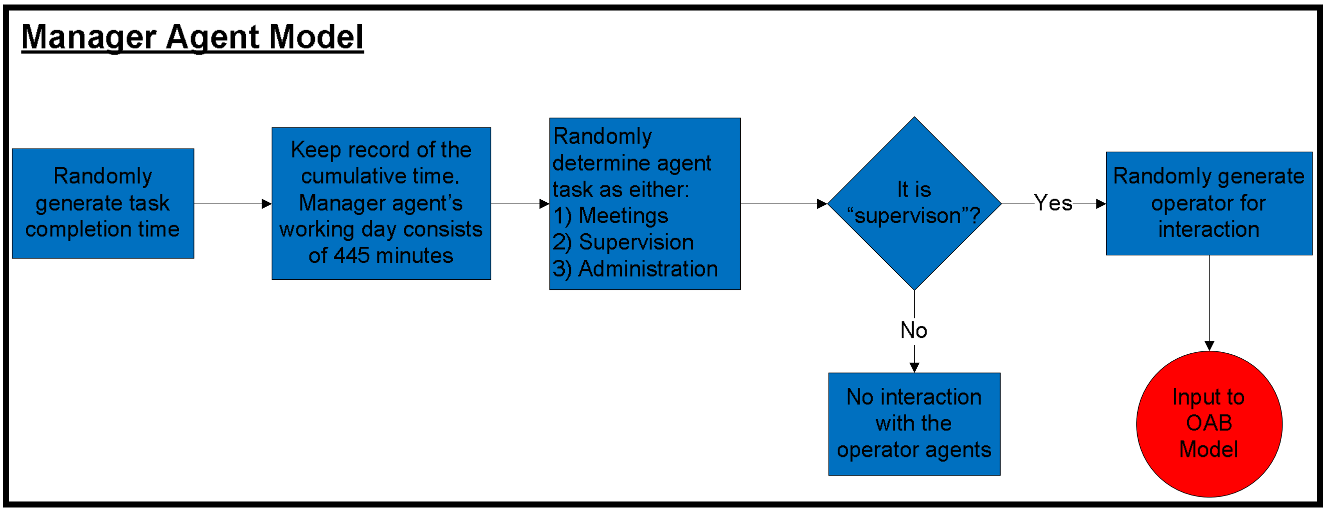

Figure 6.2: Manager Agent Interaction Model logic flow

6.4. Interaction effect

Once the manager agent's movements in the simulation environment have been decided, then the logic that determines what happens during interaction can take effect. Figure 6.3 shows the logic flow in the model, with the colour of the activity linked to the relevant model in Figure 6.1.

When a manager agent interacts with the operator agent, they must first determine whether they are a deteriorating or a new agent because this will influence the effect that the interaction will have on the operator agent's productivity. Once the operator agent's state has been determined, then the manager agent must recognise whether that agent has previously interacted with this particular operator agent during the simulation run. The result of this interrogation will also leverage an effect on the operator agent's productivity.

When discussing operator interaction with the real system manager, he indicated from his experience that a sub-normal operator would experience approximately a 25% improvement in performance through his interaction with them. This would be due to him either physically assisting the operator to complete his / her tasks, or through verbal motivation, or through on-the-job coaching for new operators.

When interacting with the sub-normal operators, the manager explained that his effect was temporary, and the operator would usually continue to perform his / her duties at a sub-normal or deteriorating level shortly after he had left them.

Operators who were new and unfamiliar with the operation showed an increase in skill after having interacted with the manager. The effect on their productivity was also approximately 25%, but the improvement was permanent. This was compounded if the manager interacted more than once with the same operator.

Within the MAI model, the effect that the manager agent has on operator agent is a variable and can be adjusted for management purposes.

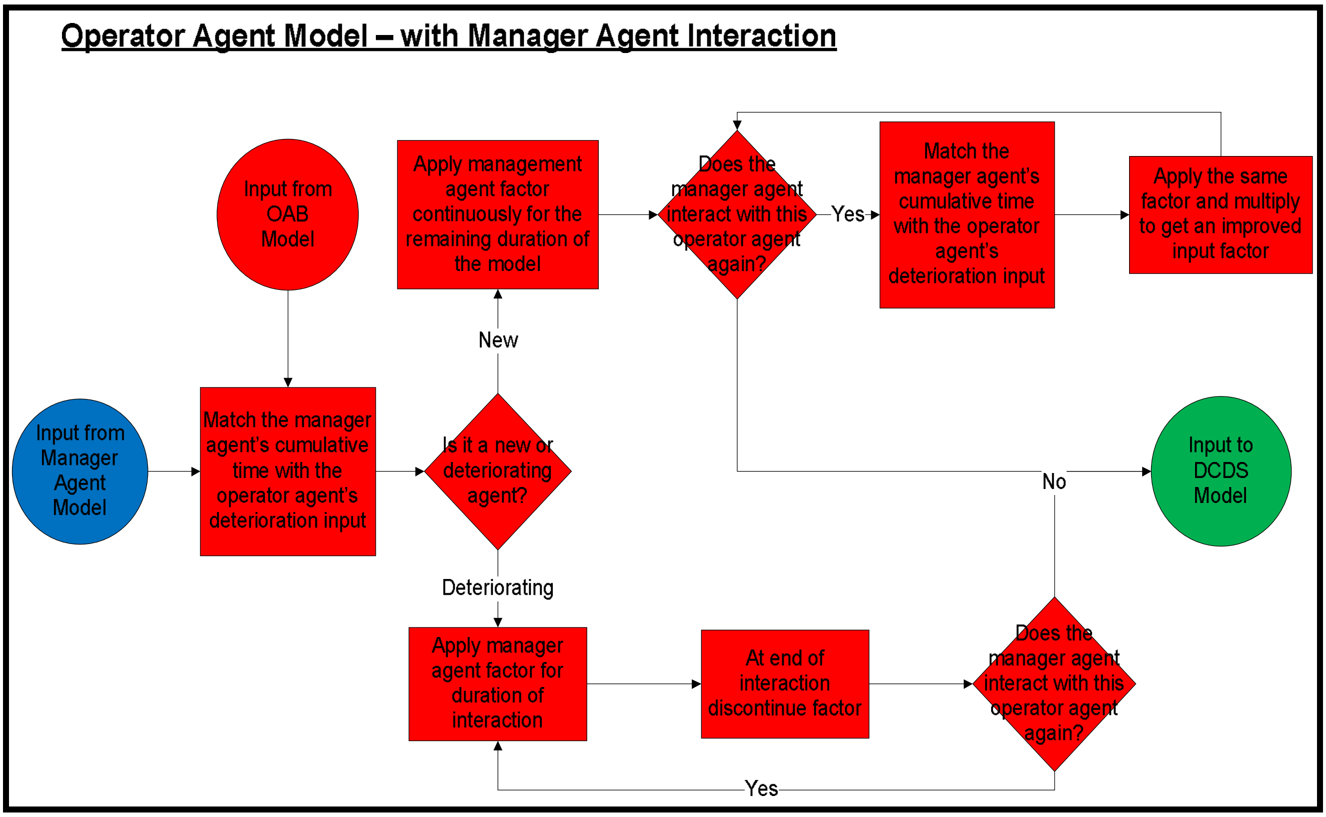

Figure 6.3: Interaction effect logic flow within the MAI model

6.5. Model synchronisation

As discussed consistently through this work, a key part of the hybrid model's success is the synchronisation of the three independent models together to form the single simulation entity.

Figure 6.2 illustrates how the MAI model logic flows into the OAB model, and Figure 6.3 shows how the OAB model logic accommodates the MAI model logic to create a new input to the DCDS model. Both the MAI and OAB models are created using spreadsheets and random number generators, and are thus relatively simple to construct using easy available software tools.

The rest of this chapter discusses how the logic described above has been transformed into a working model, and then the experiments conducted to test the hybrid model.

6.6. Model development

As discussed in the beginning of this chapter, the MAI model is developed separately and then integrated with the OAB model. Thus, the OAB model spreadsheets need to be modified to accommodate the instructions coming from the MAI model spreadsheet.

6.7. The manager agent model

Table 6.1 shows an example of the MAI model spreadsheet. Each day lasts 445 minutes, as per the DCDS and OAB models, which ensures synchronicity between the three models.

The spreadsheet consists of 5 columns:

6.7.1. Day (A)

The day of the week that the manager agent is currently working in is displayed here.

6.7.2. Task time (B)

This is the list of times to complete one task. The triangular distribution was chosen to be the best fit for the time taken per task (refer to Chapter 5 for justification for using this statistical distribution). Crystal Ball was used to generate the random streams that were assigned to the tasks identified.

6.7.3. Cumulative time (C)

The manager agent has a limit of 445 minutes of working time per day. This column keeps track of the time they have spent completing the individual tasks. Once 445 minutes of cumulative time has been reached, then a new day is started.

6.7.4. Task list (D)

This column indicates the tasks that the manager agent is going to complete during the day, and in what order. During the data gathering phase, it was noticed that there is an equal chance of any of the three types of tasks occurring (except at the beginning and end of the day, which are fixed tasks). Thus the uniform distribution was used to generate a random number stream in Minitab. The key is: (1) is for meetings, (2) for supervision, and (3) for administration

6.7.5. Operator agent number (E)

When a manager agent performs a supervisory task, an operator is identified as the target for the manager agent's attention. This was done by indexing the 20 operator agents in the model and assigning them a value from 1 to 20. Using uniform distribution, a random number stream was generated in Minitab and used.

A	B	C	D	E
Day	**Task Time**	**Cumulative Time**	**Task Type**	**Operator Agent Number**
	39.68149		1	
	28.42113	68.103	2	
	32.62645	100.729	3	18
	28.09355	128.823	2	
	31.26308	160.086	2	
	26.35597	186.442	3	11
	34.33551	220.777	3	15
	26.03086	246.808	2	
	23.4058	270.214	1	
	41.15945	311.373	3	21
	25.47063	336.844	1	
	35.4374	372.281	2	
	29.69085	401.972	2	
1	39.92157	441.894	2	

Table 6.1: Example of Day 1 of the Manager Agent's Virtual Day

6.8. The interaction effect factor

When the manager agent model indicates that interaction is due to occur, then a factor is applied to the operator agent of column E (refer to Table 6.1) in the input to the OAB model (refer to Chapter 5, section 3). Thus this factor is now a variable that can be manipulated for experimentation purposes.

This influencing factor was developed from information gathered from interviews with the manager (as discussed earlier in this chapter). The following model logic was the result:

6.8.1. When the manager agent interacts with a deteriorating operator agent, then the input parameter would be multiplied by a factor of 0.75 (which is 100% - 25%). This factor would continue to be applied whilst the manager agent was interacting with the operator agent, but would cease once the manager agent moved on to the next task.

6.8.2. When the manager agent interacts with a new operator agent, the interaction effect is permanent. The factor of 0.75 is applied to the input parameter, but continues to be applied even after the manager agent has moved on to the next task. If the manager returns to the same new operator agent, then the factor is further multiplied by 0.75 to indicate a steeper learning curve.

Table 6.2 demonstrates how the factor changes with multiple manager agent interactions, and Figure 6.4 shows how the manager agent factor may be applied to a typical operator agent over the simulation period. Note how the multiple interactions during the latter part of the simulation change the interaction factor applied to the operator agent.

