DOCTOR OF PHILOSOPHY

An agent approach to improving radio frequency identification enabled Returnable Transport Equipment

Rebecca Aggarwal

2014

Aston University

AN AGENT APPROACH TO IMPROVING RADIO FREQUENCY IDENTIFICATION ENABLED RETURNABLE TRANSPORT EQUIPMENT

VOL. 1

REBECCA AGGARWAL

Submitted for the Doctor of Philosophy Degree

ASTON UNIVERSITY

April 2013

Aston University

An Agent Approach to Improving Radio Frequency Identification Enabled Returnable Transport Equipment

Rebecca Aggarwal

Doctor of Philosophy, 2013

Thesis Summary

Returnable transport equipment (RTE) such as pallets form an integral part of the supply chain and poor management leads to costly losses. Companies often address this matter by outsourcing the management of RTE to logistics service providers (LSPs). LSPs are faced with the task to provide logistical expertise to reduce RTE related waste, whilst differentiating their own services to remain competitive. In the current challenging economic climate, the role of the LSP to deliver innovative ways to achieve competitive advantage has never been so important. It is reported that radio frequency identification (RFID) application to RTE enables LSPs such as DHL to gain competitive advantage and offer clients improvements such as loss reduction, process efficiency improvement and effective security. However, the increased visibility and functionality of RFID enabled RTE requires further investigation in regards to decision-making.

The distributed nature of the RTE network favours a decentralised decision-making format. Agents are an effective way to represent objects from the bottom-up, capturing the behaviour and enabling localised decision-making. Therefore, an agent based system is proposed to represent the RTE network and utilise the visibility and data gathered from RFID tags. Two types of agents are developed in order to represent the trucks and RTE, which have bespoke rules and algorithms in order to facilitate negotiations. The aim is to create schedules, which integrate RTE pick-ups as the trucks go back to the depot. The findings assert that:

- agent based modelling provides an autonomous tool, which is effective in modelling RFID enabled RTE in a decentralised format utilising the real-time data facility.
- the RFID enabled RTE model developed enables autonomous agent interaction, which leads to a feasible schedule integrating both forward and reverse flows for each RTE batch.
- the RTE agent scheduling algorithm developed promotes the utilisation of RTE by including an automatic return flow for each batch of RTE, whilst considering the fleet costs and utilisation rates.
- the research conducted contributes an agent based platform, which LSPs can use in order to assess the most appropriate strategies to implement for RTE network improvement for each of their clients.

Keywords: Multi-Agent Systems, Competitive Advantage, Logistics Service Providers, Advanced Auto-Identification Technology, Reverse Flow Scheduling

Dedication

To my dad, Laj Aggarwal, who instilled that the price of success is hard work, dedication to the job at hand, and the determination that whether we win or lose, we have applied the best of ourselves to the task at hand.

To my mom, Meena Aggarwal, who has sacrificed so much and taught me that true success lies in people who continue to preserve even after falling and getting hurt, whilst remaining dedicated to the truth.

To my brother, Gagan Aggarwal, who shows me daily that every time someone tells you that you can't do something, all you have to do is to turn around, smile and say 'watch me'…

I dedicate this thesis to my family who have provided me with the opportunity, strength and motivation to work hard and finish my Ph.D.

Thank you very much.

Rebecca

Acknowledgements

Firstly, thank you to the EPSRC and DHL Exel Supply Chain for funding this project. I would like to thank my supervisor Dr. John Elgy for his advice and good humour. I would also like to thank my second supervisor Dr. Jane Andrews for all the invaluable support and encouragement. My acknowledgements would not be complete without thanking Sandra Mosley and Susan Doughty who have been wonderfully supportive during the Ph.D process. Thank you all for the advice, support and for helping me to complete this thesis.

Thank you to Jos Daalhuizen and Karen Visgarda from DHL Exel Supply Chain for providing me with insight into the business, access to meetings and site visits. Thanks also to Dr. Michael Winsper for providing support with the Java programming skills. Prior to personnel changes, Dr. Ming K. Lim is acknowledged as an initial supervisor.

The biggest thanks of all go to my colleagues and friends who have been on the rollercoaster ride with me over the last few years. Thanks to Karin Qureshi, Ashwanta Rajamani, Lucy Killoch, Charanjit Kaur, Magesh Nagarajan, Samia Yasmin, Sarah Peacock, Zsuzsi Mayer, Nasreen Yasin, Johanna Virkki, Alina Patelli, Jennifer Allen, Constant Iannacci, James Stone, Rachel Darby, Nuch Phusunti, Emmanuel Farrugia, Patricia Hernandez, Alice Mitchell, Javad Khazaee, Samuel Fosso Wamba, Evelyn Chalencon, Yassine Falaqui, Charlotte Sketchley-King, Karen Kaur, Megha Prasad, Mishbah Ali, Chuhui Li, Helen Higson, Margaret Banks, Louise Knight, Alison Hodge, Grace Mills, Diar Nasiev, Werachai Lamo, Rajinder Kaur Dhillon, Ravinder Judge, Aniko Kalas, Sham and my flatmates - Kyle, Gabby, Colin, Megan, Blessed and Arune. Thanks for the all the kind words and laughs… all you guys kept me sane!