No of manager agent interactions	Management influence factor
1	0.75
2	0.5625
3	0.4219
4	0.3164
5	0.2373

Table 6.2: The manager influencing factor when applied to a new operator agent

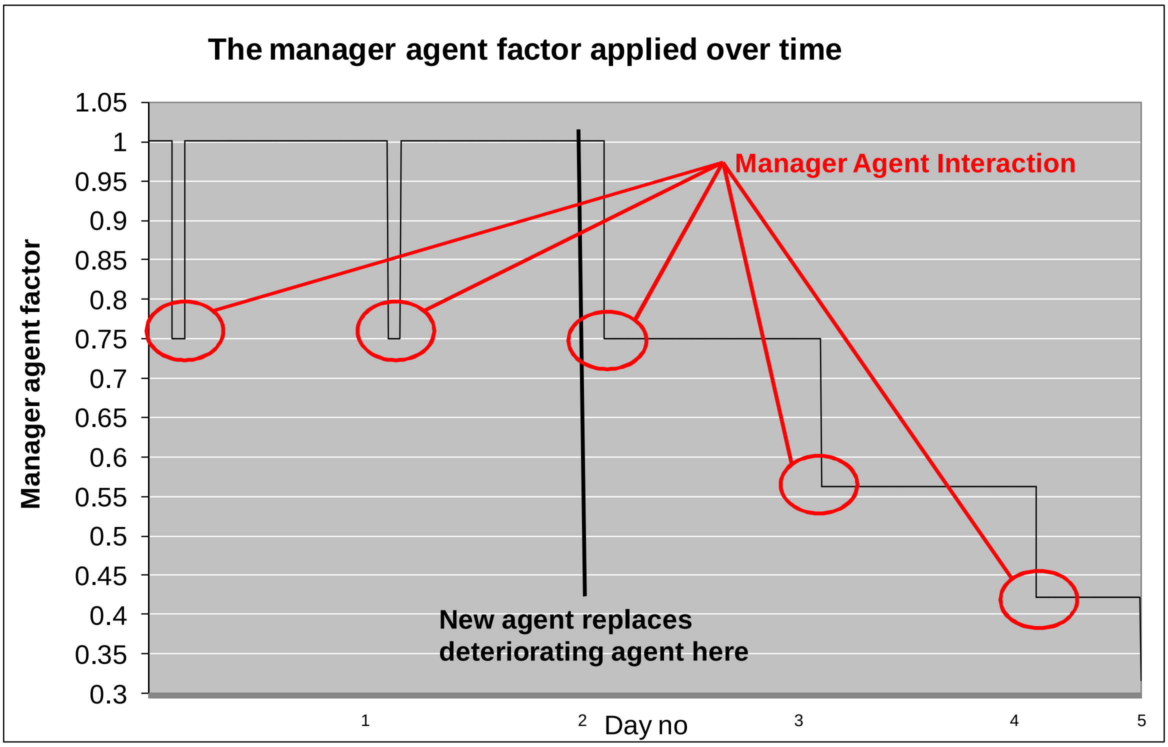

Figure 6.4: An example of the manager agent factor applied to an operator agent over time

6.9. Synchronicity between the MAI and OAB models

The interaction between the OAB model and the DCDS model as discussed in Chapter 5 was relatively simple due to the fact that Simul8 allows for a certain degree of compatibility with MS Excel. The interaction between the MAI and the OAB models proved to be more challenging because the compatibility mechanisms had to be developed.

It was decided that an effective agent interaction model run time should match one production week (5 days of 445 minutes) and this had to be synchronised with the OAB spreadsheet. The one item that both the MAI and the OAB models have in common is cumulative time. Both track the progress of time through the working day, and both have a limit of 445 minutes, thus this variable was used to synchronise the two models.

The key to the interaction between the two models was to find the common point (cumulative time) during the day that a particular operator agent and the manager agent were scheduled to interact, and then to use output data from the MAI model to influence the OAB model. This then creates an input that influences the relevant individual operator agent in the DCDS model.

For example; Table 6.1 shows the manager agent's activity plan for the first day of a random simulation period and indicates how they interacted with four operator agents. The first of which was operator agent 18, which received just over 32 minutes of attention from the manager agent. Column B (cumulative time) indicates that this interaction took place from 68 minutes to 100 minutes into that day.

Table 6.3 shows the model input spreadsheet for operator agent 18 (as used in the OAB model), with column B tracking cumulative time. In order to synchronise the two models together, a common cumulative time is found.

In this example the closest common time between the manager agent model and the operator agent model is 67.3766 minutes, and this is the first instance where the manager influencing factor is applied. This factor is then applied for 32 minutes of cumulative time (as per Table 6.1 column B). This particular operator agent is experiencing deterioration at this time point because the interaction falls within the first two days of the operator agent's simulation period. Thus the manager influencing factor will stop being applied once the manager agent's interaction has been completed.

This process is repeated each time the manager agent interacts with an operator agent.

Operator Agent 18 – Input number stream to the DCDS model					
A	**B**	**C**	**D**	**E**	**F**
Day Number	**Cumulative Total**	**Process Time**	**Deterioration / Learning Ratio**	**Mgt Influence Factor**	**Input Variable**
1	61.46105	2.56442	1.284876	1	3.294961
1	63.78826	2.32721	1.415647	1	3.294507
1	65.28781	1.49955	1.193988	1	1.790445
1	67.3766	2.08879	1.393603	0.75	2.183209
1	69.80509	2.42849	1.400404	0.75	2.55065
1	72.31768	2.51259	1.451937	0.75	2.736091
1	73.27121	0.95353	1.312082	0.75	0.938332
1	76.42078	3.14957	1.022076	0.75	2.414325
1	78.36527	1.94449	1.222454	0.75	1.782787
1	80.03535	1.67008	1.334277	0.75	1.671263
1	81.86139	1.82604	1.454691	0.75	1.992242
1	83.13729	1.2759	1.376602	0.75	1.317305
1	84.93113	1.79384	1.412249	0.75	1.900011

Table 6.3: Example of the synchronisation of the manager agent interaction model and the operator agent behaviour model

6.10. Model verification and validation

Due to the lack of empirical data, it is difficult to test this model. An attempt was made to at least test that the results of a full run were reasonably consistent with what was expected. The model was run using five different management interaction scenarios, and the results were shown to the manager of the department. He agreed that these results were in line with his assumptions, and thus it can be said that in his opinion the model appeared sufficiently valid for experimental scenarios.

6.11. Model experiments

6.11.1. Objective of the experiments

In Chapter 5 it was shown that the deterioration of human beings working within a production environment can result in a loss of productivity to the system. This conclusion was possible due to the development of a model that could simulate the state change of the operator agents over the designated time period.

The experiments in this chapter are focused on exploring whether manager agent interactions have an effect on the productivity of the operator agents during their deterioration and learning cycles.

The experiments conducted in Chapter 5 showed to what extent the production system was sensitive to deterioration / learning experienced in different parts of the model. This was shown by the differences in system output when the receiving zone was affected versus the despatch zone. The experiments in this chapter take cognisance of this sensitivity, and the experiments are orientated around these two areas.

Initially the model was tested in a series of six experiments that aimed to determine how sensitive the system productivity was to one manager agent interacting with operator agents at different points in the system for varying amounts of time. The aim was to see whether management interaction within the existing system's setup of 20 operators and one manager would result in a significant difference in output of activities.

The first set of experiments was focused on the receiving zone and tested the system's productivity when different levels of manager agent interaction were experienced. The manager agent was only permitted to interact with operator agents within the receiving zone. The experiments are organised in the following way:

- Experiment 1: 25% of the manager agent's day is spent interacting
- Experiment 2: 50% of the manager agent's day is spent interacting
- Experiment 3: 75% of the manager agent's day is spent interacting

The next set of experiments focused on the despatch zone, with the manager agent limited to interaction here only. The experiments are organised in the following way:

- Experiment 4: 25% of the manager agent's day is spent interacting
- Experiment 5: 50% of the manager agent's day is spent interacting
- Experiment 6: 75% of the manager agent's day is spent interacting

There results of both sets of experiments were then analysed in preparation for further model experimentation.

The next batch of testing focused on whether the number of manager agents in the model would leverage an effect on system productivity. In these experiments a further four manager agents were added to the system and then allowed to interact with the operator agents at the main areas in the model.

The aim of this situation was to determine whether a productivity problem could be solved simply by adding resources to this system, and if so how many resources would be required to negate the effect of operator deterioration and learning. The experiments are organised as follows:

- Experiment 7: Five manager agents spend 50% of their working day interacting with the receiving zone only
- Experiment 8: Five manager agents spend 50% of their working day interacting with the despatch zone only.

The final experiment involved five manager agents interacting randomly within the modelling environment for the entire simulation period. The aim here was to determine whether extreme measures of having a significant amount of extra supervisory staff roaming the work environment would have an effect on the system output. The details of the experiment are as follows:

- Experiment 9: Five manager agents spend 100% of their working day interacting with the despatch and receiving zones. Five manager agents interact with the receiving zone whilst the other five interact with the despatch zone.

6.11.2. Model setup

The MAI and OAB models are closely linked because they work together to create the inputs to the DCDS model which is then used as the platform for the hybrid model experiments.

Thus the model setup is exactly the same for the MAI as the OAB. Each simulation replication consists of the equivalent of five working days (a working week) of 445 minutes of work time per day, which equates to 2225 minutes in total.

In all the experiments in this chapter every operator agent in both the receiving and despatch zones initially experiences 890 minutes of deterioration (the equivalent of 2 days) from the start of the simulation run at the same rate discussed in Chapter 5 (section 4.3). At the end of this period, the deteriorated operator agent is replaced by a new operator agent who experiences learning for 1112 minutes (the equivalent of 2.5 days) at the rate discussed in Chapter 5.

The MAI model will control the manager agent’s movements within the model space according the criteria defined by the experimental parameters.

6.11.3. Experiments 1 – 3: Manager agent interaction in receiving zone only

A summary of the results of the experiments are found in Table 6.4 and Table 6.5. Of interest is the relative similarity in output of the three different experiments, which shows that one manager agent interacting in the receiving zone is not likely to appreciably raise the productivity levels of the system from its worst case scenario (Table 6.5).

In addition to this conclusion, the results shown in Table 6.4 indicate that the amount of time that the manager agent spends in the receiving zone interacting with operator agents is irrelevant as the system output only improves slightly. This is surprising given the large amount of time the manager agent spends interacting with operator agents in Experiment 3. If this scenario were to occur in the real system, the manager would be left with no time to conduct his other manager responsibilities, thus weakening the long term productivity of the organisation.

An Analysis of Variance statistical test was conducted to determine if the three experiments yielded output values from different populations, and are thus independent of each other. A p-value of less than 0.05 resulted, thus concluding that the experiment output results come from different populations although the differences are slight when compared to the baseline outputs without deterioration.

Figure 6.5 and 6.6 show a graphical summary of the conclusions outlined in these paragraphs.

Measure	Experiment 1: 25% of MA day spent interacting	Experiment 2: 50% of MA day spent interacting	Experiment 3: 75% of MA day spent interacting
Average	7714	7730	7771
Standard Deviation	10.98	15.49	14.71

Table 6.4: Summary of output results with manager agent interacting in receiving zone only

Measure	DCDS model run	OAB – all operator agents deteriorating / learning model run	Experiment 3: 75% of MA day spent interacting
Average	12495	7608	7771
Standard Deviation	47.20	6.99	14.71

Table 6.5: Summary of output results of experiment 3 compared with the DCDS and OAB (all deteriorating / learning) model runs

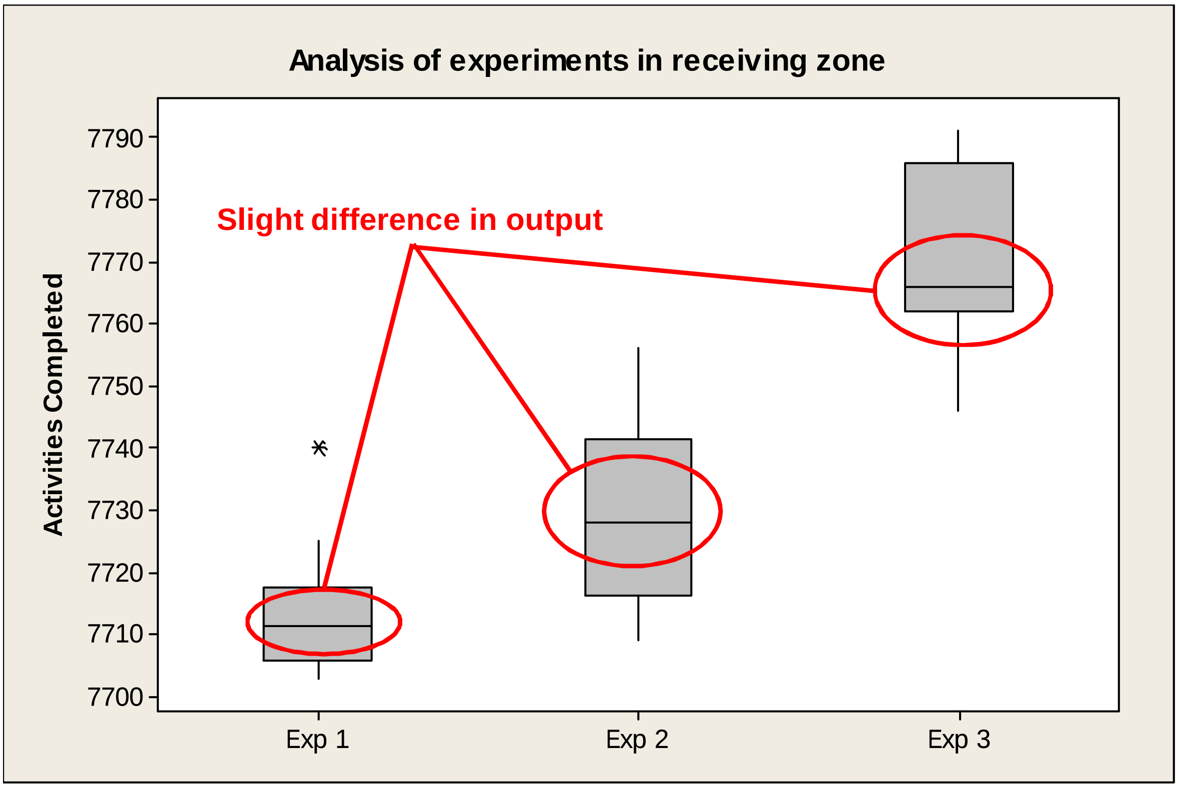

Figure 6.5: Summary of experiment with the manager agent interacting in receiving zone only