Contents

List of Figures

List of Tables

List of Acronyms

LSPs	Logistics Service Providers
RTE	Returnable Transport Equipment
RFID	Radio Frequency Identification
MAS	Multi-Agent Systems
ABM	Agent Based Modelling
RFIDERTEAM	Radio Frequency Identification Enhanced Returnable Transport Equipment Agent Model
RTEASA	Returnable Transport Equipment Agent Scheduling Algorithm
UML	Unified Modelling Language
auto_ID	Automatic Identification
RL	Reverse Logistics
ROI	Return on Investment
DDM	Decentralised Decision-Making
DAI	Distributed Artificial Intelligence
GPS	Global Positioning System
AI	Artificial Intelligence
GSM	Global System for Mobile Communication
ERP	Enterprise Resource Planning
APS	Advanced Planning System
DF	Distribution Facilities
cbm	Cubic meters

Chapter 1

Introduction

Returnable transport equipment (RTE), such as pallets, crates, trolleys, trays and totes form an essential part of the supply chain. RTE usage continues to rise, particularly as a consequence of the recent economic and environmental pressures as a more sustainable system to transport goods is pursued in comparison to the traditional one-way packaging system (Kroon and Vrijens, 1995, Ilic et al., 2009, LogicaCMG, 2004). The basic function is to assist with the movement of client's goods to the right place, at the right time and in the right condition. The RTE forms a closed loop system increasing the number of times each piece of equipment is handled in comparison to the goods being transported, increasing operating costs if efficient management techniques are not deployed. The number of times a piece of RTE is used during its lifecycle is important and significant to the purpose of its function as a piece of returnable equipment (Kroon and Vrijens, 1995).

1.1 RTE Management

RTE is an important and essential component of the supply chain, as it moves in a continuous manner across industries and all over the globe. However, the resources of the supply chain are assembled to focus solely on moving the products being carried by the RTE, and therefore the equipment is not individually managed and often forgotten. It is becoming increasingly difficult to ignore the poor management of RTE as they are expensive assets and any losses are costly, as well as disrupting the network. Therefore, the management of RTE is an important feature in the supply chain; however due to the dynamic nature of the RTE network, it has proven to be a challenge to improve (LogicaCMG, 2004). The management is often outsourced to logistics service providers (LSPs) as their expertise is utilised to find better solutions (Twede and Clarke, 2005, McKerrow, 1996). In recent years, there has been an increasing interest to create sustainable supply chains, one component of which concerns packaging, and the movement away from a one-way packaging system (such as disposable cardboard) to RTE usage.

1.2 Motivation for Research

Today, competition between logistics service providers (LSPs) continues to intensify. The advent of technology and transportation systems has enabled fast rates of globalisation for LSPs, which in turn has led to increased levels of competition for market share across the industry. Therefore, in order for LSPs to increase their competitiveness, and cope with recent global economic pressures, companies must develop logistics related IT capabilities and be proficient in offering services such as RTE management and auto-ID application of technologies such as RFID (Power et al., 2007, Coyle et al., 2003). A RTE network of an LSP becomes more complex as each client and product may require a different treatment, and therefore a range of supply chain solutions need to be available. RFID, which is an advanced auto-ID technology offers information sharing capability, which an LSP exploits by providing appropriate IT solutions and technology integration techniques, creating value added advantage for their clients (Lin, 2009). In addition to this the LSPs simultaneously increase their own competitive advantage by gaining more business through the creation of higher levels of supply chain solutions. In this context the RTE network is where the LSP provides the RTE and manages the network on behalf of the clients such as the practice conducted by the RTE department at DHL Excel Supply Chain, partners of this project. The LSP has an intricate, challenging and dynamic network of RTE where RFID application provides visibility through the creation of a real-time data environment, an area which requires further exploration in regards to achieving competitive advantage.

1.3 Project Aim

The aim of the thesis is to improve the management of RFID enabled RTE. After reviewing and analysing the literature, it is observed that there is a requirement for both lean and agile approaches to be introduced in order to improve RTE management. Currently, the major problems with RTE management include: low rotation rate, long cycle times and losses. These problems combined with poor visibility and management lead to an unresponsive and dormant network. Increasing agility in the RFID enabled RTE network will improve the network's ability to respond to unexpected changes and cope with dynamic elements. The introduction of effective decision-making units via agent-based RTE plays a significant role in increasing RTE network mobility.

1.4 Research Questions

Ilic et al. (2009) has already drawn attention to the lack of the simulation models for RTE management improvement, arguing that further research needs to be conducted in the RTE management problem domain. There has been some research, which has explored the application of RFID to improve RTE management by utilising the feature of object-level visibility (Johansson and Hellström, 2007, Martínez-Sala et al., 2009). However, these studies tend to stress the benefits of RFID application and do not offer a modelling and simulation environment to improve the decision-making process from the perspective of the RTE network. Creating an agile RTE network where each unit is being fully utilised is challenging, but this is further heightened when there are a variety of products, with multiple clients, moving goods through various nations under a range of different supply chain set-ups. Too little attention has been paid to the development of modelling and simulation to aid in multi-layered decision-making in challenging environments such as RTE networks, particularly decision-making which benefits both the LSP and their clients. Therefore, by developing a model which conducts decisions to integrate the reverse flows into the forward scheduling system and allows the RTE to be returned as soon as possible for re-use, increases the RTE rotation rate and improves utilisation. This will produce resilient RTE networks with many benefits for the clients and enable the LSP to maintain or gain new clients, improving competitive advantage. The main questions addressed in this thesis are:

i) How to improve the RFID Enabled RTE network;
ii) How the data gathered in real-time from RFID tags on RTE can be utilised further;
iii) How to model the RTE network and capture its distributed nature;
iv) How to enable decentralised decision-making from the perspective of the RTE;
v) What rules and procedures are required in order to ensure that the RTE is utilised to its maximum;
vi) How to simultaneously improve the client's supply chain, increasing competitiveness of the LSP through RFID enabled RTE.

1.5 Methodology

There are many layers to the research questions. In order to address them there is a requirement for a paradigm shift from conventional modelling techniques currently used, to the implementation of distributed decision support systems. A decentralised approach to RTE management assists in encouraging agility and fostering leaner networks, both of which are characteristics lacking from current RTE practices. To represent the RTE and understand the behaviour, the network needs to be modelled in a modular manner with a micro-level approach. This research investigates the application of multi-agent systems (MAS) to facilitate distributed decision-making in reference to the RFID enabled RTE network management problem domain. The field of MAS is an exciting and rapidly expanding area of research development as the characteristics of agents offers a more realistic representation of physical entities than traditional modelling approaches, such as discrete event simulation and systems dynamics (Borshchev and Filippov, 2004). Interest in the application of MAS to supply chains is increasing considerably, and the decentralised nature of logistics networks presents an environment ripe for distributed artificial intelligence application (Swaminathan et al., 1998, Fox et al., 2000). This research fills a gap in the knowledge by applying MAS to RFID enabled RTE networks, utilising the visibility gathered from RFID tags.

1.6 Radio-Frequency Enhanced RTE Agent Model (RFIDERTEAM) Objectives

The RFID enhanced RTE agent model (RFIDERTEAM) and simulation platform is developed with RTE Batch agents and Truck agents, offering an environment to improve the RFID enabled RTE network with an autonomous and distributed approach. Each agent is developed with a set of rules, which will benefit the object that is represented. For example, the RTE endeavours to return back as soon as possible, whilst the truck aims to create profitable routes. The model initiates interactions between the agents through negotiation where the model aims to develop an autonomous approach, which achieves RTE based competitive advantage on a variety of levels as illustrated in Figure 1.1.

Cost advantage via the RFIDERTEAM for RTE network improvement is achieved by two streams: removal of RTE waste and removal of fleet waste, with the aim of creating a leaner network. The first stream, removal of RTE waste, requires the implementation of four methods: reduction in the quantity of days a RTE is empty, a reduction in the time RTE is idle, decrease of RTE cycle time and an increase of the RTE rotation rate. Although the focus is on improving the RTE network through the achievement of the objectives set in the first stream, the fleet has a significant impact on RTE network productivity. The second stream, removal of fleet waste, has three components for focus: reduction of empty miles, increase in number and quantity of loadings in the reverse flow empty RTE pick-ups and an evaluation of the RTE batch combination selected in regards to the truck constraints (such as weight, volume and time). The aim is to reduce the amount of time the truck is idle and utilise the driving hour to pick-up as many loads as possible in the reverse flow as it goes back to the distribution centre.