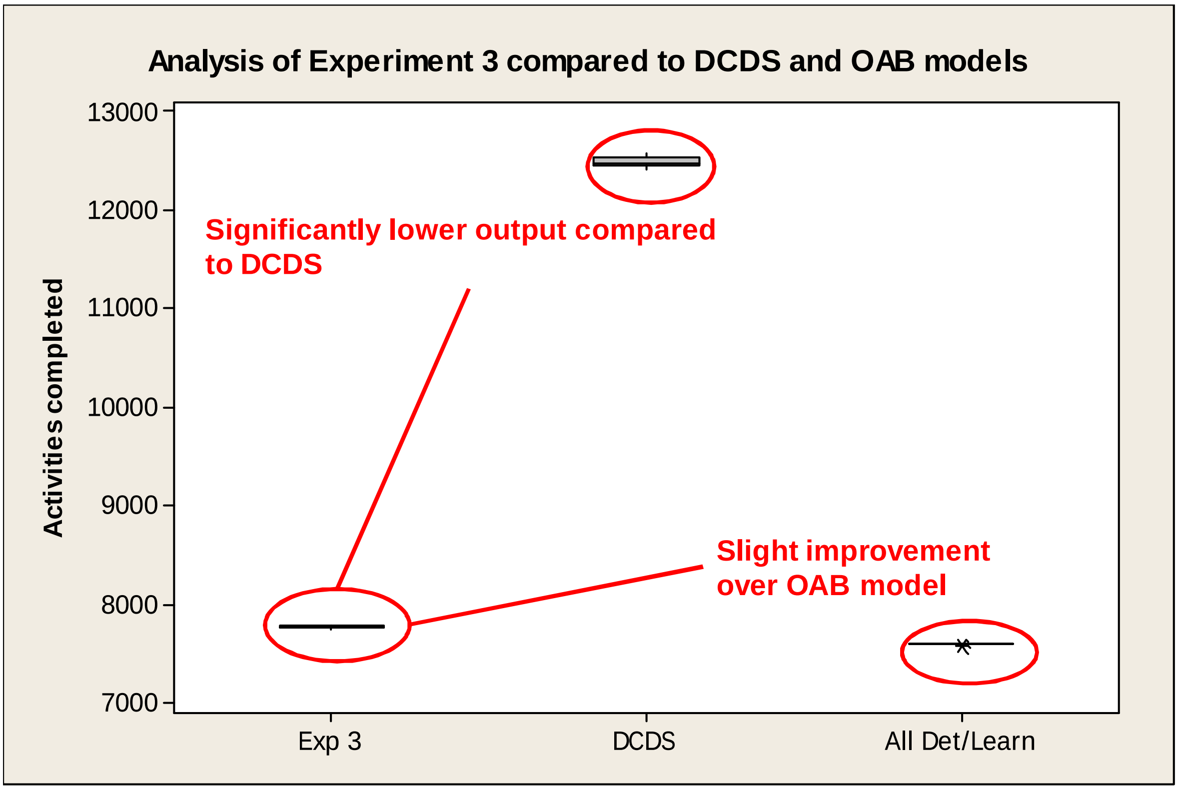

Figure 6.6: Comparison of the output results of Experiment 3 with the DCDS and OAB (all operator agents deteriorating / learning) models

6.11.4. Experiments 4 – 6: Manager agent interaction in despatch zone only

Of immediate interest with the results of these experiments is how similar the output results are to those in the first 3 experiments (refer to Figure 6.8). The results summarised in Table 6.6 and Table 6.7 show that a manager agent who interacts with operator agents in the despatch zone does improve system productivity by a small margin of 250 activities on average.

Figure 6.9 shows a comparison of Experiment 6 with the DCDS and OAB simulation experiments, and it is clearly indicated that Experiment 6's output is well below that of the DCDS, and slightly higher than the OAB simulation experiment. This confirms that one manager agent interacting with operator agents in this system has a negligible effect on overall system productivity.

An Analysis of Variance statistical test was conducted to determine if the three experiments yielded output values from different populations, and are thus independent of each other. A p-value of less than 0.05 resulted, thus concluding that the experiment output results come from different populations.

Figure 6.7 shows the box plots of the three experiments and again the differences are slight when compared to the baseline outputs without deterioration. Thus it can be concluded that the amount of time that a manager agent spends interacting with operator agents in the dispatch zone has a negligible effect on system productivity.

Measure	Experiment 4: 25% of MA day spent interacting	Experiment 5: 50% of MA day spent interacting	Experiment 6: 75% of MA day spent interacting
Average	7752	7773	7865
Standard Deviation	10.76	11.21	67.36

Table 6.6: Summary of output results with manager agent interacting in despatch zone only

Measure	DCDS model run	OAB – all operator agents deteriorating / learning model run	Experiment 3: 75% of MA day spent interacting
Average	12495	7608	7865
Standard Deviation	47.20	6.99	67.36

Table 6.7: Summary of output results of experiment 6 compared with the DCDS and OAB (all deteriorating / learning) model runs

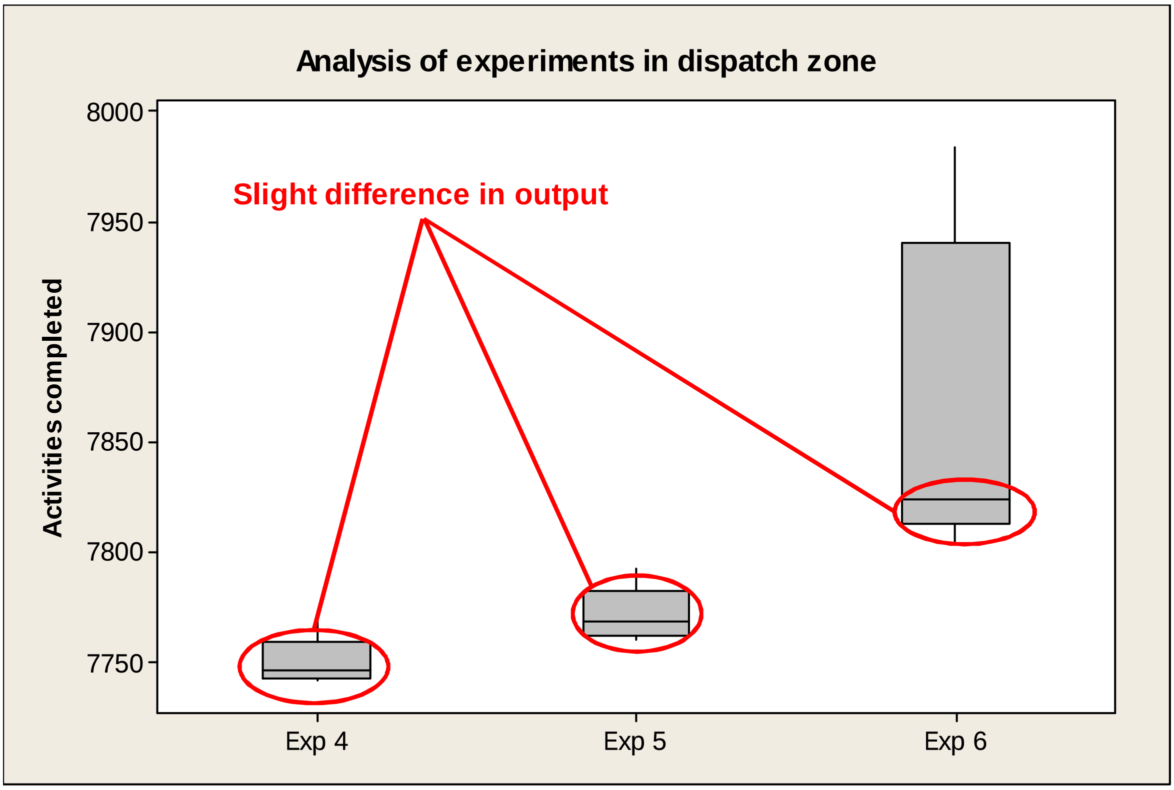

Figure 6.7: Summary of experiment with the manager agent interacting in despatch zone only

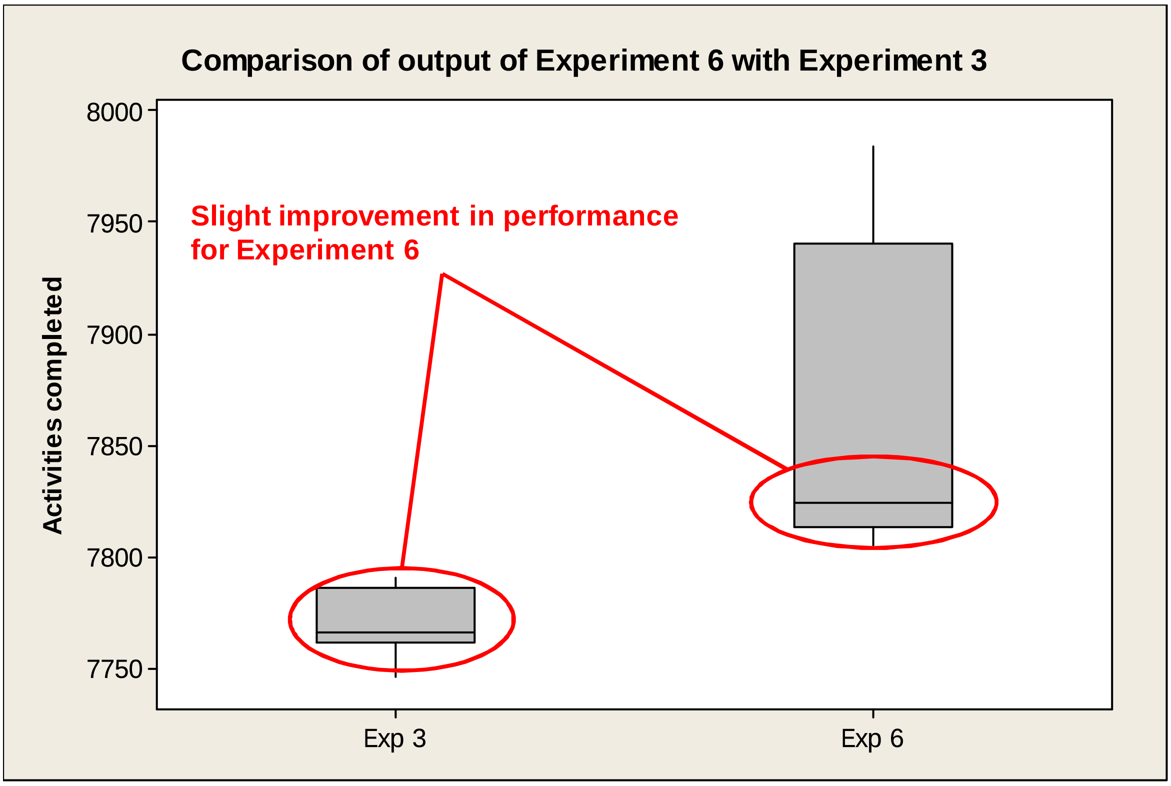

Figure 6.8: Comparison of the output results of Experiment 6 to Experiment 3

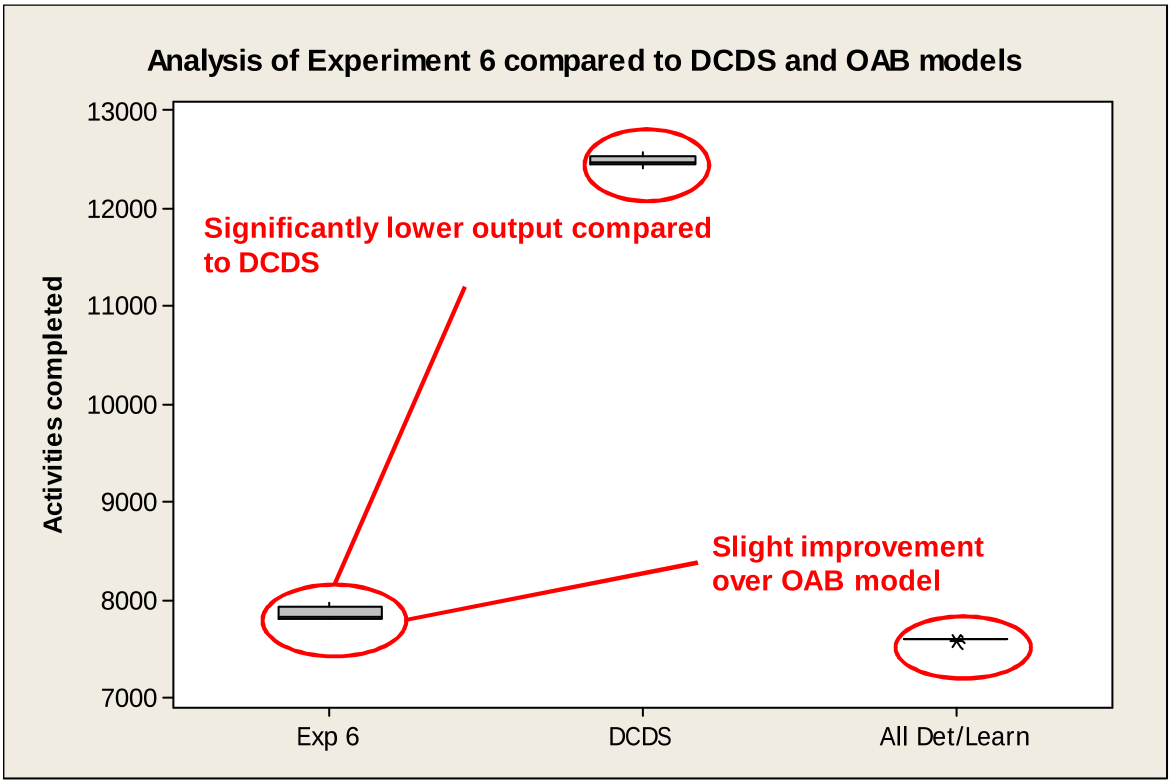

Figure 6.9: Comparison of the output results of Experiment 3 with the DCDS and OAB (all operator agents deteriorating / learning) models

It is important to note after the first six experiments that the model behaviour is consistent with human logic. In a complex system such as this one it is unrealistic to expect a single entity to fundamentally change the system dynamics. This means that the hybrid approach is proving to be a reliable method to highlighting the problems associated with the simulation of complex systems where humans interact with each other, and machines are used for much of the manual work.

Experiment 7 and Experiment 8 are structured to further test the robustness and accuracy of the hybrid model developed thus far.

6.11.5. Experiment 7 – Five Manager agents interact for 50% of their working day in the receiving zone only

The output results of this experiment are significantly higher than any of the other experiments thus far (Table 6.8), with an average of 11 057 activities completed during the simulation period. Figure 6.11 shows a graphical comparison of experiment 3 (highest activities completed for the first set of experiments), experiment 6 (highest activities completed for the next set of experiments), and experiment 7.

This improved performance can be attributed to the significantly higher throughput of activities through the receiving zone due to the interaction of the five manager agents with the operator agents there. More activities completed by the receiving zone operator agents translate into more work available for the despatch zone operator agents, which raises the overall system throughput of activities in spite of the lower levels of despatch zone operator agent productivity.

Another contributing factor is the speed of learning of the new operator agents due to abundance of manager agent mentors. The more interaction a new operator agent has with a manager agent, the faster they learn and reach healthy operator agent status.

When comparing system throughput to the DCDS and OAB experiments (Table 6.9) it can be noted that there is a significant improvement in activities completed compared to the OAB experiments. Whilst the five extra manager agents do not allow the system to reach optimum productivity as per the DCDS experiments, there is a satisfactory improvement. Figure 6.10 shows a box plot comparison of experiment 7 to the DCDS and OAB experiments.

There can be no doubt that a significant improvement can be realised from the addition of the five manager agents who spend half of their day interacting with operator agents in the receiving zone.

Measure	Experiment 3	Experiment 6	Experiment 7
Average	7771	7865	11057
Standard Deviation	14.71	67.36	25.80

Table 6.8: Summary of output results of experiment 7 compared with experiment 3 and experiment 6

Measure	DCDS model run	OAB – all operator agents deteriorating / learning model run	Experiment 7
Average	12495	7608	11.57
Standard Deviation	47.20	6.99	25.80

Table 6.9: Summary of output results of experiment 7 compared with the DCDS and OAB (all deteriorating / learning) experiments

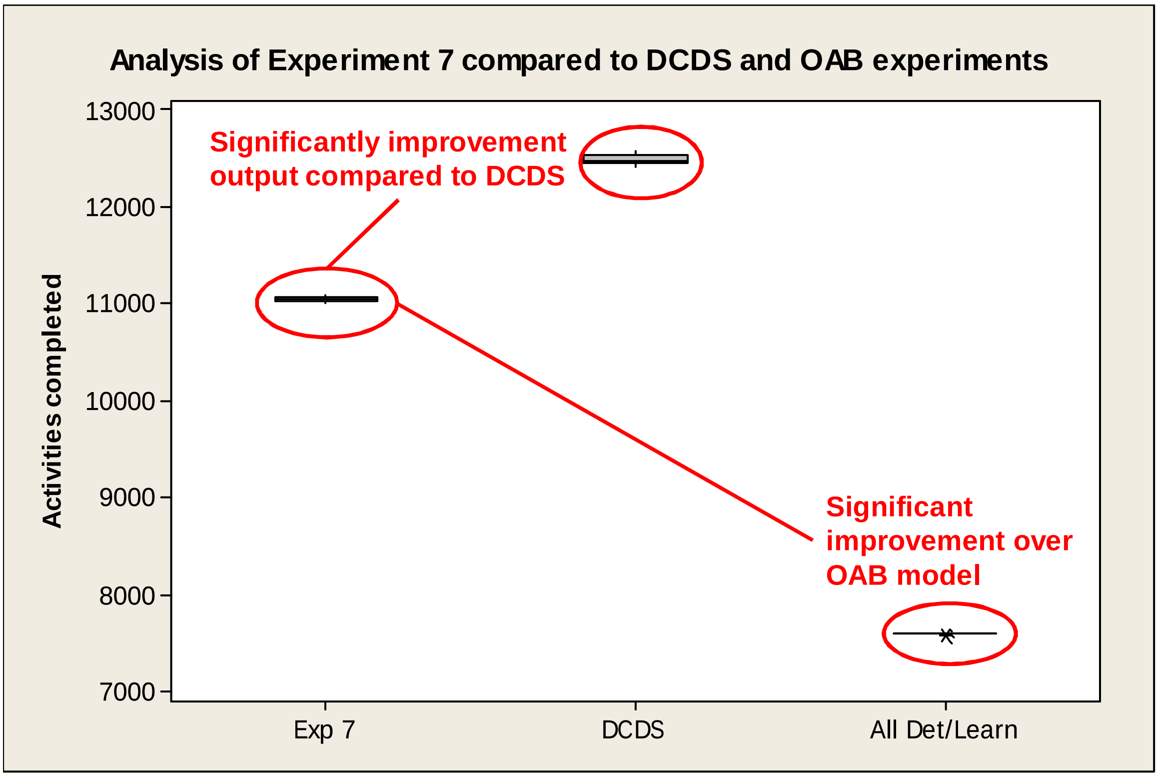

Figure 6.10: Comparison of the output results of experiment 7 with the DCDS and OAB experiments

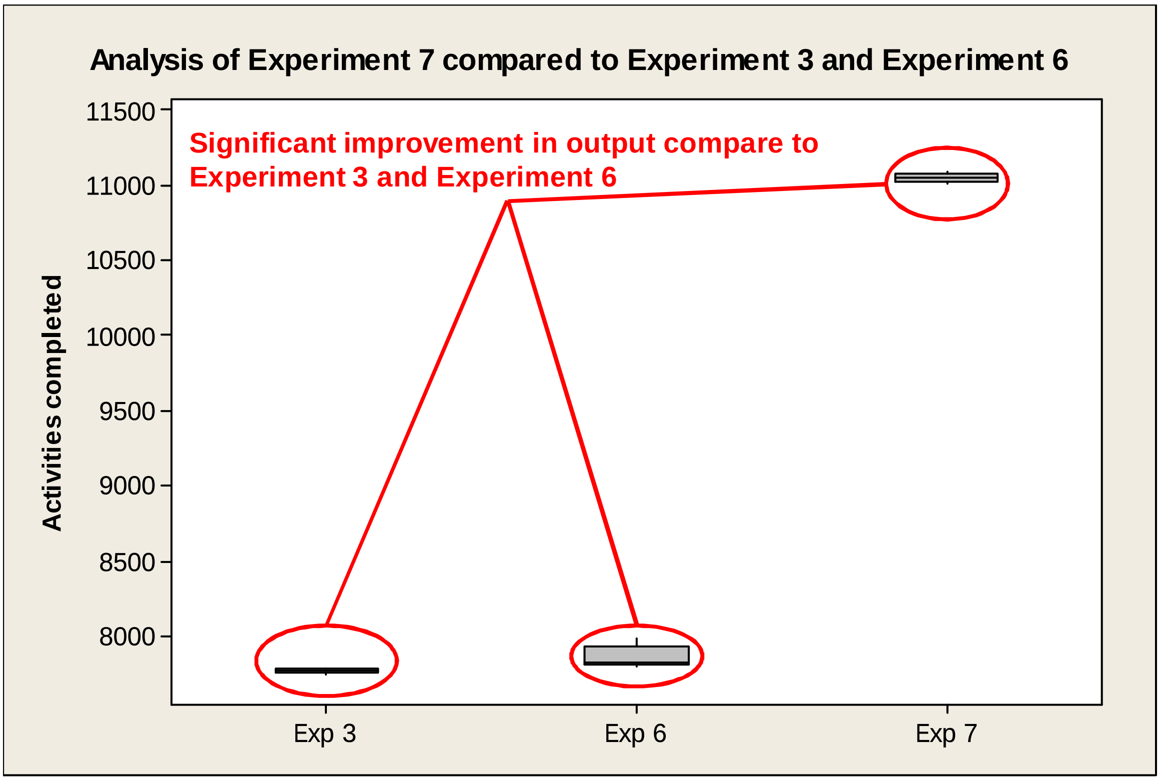

Figure 6.11: Comparison of the output results of Experiment 7 to Experiment 3 and Experiment 6

6.11.6. Experiment 8 – Five Manager agents interact for 50% of their working day in the despatch zone only

The results of experiment 8 are in contrast to those in experiment 7 because the output of activities in experiment 8 is significantly lower. Figure 6.12 shows a box plot of the two experiments and difference between the productivity of the two is obvious.

When compared to experiment 3 and experiment 6, it is clear that experiment 8 yielded slightly improved output results (Table 6.10) which indicates that the five manager agents interacting in the despatch zone did improve system productivity. However, the relatively small magnitude of improvement in spite of the extra resources allocated there indicates that the five manager agents proved to be ineffective interacting at the despatch zone only. Figure 6.14 shows a comparison of the output results of experiment 3, experiment 6, and experiment 8 using box plots.

This conclusion is validated when the output results of experiment 8 are compared with the DCDS and OAB experiments (Table 6.11). Whilst there is a marginal improvement in output when compared to the OAB experiments, the productivity of experiment 8 is far below that of the DCDS experiment. Figure 6.13 shows a comparison of box plots of experiment 8 with the DCDS and OAB experiment output results.

The low throughput of this experiment can be attributed to the low throughput of activities coming from the receiving zone due to the lower productivity of the operator agents there. The starvation of activities to the despatch zone resulted in lower amounts of activities for the operator agents, which negated the effectiveness of the manager agent's interactions with them.