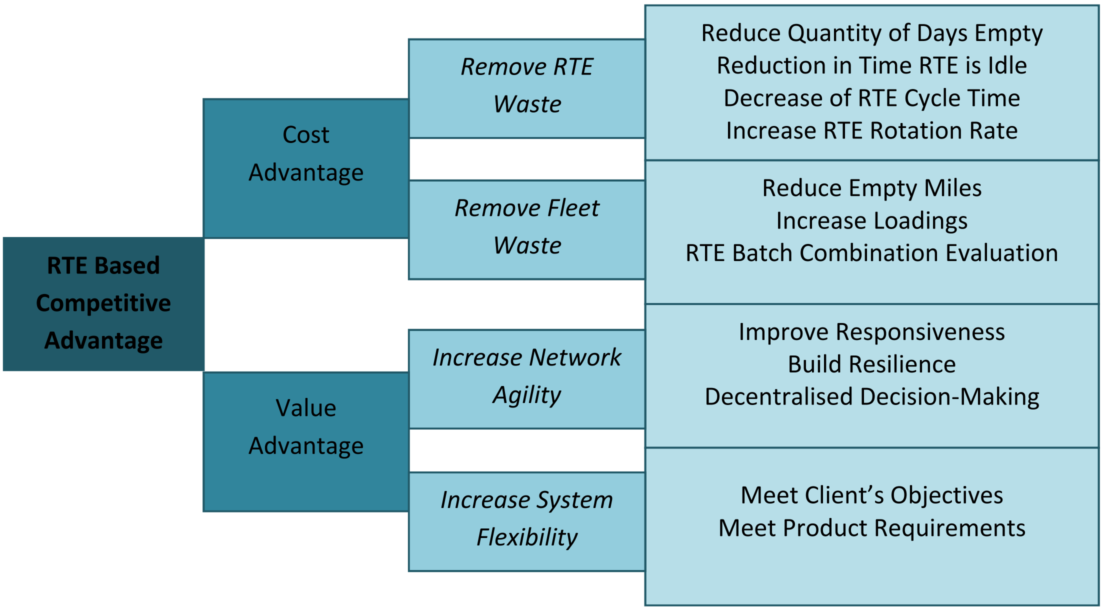


Figure 1.1 Illustration of RTE Network Competitive Advantage Objectives

The RFIDERTEAM implements agility in the RTE network by improving responsiveness, creating resilience and utilising the functionality of decentralised decision-making when formulating schedules. Firstly, the RTE network needs to be able to respond to events, particularly unexpected events in an efficient manner. Secondly, the model aims to foster resilience in the RTE by removing vulnerabilities, such as losses and under-utilisation. Finally, the decision-making is localised so that the agent representing the object makes efficient and effective decisions in a reasonable time-frame without having to include the whole network. An agile system responds to changes; however a network also requires flexibility which refers to how the network responds. A flexible RTE network needs to be able to reconfigure efficiently by returning the RTE as soon as possible back to the point of use whilst adhering to the product's requirement and client's objectives. At this point the decision-making goes beyond a RTE level to include the products being moved and the client's requirements, enhancing LSPs strategic advantage. By utilising the resources of the network, such as the fleet and the information systems, flexibility is built into the system. In the RFIDERTEAM, the flexibility achievement is complex as it is multi-layered, with a variety of products, RTE and numerous clients. A client of a LSP may require a reduction in their inventory whereas another client may prefer a reduction in their product's lead-time. Each client may be utilising the same resources of the LSP but will require different services, therefore LSPs need to offer service differentiation. Therefore, the model offers flexibility by representing different types of supply chains and the ability to meet the product requirements and client's objectives.

The RFIDERTEAM implements a decentralised decision-making system for RFID enabled RTE increasing the quantity of decisions and encouraging a high number of correct decisions by using the information-centric environment created by the real-time data visibility from the tagged equipment. The research presented in this thesis encourages the RTE to move from a waste product to an information porter once tagged and then a decision-making unit when represented by an agent, creating a network improvement tool.

1.7 Model Measures and Variables

The thesis will:

- Investigate the feasibility of the application of multi-agent systems (MAS) to model the RTE network utilising RFID based data;
- Develop a methodology to enable the agents to integrate forward and reverse flows of RTE simultaneously, generating a feasible schedule;
- Develop an algorithm, which can be used to promote RTE utilisation;

- Construct a platform for LSPs to use to assess various strategies for their client base and utilise the increased visibility, which RFID enabled RTE provides in comparison to non-RFID enabled network, enhancing their competitiveness.

The measures and variables used during the simulations to analyse the results and are listed and defined in Table 1.1 and Table 1.2, respectively.

Measures
Late Batch Costs
Amount of days batches are late, which are multiplied by a cost per day of not using it.
Truck Operating Costs
The amount of hours a truck is in use, which is multiplied by a cost of operating the truck per hour.
Total Cost
The total cost is a summation of the late batch cost and the truck operating cost.
Volume Utilisation of a Truck
The volume utilisation rate represents the loadings of a truck for a route, each day.
Weight Utilisation of a Truck
The weight utilisation rate of the truck refers to the average weight of the loads in the truck through a route, each day.
Time Utilisation of a Truck
The time utilisation rate refers to the percentage of time the truck spends carrying loads during each route.
Number of Auctions
The number of auctions conducted between the RTE Batch agent and Truck agent in order to schedule the RTE.

Table 1.1 Measures Used During Simulations

Variables
Batch Processing Days
The amount of days a RTE is being processed and awaiting pick-up, which also determines the return delivery date.
Number of Trucks
The number of trucks running through the network is modifiable.
Size of Network (Number of Facilities)
The network is built of distribution facilities, which the trucks leave and return to each day and warehouses and client's facilities.
Network Complexity Factor
The network complexity can be modified to include more or less roads between facilities.
Number of RTE in Each Batch
The amount of RTE in each batch can be modified and set between a lower and upper limit.
Dimensions of Each RTE Batch
The dimensions of each RTE Batch can be modified, which refers to the weight and volume.
Quantity of RTE Batches
The quantity of RTE batches which are placed in the system at the start of the simulation and the amount of RTE batches introduced into the system, each day.

Table 1.2 Variables Used During Simulations

Various scenarios, which are listed in Table 1.3 are simulated and compared to a control network.

The results are presented in chapter 6.

Scenarios Tested
Comparison to RTE which return slowly
Comparison to a system with RTE losses
The batch processing day versus truck operating costs
Impact of increasing the network complexity
Impact of using the model on a larger network
The effect of implementing smaller RTE batch sizes
Implementing a bigger range of RTE batches
Using the model to represent a bigger and more complex network with a larger quantity and range of batches

Table 1.3 List of the Scenarios Tested Using the RFIDERTEAM

1.8 **Contribution**

The thesis provides a comprehensive analysis of the RTE problem, a competitive advantage model for LSPs, an investigation into the application of RFID and the development of a bespoke MAS. All areas have been published across 11 publications, which are summarised in Table 1.4. Each publication presents a contribution to the field.