Measure	Experiment 3	Experiment 6	Experiment 8
Average	7771	7865	7937
Standard Deviation	14.71	67.36	29.53

Table 6.10: Summary of output results of experiment 8 compared with experiment 3 and experiment 6

Measure	DCDS model run	OAB – all operator agents deteriorating / learning model run	Experiment 8
Average	12495	7608	7937
Standard Deviation	47.20	6.99	29.53

Table 6.11: Summary of output results of experiment 8 compared with the DCDS and OAB (all deteriorating / learning) experiments

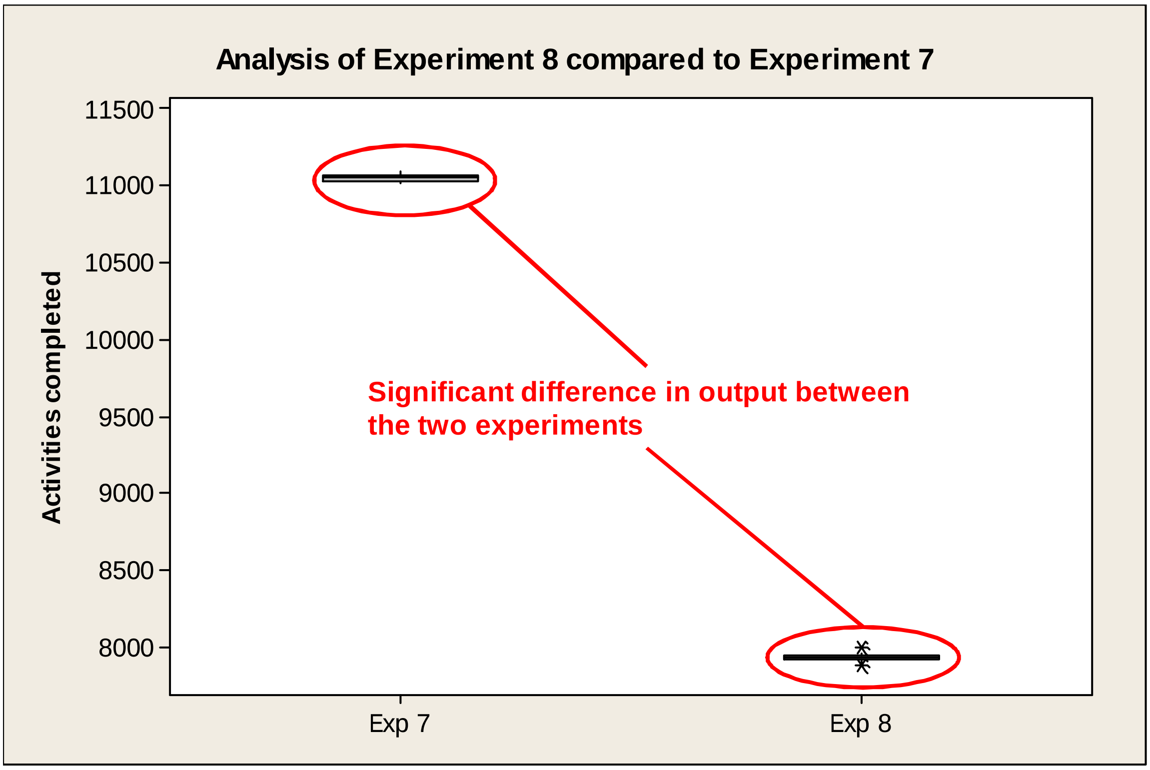

Figure 6.12: Comparison of the output results of experiment 8 to experiment 7

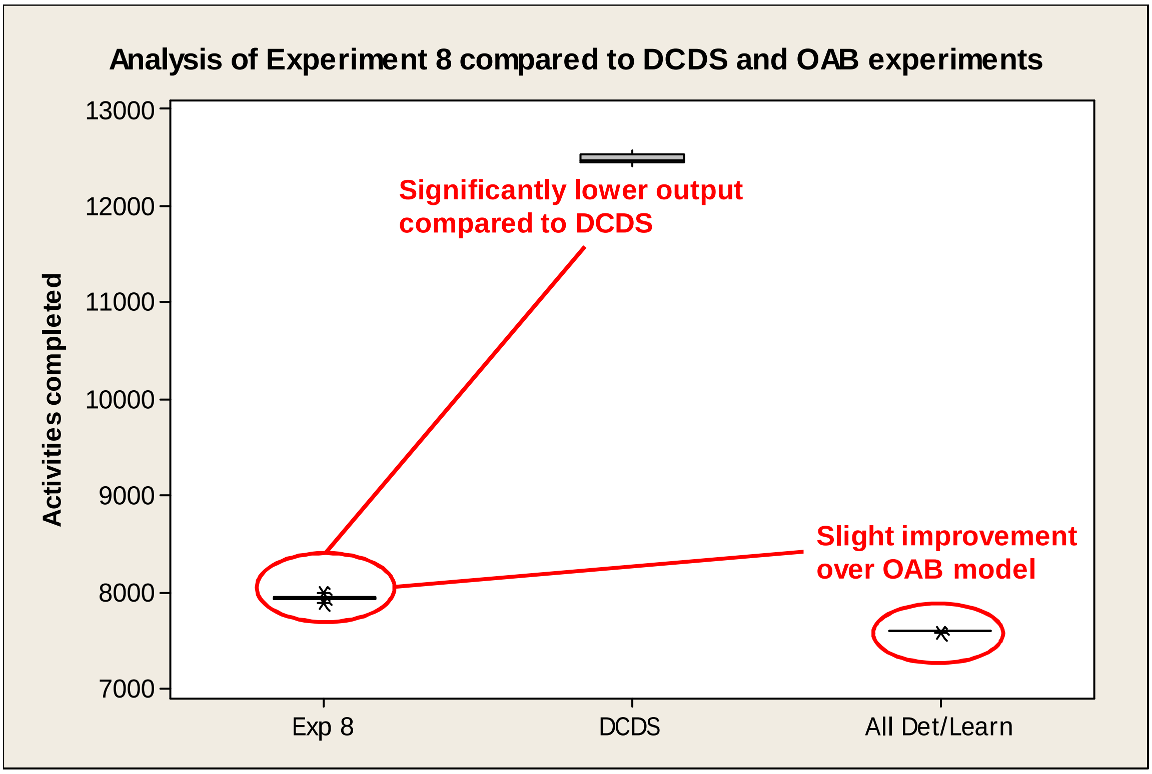

Figure 6.13: Comparison of the output results of Experiment 7 with the DCDS and OAB experiments

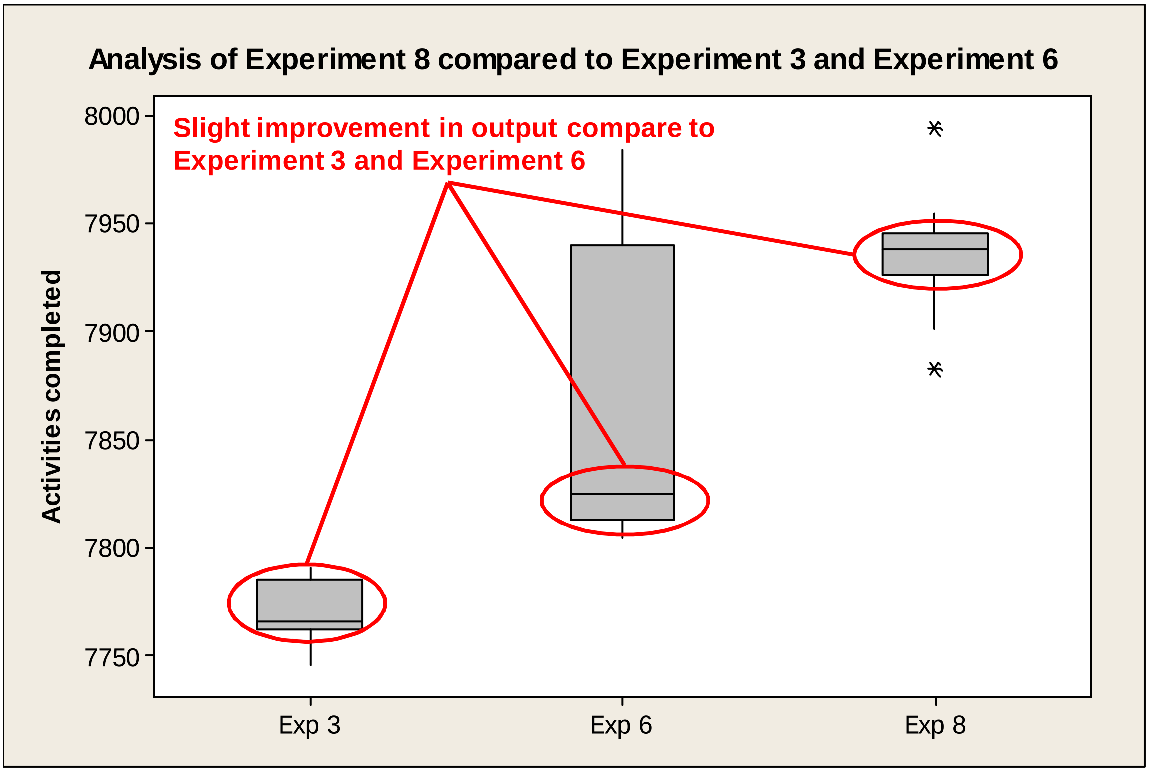

Figure 6.14: Comparison of the output results of Experiment 8 to Experiment 3 and Experiment 6

Analysis of the results of experiment 7 and experiment 8 also makes logical sense when considered within the parameters of the complex system being simulated. The improved performance of experiment 7 due to an improved flow of activities to the rest of the system, and the relatively poor performance of experiment 8 due to work starvation shows that the hybrid model is sensitive to different operating parameters. This further validates its effectiveness as a tool to test the system dynamics.

6.11.7. Experiment 9 – Five Manager agents interact for 100% of their working day throughout the simulation

The final experiment involves adding five full time manager agent resources to the simulation environment. The extra resources had a noticeable effect, with the number of activities completed significantly higher than either Experiment 3 or Experiment 6 (refer to Table 6.12)

This increase in output, as shown in Figure 6.16, can be attributed to the improved productivity of the receiving and despatch zones due to the manager agents' interaction with the operator agents in these areas.

When compared to the OAB and DCDS experiments (refer to Table 6.13 and Figure 6.15), it is noted that the output of Experiment 9 is just over 1000 activities below the DCDS output results. The extra manager agent resources thus make this situation the most productive experiment when compared to the ideal state presented by the DCDS experiment.

However, the practicality of adding five extra resources to a production system needs to be questioned. The extra cost incurred by having surplus management resources in the real system would probably negate the gains made in output.

Two Analysis of Variance (ANOVA) statistical tests were conducted here. The first to determine if Experiments 7, 8, and 9 yielded output values from different populations, and the second to determine if all the experiments in this chapter are independent of each other. The first ANOVA test yielded a p-value of less than 0.05 thus concluding that Experiments 7, 8, and 9 are independent of each other. The second ANOVA test yielded a p-value less than 0.05, indicating that the mean output results of all the experiments are independent of each other, and thus come from different populations.

Measure	**Experiment 3**	**Experiment 6**	**Experiment 9**
Average	7771	7865	11343
Standard Deviation	14.71	67.36	324.56

Table 6.12: Summary of output results of experiment 9 compared with experiment 3 and experiment 6

Measure	**DCDS model run**	**OAB – all operator agents deteriorating / learning model run**	**Experiment 9**
Average	12495	7608	11343
Standard Deviation	47.20	6.99	324.56

Table 6.13: Summary of output results of experiment 9 compared with the DCDS and OAB (all deteriorating / learning) experiments

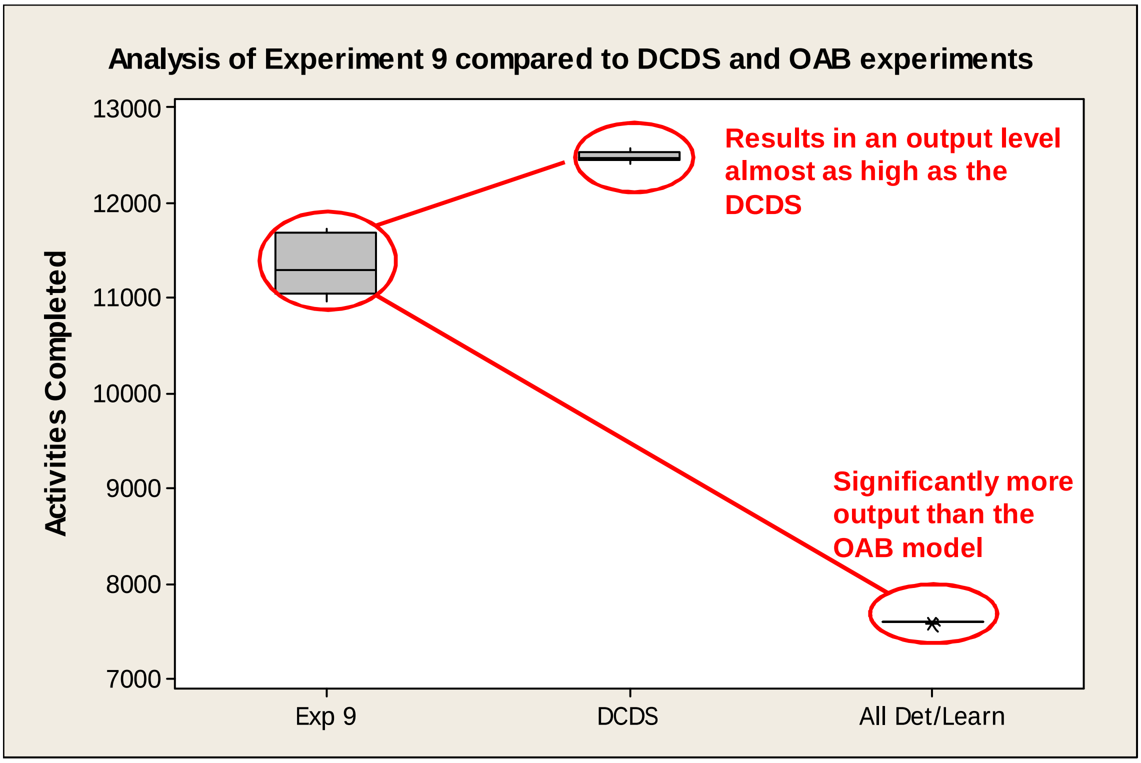

Figure 6.15: Comparison of the output results of Experiment 9 with the DCDS and OAB experiments