The RFIDERTEAM is a bespoke piece of software developed specifically for the RTE problem domain, and therefore offers a valuable platform with specific algorithms and tailored methods. These have all been implemented in Java code, where the modular nature of the system leads to high levels of flexibility as problems develop and clients change.

The scenarios developed for testing, demonstrate that the bespoke algorithms which have been implemented, addresses the problem of RTE under-utilisation by ensuring that the scheduling of the reverse flows is prioritised alongside the forward delivery scheduling. The real-time data provided by the RFID tags enables each agent to utilise the information-centric environment to conduct effective and efficient decisions.

The model is a contribution not only to the area of RTE management but also delivers a MAS in RFID enabled RTE management. One of the key requirements for further development in the emerging field of MAS is the development of applications. The research presented in this thesis addresses this by building a foundation for MAS development in RTE management.

Title	Authors	Year	Published	Description	Appendix
Enhancing competitive advantage with radio-frequency identification (RFID) enabled returnable transport equipment (RTE)	Aggarwal, R. and Lim, M.K.	2013	International Journal of Information Technology and Management, Vol. 12, Nos. 1/2, pp.3–26.	This paper proposes the implementation of RFID enabled RTE in combination with the consolidation of network assets and cross-docking. A four-level cyclic framework, which aids LSPs to achieve competitive advantage has been developed, focussing on asset reduction and utilisation, RTE cycle time reduction and the introduction of real-time data in the LSP network.	A-1
Aligning physical and virtual logistical spheres with radio-frequency identification and agent-based modelling	Aggarwal, R. and Lim, M.K.	2013	International Journal of Agile Systems and Management, Vol. 6, No. 1, pp.66–82.	This paper explores the use of RFID and ABM to improve RTE practices by increasing network agility in order to invigorate customisation of logistical services. Agility of services is amplified through the availability of real-time data and adept decision-making tools. Agent-based modelling (ABM) provides a platform to represent physical entities creating a virtual sphere. RFID technology offers full network visibility closing the gap of connectivity between the virtual sphere and physical objects such as pallets, goods and trucks.	A-2
Advances in decision support systems on the endeavour of supply chain optimality	Aggarwal, R. and Lim, M. K.	2012	International Conference on Industrial Logistics (ICIL), Zagreb, Croatia, 14th- 16th June, pp. 472-477.	This paper investigates reasons for the shift to decentralised decision-making and the impact on supply chains. Industry practitioners are seeking to create optimal logistics networks through more efficient decision-making, leading to a shift of power from a centralised position to a more decentralised approach. This has led to researchers, exploring with vigour, the application of ABM in supply chains and more recently, its impact on decision-making.	A-3
Improving returnable transport equipment management with an autonomous approach	Aggarwal, R. and Lim, M. K.	2012	3rd International Conference on Dynamics in Logistics (LDIC), Bremen, Germany, 27 February – 2 March.	This paper, investigates the nature of RTE network management building on current research and practices, filling a gap in the literature through the investigation of a product-centric approach, where	A-4

				the paradigms of "intelligent products" and "autonomous objects" are explored. Advanced research development in the RTE paradigm is proposed in order to align academic research with problematic areas in industry. This is pivotal on the endeavour to developing agile support systems, fully utilising an information-centric environment and encouraging RTE to be viewed as critical network optimising tool rather than a costly waste.	
A decision support system for waste collection management and its potential improvement with radio-frequency identification technology (RFID)	Nielsen, I., Lim, M., Nielsen, P. and Aggarwal, R.	2012	International Journal of Environmental Technology and Management, 2012 Vol.15, No.3/4/5/6, pp.305–324.	This paper presents a decision support system (DSS) framework, based on Constraint Logic Programming, for waste collection management and discusses the potential employment of RFID to improve several critical procedures involved in managing waste collection.	A-5
Increasing competitiveness of third party logistics with RFID	Aggarwal, R. and Lim, M.	2011	IEEE International Conference on RFID-Technologies and Applications (RFID-TA), Sitges, Spain, 15-16 September, pp. 437-444.	This paper highlights that RFID enabled RTE assists LSP in competitive advantage gain through cost reduction and offering innovative logistical services, which are tailored to each client via the introduction of real-time data updates.	A-6
A three-level automation approach for radio-frequency identification enabled returnable transport equipment to enhance supply chain competitiveness	Aggarwal, R., Lim, M. and Tan, K.	2011	21st International Conference on Production Research, Stuttgart, Germany, 31 July – 4 August.	This paper aims to address the gap in the literature by developing a three-level framework automation application approach with the assistance of RFID technology and RTE. The research pinpoints that RFID enabled RTE offers a viable tool to assist supply chain automation in the dynamic and process driven environment of distribution, increasing flexibility, whilst reducing inefficiencies.	A-7
Utilisation of RFID to increase efficiency in the returnable equipment	Aggarwal, R. and Lim, M. K.	2010	12th International Conference on the Modern Information Technology in	This paper explores the application RFID to RTE in the supply chain paying particular attention to the current structures of RTE networks as formulated by	A-8

supply chain			the Innovation Processes of the Industrial Enterprises, Aalborg University, Denmark, 30-31 August, pp. 284-292.	RTE providers. The problems related to RTE usage are described and the effect to the network analysed. RFID is investigated as a tool to assist with the movement of the RTE both from the client's and RTE provider's perspective.	
Impact of RFID in reverse logistics	Aggarwal, R. and Lim, M. K.	2009	11th International Conference on The Modern Information Technology in the Innovation Processes of the Industrial Enterprise, Bergamo, Italy, 15-16 Oct, pp. 185 – 192.	This paper aims to discuss the recent literature on RFID technology and reverse logistics (RL). Particular attention is applied to the bullwhip effect and its increase as RL activities are integrated into the supply chain. RFID is investigated as a tool to assist with integrating reverse and forward logistics into a seamless supply chain, and reducing the bullwhip effect.	A-9
RFID Innovation: An Exploration of the Impact of RFID Technology in Outbound Logistics	Aggarwal, R. and Lim, M. K.	Under Review	International Journal for RF Technologies: Research and Applications	This paper evaluates the importance of outbound logistics and explores the role of RFID. The definition, classification and framework developed will assist academics in the areas of research requiring further exploration and help industry practitioners to identify the benefits of RFID and how best to exploit the technology in order to create further value.	A-10
Transforming Logistics Service Provider's Activities Using IT: The Case of RFID Technology	Aggarwal, R. and Fosso Wamba, S.	Under Review	Production Planning & Control: The Management of Operations	RFID is a critical technology for enhancing logistics activities, however the literature tends to focus on application on a firm level or the overall supply chain, failing to consider the perspective of a LSP. The paper proposes a RFID integration tool, encouraging LSPs to consider the multi-dimensional benefits which are attained and the wider network improvements available, the benefits of which can be passed onto clients.	A-11

Table 1.4 Publications Demonstrating Contribution to Knowledge

1.9 Thesis Structure

This thesis has been organised as follows:

Chapter 2 begins by laying out the theoretical dimensions of the research by reviewing the literature. The review spans the areas of RTE management, RFID application, LSPs and competitive advantage. The chapter also describes the involvement of DHL and the basis for the development of the research problem.

Chapter 3 analyses the literature to decipher the most appropriate methodology to address the research questions posed. A multi-agent system (MAS) approach is selected and compared to other methodologies followed by an investigation as to the utilisation of agent based modelling (ABM) in logistics.

Chapter 4 describes the design and characterisation of the RFID Enhanced RTE Agent Model (RFIDERTEAM) developed. The architecture is described through an array of unified modelling language (UML) diagrams.

Chapter 5 describes how the RFIDERTEAM implements cost and value advantage with an analysis of the code and the investigation of agility and flexibility that the model offers.

Chapter 6 presents the results of the simulations, which were conducted with an array of scenarios, each testing a different aspect of the model.

Chapter 7 presents a discussion on the merit of the model by interpreting the findings and comparing the model to studies in the current literature. This is followed by a description of the limitations.

Chapter 8 summarises the findings in order to address the research questions. An evaluation of the impact of the research presented to RTE management improvement is followed by highlighting the significance of RFID and ABM application to the RTE problem domain. An assertion of the main contributions of the work conducted is presented. The implications of the model to the industry is explored, and finally the future scope of work is outlined.

Chapter 9 concludes the thesis with an Appendix of related work including the 11 publications listed in Table 1.4, DHL Exel Supply Chain documents, a selection of class diagrams for the agent model developed (RFIDERTEAM) and the Java code implemented.

Chapter 2

Literature Review

The advent of technology and development of transportation systems has enabled globalisation of supply chains, which in turn has led to increased levels of competition for market share across industries. Particularly with the recent recession, there is increased competitiveness to capture and retain consumers, reduce operating costs and maintain profitability across industries. Porter (1985) introduces the concept of value chain management, where it is advised that a company should become a market leader in logistics in order to be competitive. Christopher (2005) asserts that logistics represents a high proportion of a company's overall costs. Through the efficient management of logistics activities, firms can differentiate their services and strive to become a cost leader, putting companies in the strongest possible position in the market. A company which is unable to achieve competitive advantage as either a cost leader, where a company carries out all the activities in the value chain at the lowest cost in comparison to their competitors, or a service leader, where a company differentiates their products and provides unique services to customers, would be advised to outsource the logistics component of the company's operations in order to gain competitive advantage (Porter, 1985).

Today, competition between logistics service providers (LSPs) continues to intensify. This is partly due to the increasing trend of companies outsourcing (Dapiran et al., 1996, Hertz and Alfredsson, 2003). The LSP market is growing at a significant rate, in the US market alone total revenue expanded from $39.6 billion in 1998 to $127 billion in 2008, an increase of over 220% (Armstrong & Associates, 2010b). As LSPs merge and become larger, they offer competitive global logistics solutions at better rates than firms are able to achieve in-house (Selviaridis and Spring, 2007). Other motives for companies to outsource include the globalisation of markets and continuous emergence of new technologies (Razzaque and Sheng, 1998). In a recent survey, 94% of respondents stressed that it is essential LSPs have IT capability, however only 54% of firms stated that existing LSPs met their expectations (Langley, 2010). Therefore, in order for LSPs to increase their competitiveness, and cope with recent global economic pressures, companies must develop logistics related IT capabilities and be proficient in offering services such as RFID technology application, enhancing the attraction and improving the retention of customers (Power et al., 2007, Coyle et al., 2003).