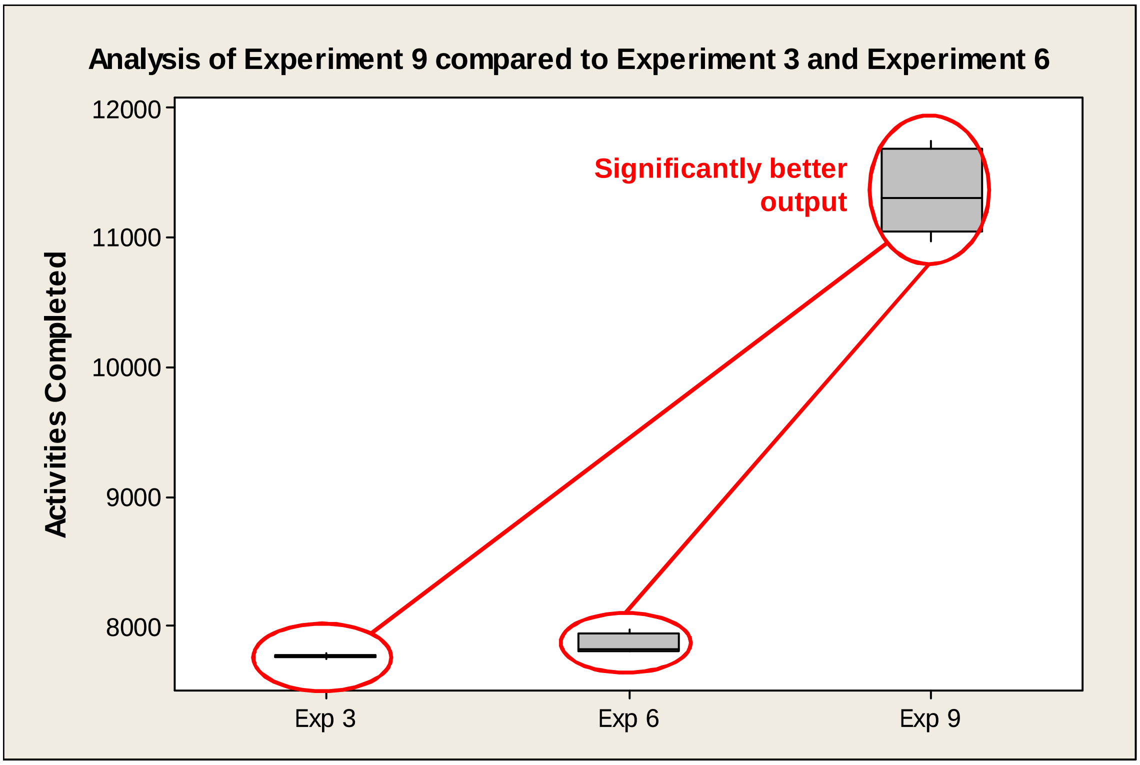

Figure 6.16: Comparison of the output results of Experiment 9 to Experiment 3 and Experiment 6

6.12. Conclusion

It can be concluded that one manager agent interacting in this system does not make a significant difference on the output. When five manager agents are used, the productivity of the system increased significantly especially when the focus of the manager agents' interaction was on the receiving zone operator agents.

When considering the manager agent's effectiveness over the simulation period of 5 days, it can be concluded that they yielded better results when working with new operator agents (days 3 to 5). This is because whilst the manager agents proved to be effective when assisting the deteriorating operator agents to complete their work, they proved to be most beneficial when mentoring new operator agents to learn the job faster.

The system is more sensitive to where interaction occurs than it is to the number of manager agents introduced. This was shown for 10 manager agents working at both the receiving and dispatch zones which yielded an output result only slightly higher than the experiment where 5 manager agents interacted at the receiving zone only,

The second question posed in this work was "Does defined management interventions on a deteriorating / learning system have a positive influence on productivity?" This chapter has established that management intervention can positively affect the productivity of the system being modelled here.

In Chapter 7, the results of each experiment completed in this work will be evaluated using business principles to establish which solution to the deterioration problem is the most effective for the managers of the real system being modelled.

This is an important aspect of the presentation of results because the ultimate aim of this work is to provide a method that can be used by industry to develop solutions to the unique problems that they experience with employee deterioration.

Chapter 7

Conclusion and Discussion

7.1. Introduction

The main purpose of this work was to develop a method that could be used by industry to simulate the effects of operator deterioration on a production system. The previous chapters have explained how the hybrid model was developed, culminating in the management agent interaction model in Chapter 6.

The results of simulation experiments conducted up to this point show that the methodology used in this work can be used by decision makers in industry to understand the impact that human deterioration has on a production system. What is still necessary is to indicate how the results could be analysed financially, because this is how industry analysts decide on a feasible solution based on any of the simulation scenarios.

This chapter uses information from the fourteen simulation experiments conducted in this work to create six scenarios which are analysed using a cost accounting method. The results of this analysis show the cost impact that employee deterioration has on the business process that was modelled, and is used to evaluate the alternate solutions that could be employed to negate the effect of that deterioration.

7.2. Summary of simulation experiments

In Chapters 4, 5, and 6, fourteen experiments were conducted with the various stages of the hybrid simulation model, the main output measure being activities completed. In Chapter 4, the Distribution Centre Discrete Event Simulation (DCDS) model was introduced and an ideal state experiment was conducted where no operator agent deterioration was experienced.

In Chapter 5, the concept of operator agent deterioration and learning was introduced, and the Operator Agent Behaviour (OAB) simulation model was developed using the DCDS model as its foundation. Four experiments were conducted, and each tested the sensitivity of the model's output at different states of operator agent deterioration and learning.

The concept of a manager agent was introduced in Chapter 6, and nine experiments were completed. Each of these investigated the effects of the manager agent(s) interaction with deteriorating or learning operator agents to determine whether an improvement in system output could be realised.

All experiments are summarised in Table 7.1, and a Box plot showing the comparative output measures (activities completed) for each experiment can be seen in Figure 7.1. Up to this point system output has been the main measure of the simulations. Whilst this is not incorrect, within industry a financial comparison / justification is preferred. The next section of this chapter will introduce and use a financial comparison methodology that can be used in conjunction with the hybrid model to assist management teams to decide on the best approach to negating the effect of employee deterioration in their industrial environments.

Experiment	From Chapter:	Details
DCDS	4	All operator agents remain healthy for the entire simulation period
OAB 1	5	Only operator agents in the receiving zone experience deterioration and learning
OAB 2	5	Only operator agents in the dispatch zone experience deterioration and learning
OAB 3	5	All operator agents experience deterioration and learning
OAB 4	5	All operator agents experience deterioration and learning, but for 1 day only (instead of 2 days as with the rest of the experiments
MAI 1	6	All operator agents experience deterioration / learning, but with 25% manager interaction in the receiving zone only
MAI 2	6	All operator agents experience deterioration / learning, but with 50% manager interaction in the receiving zone only
MAI 3	6	All operator agents experience deterioration / learning, but with 75% manager interaction in the receiving zone only
MAI 4	6	All operator agents experience deterioration / learning, but with 25% manager interaction in the dispatch zone only
MAI 5	6	All operator agents experience deterioration / learning, but with 50% manager interaction in the dispatch zone only
MAI 6	6	All operator agents experience deterioration / learning, but with 50% manager interaction in the dispatch zone only
MAI 7	6	All operator agents experience deterioration / learning, but with 5 manager interacting in the receiving zone for 50% of the simulation period
MAI 8	6	All operator agents experience deterioration / learning, but with 5 manager interacting in the dispatch zone for 50% of the simulation period
MAI 9	6	All operator agents experience deterioration / learning, but with 10 manager interacting with all operator agents for 50% of the simulation period

Table 7.1: Summary of all fourteen experiments completed in this research work

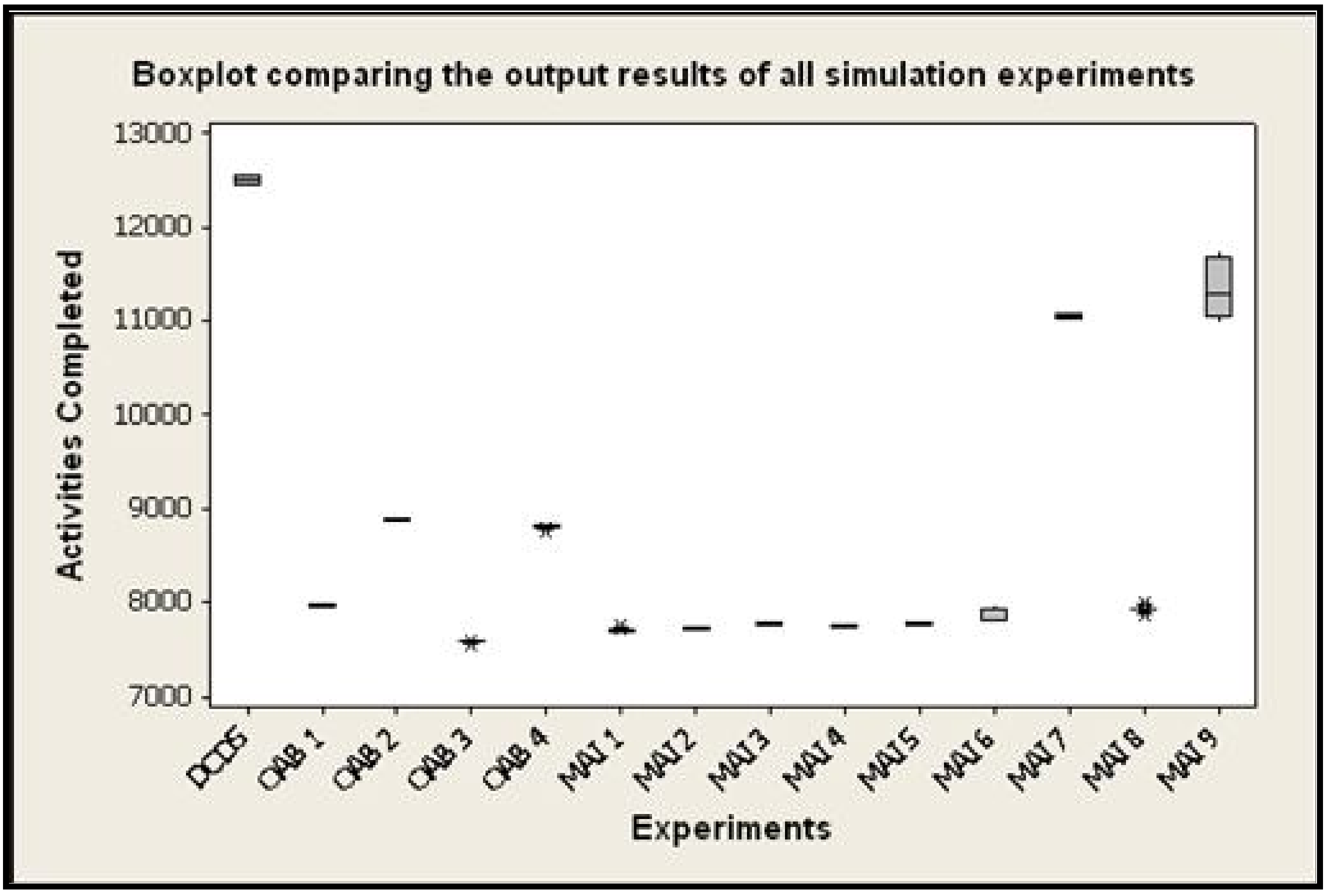

Figure 7.1: Comparison of all experiment output results

7.3. Cost evaluation method

Table 7.2 shows the basic calculation input parameters used to evaluate the scenarios. The costing model used is based on fundamental management accounting principles [34], and is split into three sections:

7.3.1. Input and output parameters (highlighted in yellow), which indicate how many manager agents (A) and operator agents (B) were used in the particular scenario, and what the output was in terms of activities completed (C).

7.3.2. The unit cost parameters which are used to build the financial model are highlighted in orange, and include the daily cost of a healthy / deteriorating operator (D), a new agent (G), and a manager agent (H). Also included is the variable cost component (F) and the amount of income received per unit completed (E).

7.3.3. The section shaded in green takes into account the number of operator agents that experienced deterioration / learning in the scenario (I), as well as the period of deterioration in days (J)

These sections are used in each scenario to calculate the financial benefits and consequences, and the calculation formulas can be seen Table 7.3. This table can be divided into two sections:

7.3.4. The red section contains all the cost and income calculations, of which the total operator agent cost (K) is the most complex. This is because the calculation needs to account for the changing operator agent states over time, which is important because new operator agents incur a lower cost to the business than healthy or deteriorating operator agents.

7.3.5. The profit / loss calculations are found in the blue section where the feasibility of each scenario is calculated for comparison. Most important of these is the annualised profit / loss calculation (P) which shows the effect of the scenario over the 49 weeks of a working year.

Ref	Details	Units
A	No of manager agents	1
B	No of operator agents	20
C	No of units completed in this scenario	12495
D	Unit cost of 1 healthy / deteriorating operator agent per day (ZAR)	275.00
E	Income received per 1 activity of output (ZAR)	18.00
F	Variable cost per unit (excluding labour)	14.75
G	Unit cost of 1 new operator agent per day (ZAR)	175.00
H	Unit cost of 1 manager agent per day (ZAR)	575.00
I	No of deteriorating / new operator agents in this scenario	20.00
J	No of days worked by deteriorating agents before being replaced by new operator agents	-

Table 7.2: An example of the input parameter table to the financial calculations

Ref	Details	Calculation Formula
K	Total operator agent cost (ZAR)	(B*D)-[(I*D*J)+(5-J)*I*G]
L	Total manager agent cost (ZAR)	A*H
M	Total variable cost	C*F
N	Total income received	C*E
O	**Profit / Loss**	**N-(M+L+K)**
P	Annual Profit / Loss (49 weeks)	**O*49**

Table 7.3: Calculation formulas

7.4. Overview of the scenarios

Six scenarios were chosen to show the power of the hybrid model as a decision making tool. Each scenario is based on the fact that it could conceivably occur within a real system. The cost evaluation methodology is applied to each scenario to determine the financial effect so that comparison can be made. A summary of the annualised profit or loss value resulting from each scenario can be seen in Table 7.4.

Scenario's 1 and 2 are the extreme cases of the simulations with the operator agents in Scenario 1 not experiencing deterioration, whilst those in Scenario 2 all experiencing deterioration and learning. Scenario 3 explores what happens when the operator agents experience deterioration for a shorter time before being replaced by new operator agents. Scenarios 4 to 6 analyse different levels of manager agent interaction.

Scenarios	1	2	3	4	5	6
Annual Profit / Loss (49 weeks) (ZAR)	501,453.75	15,925.00	308,528.50	36,627.50	2,952.25	48,497.75

Table 7.4: Comparison of scenario profitability

7.4.1. Scenario 1: No deterioration / learning experienced

This scenario reviews the ideal state, where all operators are healthy operator agents and are able to work at full pace for the entire five day period (2225 minutes) of a simulated production week. The original simulation is found in Chapter 4, and it sets the benchmark for the most activities completed. The high productivity of the system also results in high profits, realising annualised profits of ZAR 500 000. Table 7.5 shows a summary of the input calculations.

Details	Value (ZAR)
Total weekly operator agent cost (ZAR)	27,500.00
Total weekly manager agent cost (ZAR)	2,875.00
Total weekly variable cost	184,301.25
Total weekly income received	224,910.00
Profit / Loss per week	**10,233.75**
Annual Profit / Loss (49 weeks)	**501,453.75**

Table 7.5: Scenario 1 profit/loss summary table

7.4.2. Scenario 2: All operator agents experience deterioration / learning

This situation is the worst case scenario, with all 20 of the operator agents experiencing deterioration and learning as per experiment 3 of Chapter 5. The annualised profit of ZAR 15 000 (refer to Table 7.6) is significantly lower than that in Scenario 1, which shows how badly a business can be affected by employee's who are not able to work at their full capacity.

Details	Value (ZAR)
Total weekly operator agent cost (ZAR)	21,500.00
Total weekly manager agent cost (ZAR)	2,875.00
Total weekly variable cost	112,100.00
Total weekly income received	136,800.00
Profit / Loss per week	**325.00**
Annual Profit / Loss (49 weeks)	**15,925.00**

Table 7.6: Scenario 2 profit/loss summary table

7.4.3. Scenario 3: All operator agents experience deterioration for a shorter period

The financial advantages of being able to detect a downward trend in an operator's productivity, and then replacing him / her with a new and inexperienced replacement can been seen in this scenario. In this simulation, all operator agents experience deterioration, but for one day only.

On the morning of the second day they are all replaced with new operator agents who then go through an ordinary learning cycle (as per the other simulation scenarios). Table 7.7 shows that this scenario realised an annualised profit in excess of ZAR 300 000, which is significantly better than the other scenarios where some operator deterioration is experienced (refer to Figure 7.2).

Details	**Value (ZAR)**
Total weekly operator agent cost (ZAR)	19,500.00
Total weekly manager agent cost (ZAR)	2,875.00
Total weekly variable cost	130,124.50
Total weekly income received	158,796.00
Profit / Loss per week	**6,296.50**
Annual Profit / Loss (49 weeks)	**308,528.50**

Table 7.7: Scenario 3 profit/loss summary table

7.4.4. Scenario 4: All operator agents experience deterioration with one manager agent interacting for 50% of the day in the receiving zone only

In Chapter 5, it was revealed that the system output is more sensitive to deterioration experienced in the receiving area than anywhere else. In this scenario, a manager agent spends half of its day interacting with the operator agents in this area. The presence of the manager agent means that the system incurs an extra cost of ZAR 575 per day, which must be accounted for in the calculation.

When scenario 4 is compared with scenario 2 in Figure 7.2, a marginal increase in profitability can be noticed. This is due to the increased revenue received from the slightly higher throughput due to the manager agent's influence on the operator agents.

However, the annualised profit of ZAR 36 627(refer to Table 7.8) is too small to make this option financially viable in the long term.

Details	Value (ZAR)
Total weekly operator agent cost (ZAR)	21,500.00
Total weekly manager agent cost (ZAR)	2,875.00
Total weekly variable cost	114,017.50
Total weekly income received	139,140.00
Profit / Loss per week	**747.50**
Annual Profit / Loss (49 weeks)	**36,627.50**

Table 7.8: Scenario 4 profit/loss summary table

7.4.5. Scenario 5: All operator agents experience deterioration with five manager agents interacting for 50% of the day in the receiving zone only

In Chapter 6, various levels of manager agent interaction were simulated, and it was concluded that the more manager agents added to the simulation the higher the output. This may be acceptable from a pure throughput perspective, but when analysed financially, the scenario is not as attractive. This is due to the extra costs incurred to have the operator agents in the system.

As seen in Figure 7.2 this is the least profitable scenario, yielding just under ZAR 3 000 per annum (refer to Table 7.9). Part of the reason is the low productivity of the manager agents, because they only interact with the operator agents for half the of their time in the system, yielding similar results to having only 2.5 manager agents interacting with the system on a full time basis.

Details	Value (ZAR)
Total weekly operator agent cost (ZAR)	21,500.00
Total weekly manager agent cost (ZAR)	14,375.00
Total weekly variable cost	163,090.75
Total weekly income received	199,026.00
Profit / Loss per week	**60.25**
Annual Profit / Loss (49 weeks)	**2,952.25**

Table 7.9: Scenario 5 profit/loss summary table

7.4.6. Scenario 6: All operator agents experience deterioration with five manager agents interacting with all operator agents for 100% of the day

The final experiment in Chapter 6 involves five manager agents interacting with deteriorating / learning operator agents for the entire simulation period. The results show that a significant increase in throughput could be realised from having the extra manager agents in the simulation space (refer to Figure 7.1). However, the feasibility of this option must be questioned because the extra cost of the manager agents takes a heavy toll on the profitability of the scenario (refer to Figure 7.2). An annualised profit of just below ZAR 50 000 is realised (refer to Table 7.10).

Many businesses in South Africa are faced with a similar reality. In order to stay competitive they are forced to add extra resources into their production systems to sustain output levels in order to satisfy customer demand. The extra costs erode hard won profits, which can cause investors to look for alternate countries with lower labour costs or a healthier workforce.

Details	**Value (ZAR)**
Total weekly operator agent cost (ZAR)	21,500.00
Total weekly manager agent cost (ZAR)	14,375.00
Total weekly variable cost	167,309.25
Total weekly income received	204,174.00
Profit / Loss per week	**989.75**
Annual Profit / Loss (49 weeks)	**48,497.75**

Table 7.10: Scenario 6 profit/loss summary table

7.5. Conclusion of scenario results

Figure 7.2 summarises the profit made by each scenario, and the most desirable scenario, from both an output and profitability point of view is Scenario 1 where all operator agents are healthy for the entire simulation period. The least desirable is Scenario 2 where all the operator agents experience deterioration and are replace by new operator agents who have to learn how to do the job from scratch.

In an environment where deterioration is inevitable, it would be in the best interests of the management team of this system to remove the deteriorating entities as quickly as possible, even if they were replaced with new operator agents. The current South African practice of labour brokering would allow employees to be removed and replaced at no cost. This is confirmed by Scenario 3, which is the second most profitable scenario conducted.

The scenarios in which manager agents are involved in interacting with the operator agents proved to be marginally profitable, even though the output was higher than the worst case scenario. This shows that whilst introducing more management entities to a system is good for throughput, it could be to the detriment of profitability.

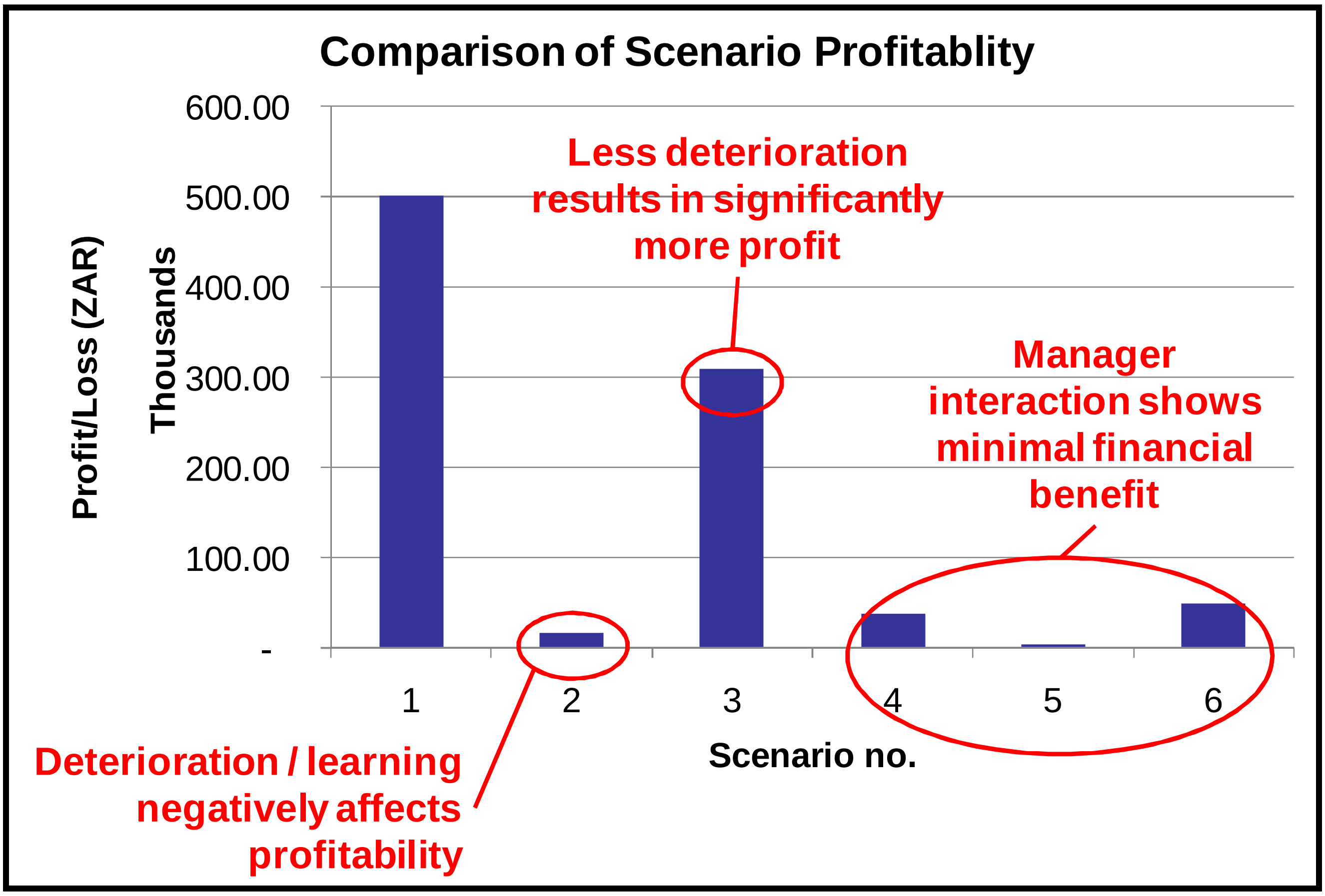

Figure 7.2: Comparison of scenario profitability

7.6. Discussion

Simulation is a decision making tool for managers who are under pressure to make decisions in environments that contain multiple variables. The typical South African production manager is expected to deliver an optimum output from his/her production system despite the influence of AIDS, Influenza, Tuberculosis, or other conditions that can affect the productivity of a typical employee.