Radio frequency identification (RFID) is an evolving advanced auto-identification (auto-ID) technology from the same group as barcodes and optical character recognition. A RFID system has four main components to enable full application as illustrated in Figure 2.1. When requested, or if the tag is active, information is transferred from the tag to the reader via radio waves. The data is then transferred to middleware to be processed. Then the processed data is used in the application for which it is required for example, monitoring temperatures of chilled foods or acknowledging the receipt of an item in store. In comparison to the current widespread auto-ID technology the barcode, RFID has numerous advantages. The most significant advantage is that no line of sight is needed when scanning, and tagged items are read in any position and through materials at a distance (Finkenzeller, 2010). Also, each tag possesses a unique serial number allowing each item to have an individual identification. Each tagged item has numerous item related conditions written onto the tag (for example, special customer requirements or shipping details). In addition, the reusable nature of the tags means that they are transferred between items and continually re-written, making tags more efficient and flexible than the barcode.

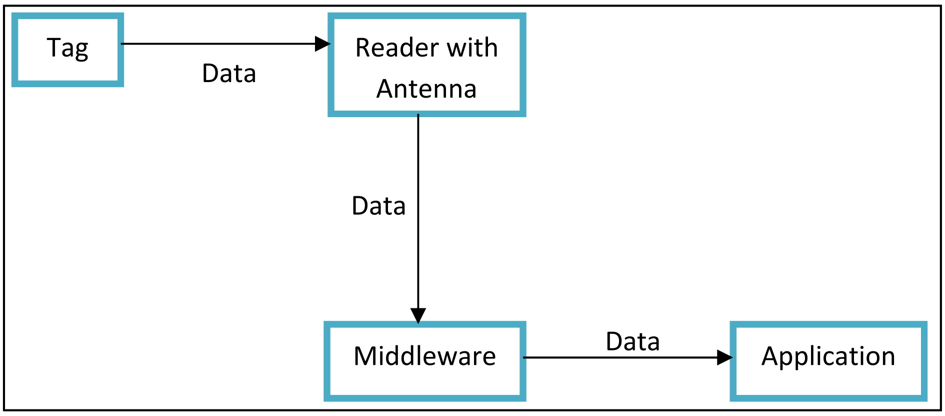


Figure 2.1 Components of a RFID System Illustrating Data Transfer

The advantageous attributes of RFID offers advanced information sharing capability, which a LSP can exploit by providing the appropriate IT solutions and integration techniques in order to create value added advantage (Lin, 2009). RFID is viewed as a crucial technology for application amongst CEOs of the largest LSPs in North America where it is expected that as the number of LSPs and clients collaborating on RFID implementation increases, more successful cases will be prominent, encouraging further adoption (Lieb and Bentz, 2005, Lin, 2008, Collins, 2005b). DHL, one of the world's largest LSPs, assert this sentiment through the continuous development of RFID applications, this is explored further in section 2.1.4. Within DHL's RFID development, it is observed that one area of interest for further research in RFID technology is the application to returnable transport equipment (RTE).

RTE is an important and essential component of the supply chain (France and partners, 2007). However, the management of RTE is challenging as often the assets are not individually managed, hence there is an absence of information concerning the location, status and utilisation rate of each

RTE. This leads to network delays, excess inventory, asset damage and inefficient utilisation rates, which is also verified by a case study of DHL, presented in section 2.1.4 (Meng et al., 2010). RFID makes assets more manageable and assists in advanced levels of network automation, improving supply chain competitiveness. However, literature from the perspective of a LSP is often disjointed and scattered with much of it survey-based, lacking any theoretical development or real-world application (Selviaridis and Spring, 2007).

This chapter explores RTE management practices in the supply chain, investigating advanced technology application and decision-making techniques. The focus is on reducing problems and costs whilst providing the level of agility required for RTE networks to cope with unexpected events quickly and efficiently. Section 2.1 introduces the RTE management problem and presents a DHL Exel Supply Chain case study.

In section 2.2 the application of RFID in supply chains is explored with a focus on the impact on LSPs, outbound logistics, reverse logistics and decision support systems. Aggarwal and Fosso Wamba (In Preparation) (Appendix A-11) explore the multi-level benefits of RFID technology by investigating the possibilities for a LSP to gain advantage on operational, tactical and strategic levels. Aggarwal and Lim (In Preparation) (Appendix A-10) evaluate the importance of the role of RFID application in outbound logistics. This is followed by an investigation by Aggarwal and Lim (2009) (Appendix A-9) into RFID application in reverse logistics, where the focus is on the integration of reverse flows into the supply chain and the impact of creating a smoother transition. Finally application of RFID to enhance decision support systems is explored in the application to a waste management system by Nielsen et al. (2012) (Appendix A-5) where the RFID data is used to bring value-added benefits to the waste collection business.