It is important that these managers have at their disposal, a range of simple yet effective tools that they can use to assist in making the correct decisions to ensure that targets are met. The aim of this work was to create a conceptual model that is accessible to managers of flow production facilities to assist them to understand the effects of employee deterioration in their production environments through the development of logical models similar to the one used in this work.

Part of this work also endeavoured to understand the effect that management intervention might have on the productivity of a facility through direct interaction with the operators working on the production floor. This is important because South African managers in particular tend to believe that it is effort rather than organisational structures that are recognised as effective, and thus tend to spend more time physically assisting operators instead of searching for ways to optimise processes.

The results of the model experiments in Chapter 6 show that the hybrid model can be used to assist managers to predict the effects that deteriorating operators will have on the output of a production system. The experiments indicate that deterioration can have a significant effect on the productivity of a system, especially when experienced at the beginning of a flow process where throughput can be stifled.

The model also shows the value of identifying the symptoms of deterioration early enough to be able to remove the operators before too much damage is done to the output of the production system. This is evident in Chapter 5 where deteriorating operators were removed from the production system after only a day of deterioration, leading to a significant increase in output over the week.

Instead of simply accepting lower throughput, it may be advantageous for a production system to maintain a team of semi-skilled replacement workers who would be ready to quickly replace operators who display early signs of deterioration.

This is important when considered in a South African context. The country's isolation from the rest of the world requires a significant amount of time be set aside for transporting products to their destinations around the world. Shipping products to East Asia, Europe, or the Americas usually requires between 6 and 8 weeks. If South African is to be a global supplier of product, it needs to manufacture within tight lead times in order to meet international delivery dates that are promised to clients.

The hybrid model provides managers with a tool that can be adapted to understand the early signs of deterioration in their production systems, and thus allow them to take preventative action. This is especially relevant to flow production systems where bottleneck operations dictate the throughput of the system. Goldratt [18] introduces the concept that a unit of throughput lost to a bottleneck operation is a unit lost to the production system's throughput.

In Chapter 5, it was obvious that the receiving zone was the bottleneck operation in the system because deterioration experienced here had a more significant effect on throughput than when experienced in the dispatch zone.

For the production system used here, the model also indicates the importance of having a manager who does not get involved in the physical nature of the job, but instead oversees and organises the work. This was shown in Chapter 6 where one manager agent interacting on the system for 75% of the production day was unable to realise a significant increase in output. This indicates that no matter how willing or enthusiastic one manager may be his or her physical efforts to increase throughput in a deteriorating system are futile.

However, when five manager agents were introduced to the receiving zone, a significant difference in output was observed. It would not be economical for an organisation to begin employing more managers to compensate for deteriorating employees, but this further illustrates the point that a pool of extra operators would be an advantage to a production system.

Further to this point is the importance of good operator training. When new operator agents were introduced to the operation without any training (refer to Chapter 5), they took 2.5 days to become fully productive on the job. However, when they were given more management attention (refer to Chapter 6) they become qualified at a much faster rate and display better productivity attributes sooner. This improved the output of the production system, which is a favourable outcome.

The hybrid model can thus also be used to predict the effect of qualified and unqualified operators on system productivity when they are introduced at different stages of the model run. The results of these experiments could be used to justify larger training budgets for organisations that experience high employee deterioration so that they are always able to provide experienced substitute operators when the need requires. The simulation tool points to a possible solution where ongoing recruitment and training of operators is employed to replace deteriorating operators as soon as their productivity decreases.

It is interesting that the simulations indicate improved throughput when manager intervention is applied but the cost analyses of this chapter show that this does not necessarily improve profitability. Thus, the use of simulation together with a cost analysis is shown to be a possible tool for assessing functionality of management scenarios before implementation which could fail and result in costly mistakes.

The concepts introduced in this work can be further expanded to encompass other organisations that might not necessarily be flow production systems (administrative departments for example), as well as profit driven industries. For example, South Africa's Public Health system has been set optimistic service delivery targets which require all resources within the department to operate at optimum. The impact of poor productivity within this department could be important factor for the general health of the entire nation. The hybrid model could be adapted for research on these types of topics.

The mechanics of the hybrid model could be further improved by developing a better model interface module. The model as discussed in this work requires four different software programs, namely: Simul8, MS Excel, Crystal Ball, and Minitab. A customised software program could be written to substitute the interface and calculations that were done using these packages. For example a Visual Basic program could be written to make the interface and programming of the model easier to do, allowing less skilled model users to develop customised simulation models for their specific systems.

Simulation research to-date has shown that modelling human behaviour is limited [25]. Modelling the full complexity of human beings becomes a philosophical pursuit, rather than a quantitative one. The continuous evolution of agent based simulation techniques and software is allowing for exciting innovation and sophistication in simulation of human interaction in a number of different environments. The work done in this thesis is a start in a possible application of simulation to assist in the management of worker deterioration problems.

Chapter 8

References

1. Remondino M., 2004, Analysis of Agent Based Paradigms for Complex Social Systems Simulation, *Doctoral Thesis, Department of Computer Science, University of Torino, Italy*

2. Remondino M., 2003, Agent Based Process Simulation for Management and Economics, *Economic Complexity Online Journal*, get the address here

3. Kelton W.D., Sadowski R.P., and Sadowski D.A., 1998, *Simulation with Arena*, McGraw-Hill Inc, Boston, MA,

4. Remondino M., 2003, Minority Rule Applied to Multiple Agents Linked into a Social Network with Communication and Memory, *No journal reference as yet,* pp

5. Cioppa T., and Lucas T., 2004, Military Applications of Agent Based Simulations, *Proceedings of the 2004 Winter Simulation Conference,* pp 171-180.

6. Sanchez S., and Lucas T., 2002, Exploring the World of Agent Based Simulations: Simple Models, Complex Analyses, *Proceedings of the 2002 Winter Simulation Conference,* pp 116-126.

7. Rocha L.M., 2003, From artificial life to semiotic agent models: review and research directions, http://informatics.indiana.edu/rocha/sim/review.html

8. Carson II J.S., 2004, Introduction to modelling and simulation, *Proceedings of the 2004 Winter Simulation Conference,* pp 9-16

9. Forrester J.W., 2002, System Dynamics and K12 Teachers, *Roadmaps – A guide to learning Systems Dynamics*, Sloan School of Management, Massachusetts Institute of Technology, http://sysdyn.mit.edu/road-maps/home.html

10. Mayo D.D., Dalton W.J., and Callaghan M.J., 2003, Steering strategic decisions at London Underground: evaluating management options with System Dynamics, *Proceedings of the 2003 Winter Simulation Conference,* pp 1578-1584.

11. Rabelo L., Helal M., and Lartpattarapong C., 2004, Analysis of supply chains using System Dynamics, neural nets, and eigenvalues, *Proceedings of the 2004 Winter Simulation Conference,* pp 1136-1144.

12. Banks J., Carson J., Nelson B., and Nicol D., 2000, *Discrete Event Systems Simulation – 3rd edition*, Prentice Hall, Upper Saddle River, NJ

13. Schmidt B., 2002, What are agents, and what are they used for, *Agent based simulation 2 Workshop,* Lehrstuhl fur Operations Research and Systemthoerie, Passau University, Germany.

14. Rocha L.M., 1999, Complex systems modelling: using metaphors from nature in simulation and scientific models, *BITS: Computer and Communication News – November 1999,* Computing, Information, and Communications Division, Los Alamos National Laboratory, http://informatics.indiana.edu/rocha/sim/review.html

15. Remondino M., 2005, Different Paradigms of Software Agents Applied to Modeling and Simulation of Complex Social Systems, *Proceedings of CogSCI 2005*.

16. Franklin S., and Graesser A., 1997, Is it an Agent, or just a Program, *Proceedings of the ATAL Workshop*, pp 193-206.

17. Arsham H., 2007, Systems Simulation: The shortest route to application, University of Baltimore Website Article, http://ubmail.ubalt.edu/~harsham/simulation/sim.htm

18. Goldratt E., 1984, *The Goal*, Penguin, New York, NY

19. Hao Q., and Shen W., 2007, Implementing a hybrid simulation mode for a Kanban-based material handling system, *Robotics and Computer-integrated Manufacturing 24 (2008)*, pp 635-646

20. Xiang W., and Lee H.P., 2007, Ant colony intelligence in multi-agent dynamic manufacturing scheduling, *Engineering Applications of Artificial Intelligence 21 (2008)*, pp 73-85

21. Zhang D.Z., Anosike A.I., Lim K.M., Akanle O.M., 2006, An agent-based approach for e-manufacturing and supply chain integration, *Computers and Industrial Engineering 51 (2006)*, pp 343-360

22. Anosike A.I., Zhang D.Z., 2007, An agent-based approach for integrating manufacturing operations, *International Journal of Production Economics (2007)*, pp 1-20

23. Pavon J., Arroyo M., Hassan S., Sansores C., 2007, Agent-based modelling and simulation for the analysis of social patterns, *Pattern Recognition Letters 29 (2008)*, pp 1039-1048

24. Urlings P., Sioutis C., Tweedale J., Ichalkaranje N., Jain L., 2004, A future framework for interfacing BDI agents in a real-time teaming environment, *Journal of Network and Computer Applications 29 (2006)*, pp 105-123

25. Brailsford S., Schmidt B., 2003, Towards incorporating human behaviour in models of health care systems: An approach using discrete event simulation, *European Journal of Operations Research 150 (2003)*, pp 19-31

26. Leombruni R., Richiardi M., 2005, Why are economists sceptical about agent-based simulations, *Physica A 335 (2005)*, pp 103-109

27. Labarthe O., Espinasse B., Ferrarini A., Montreuil B., 2006, Toward a methodological framework for agent-based modelling and simulation of supply chains in a mass customisation context, *Simulation Modelling Practice and Theory 15 (2007)* pp 113-136

28. Simon H.A., 1957, *Models of man. Social and rational*, Fifth printing 1967, Wiley and Sons, New York

29. Simon H.A., 1955, A behavioural model of rational choice, *Quarterly Journal of Economics 69 (1955)*, pp 99 -118

30. Baesler F.F., Sepulveda J.A., 2004, The use of simulation and design of experiments for productivity improvement in the sawmill industry, *Proceedings of the 2004 Winter Simulation Conference*, pp 1218 – 1221

31. Albino V., Carbonara N., Giannoccaro I., 2006, Innovation in industrial districts: An agent-based simulation model, *International Journal of Production Economics (2006)*, pp 30-45

32. Cantamesssa M., 1997, Hierarchical and heterarchical behaviour in agent-based manufacturing systems, *Computers in Industry 33 (1997)*, pp 305-316

33. Fioretti G., Lomi A., 2008, The garbage can model of organisational choice: An agent-based reconstruction, *Simulation Modelling Practice and Theory 16 (2008)*, pp 192-217

34. Faul MA., Van Vuuren SJ., Du Plessis PC., 1992, *Fundamentals of cost and management accounting*, Butterworths, Durban, South Africa

35. Levene, Howard (1960). "Robust tests for equality of variances". in Ingram Olkin, Harold Hotelling et al. *Contributions to Probability and Statistics: Essays in Honor of Harold Hotelling*. Stanford University Press. pp. 278-292.

36. Kanawaty, George, 1992, *Introduction to Workstudy*, International Labour Organisation, Geneva, Switzerland