Section 2.3 addresses RFID application research by investigating: the impact of RFID enabled RTE as a supply chain tool; competitive advantage for LSPs, the creation of automation opportunities; the concept of autonomous logistics and decentralised decision-making. Aggarwal and Lim (2010)

(Appendix A-8) assert that RFID can be utilised as a value adding solution to assist with the smooth flow of RTE, providing supply chain efficiency and effectiveness, benefiting both the LSP and their clients. Further investigation into the opportunities to gain competitive advantage with RFID enabled RTE is explored by Aggarwal and Lim (2011) (Appendix A-6). After applying competitive advantage theory to LSPs using RFID enabled RTE, a cost and value advantage framework is presented by Aggarwal and Lim (2013b) (Appendix A-1). Section 2.3.3 investigates the functionality and possibilities of RFID enabled RTE to create automation. Aggarwal et al. (2011) (Appendix A-7) present a three-level framework in regards to data retrieval, physical processes and cognitive features. Section 2.3.4 explores the concept of autonomy where Aggarwal and Lim (2012b) (Appendix A-4) discuss a product-centric approach of RTE management and related activities by using the RTE as a network improving tool. Further to this, Aggarwal and Lim (2012a) (Appendix A-3) identify the natural alignment between the supply chain and decentralised decision-making due to its distributed nature, in section 2.3.5. The chapter is concluded with a discussion in section 2.4.

2.1 Returnable Transport Equipment and Management

Returnable transport equipment (RTE) forms an essential part of the logistics network and usage is set to continue to rise (Ilic et al., 2009). This is partly due to mounting pressure from governments and consumers amid growing concerns of the negative effect of packaging waste on the environment (Prendergast, 1995). As LSPs are pressed to shred disposal packaging, the supply chain is extended as RTE return flows are introduced. These flows inflate logistics costs due to increased handling and storage space requirements for the RTE (Wu and Dunn, 1995). On the other hand, as the packaging is reusable, the disposal costs are lower (particularly, if the RTE is recyclable), therefore a reduction of supply chain costs may be achieved, however this requires further investigation. In this section, RTE is defined and the main functions explored. An overview of the RTE management problem is presented where the application of RFID to RTE is discussed, providing examples of improvements to logistics networks and an explorations of the additional value streams to be gained, leading to an increase in network competitiveness. A case study of DHL Exel Supply Chain is introduced where site visits and interviews with managers asserts the RTE problem and potential of RFID application. The section is concluded with a discussion.

2.1.1 Definition

RTE , which is also be known as returnable transport items (RTIs), is defined by the International Organisation for Standardisation as "*all means to assemble goods for transportation, storage, handling and product protection in the supply chain which are returned for further usage, including for example pallets with and without cash deposits as well as all forms of reusable crates, trays, boxes, roll pallets, barrels, trolleys, pallet collars and lids*" (LogicaCMG, 2004). RTE may also be referred to as logistical packaging or supply chain packaging as it protects goods throughout the supply chain (Twede, 1992, Duhaime et al., 2001). In this thesis, RTE refers to any equipment that is continuously used in a cyclic fashion within the supply chain to transport or store goods. A single cycle of the equipment begins and ends at the distribution centre.

2.1.2 Function

RTE has multiple functions, which increase when integrated further into efficiency enhancing supply chain roles. Firstly, the function of RTE is to move, store and handle products. The product and RTE are required to be moved to the right place, at the right time and in the right condition. Secondly, the equipment needs to protect the goods and act as a stable packaging format. Thirdly, the RTE needs to be reusable with supporting logistics systems in place to enable efficient returns in order to compete with one way disposable packaging. This is particularly important as the environmental concerns of packaging are increasing (Prendergast, 1995). However, it is in the latter function, inadequate support systems, which is leading to a draining of resources throughout logistics networks and requires further investigation.

2.1.3 RTE Research Development

There is limited work in the area of RTE management which is very surprising considering the importance of the area in regards to cost reduction, improved environmental impact and operational logistics efficiency (Kye et al., 2013, Karkkainen et al., 2004, Johansson and Hellström, 2007, Hellstrom and Johansson, 2010, Carrasco-Gallego et al., 2012, Fleischmann et al., 2001, García-Arca and Prado-Prado, 2006).

	Focus Area	Method Used	RTE
Kroon and Vrijens (1995)	Design of a return logistic system	Case study and optimisation model	Containers
Mckerrow (1996)	Costs and benefits analysis List of areas where it works well	Literature Review	Various
Fleischmann et al. (1997)	Quantitative models for reverse logistics	Literature review	Various
Hanh (2003)	Management of empty containers	Industry report	Containers
Aberdeen Group (2004)	Benefits of RFID-enabled RTE management	Industry report	Various
Kärkkäinen et al. 2004	Case studies on rotation management models	Case study	Various
LogicaCMG (2004)	Benefits of RFID-enabled RTE management	Industry report	Various
Twede and Clark (2004)	Analysis to manage relationships and operational aspects	Case Study	Containers
Angeles (2005)	Effect of RFID visibility on third-party assets	Case study	Trailers
Evers et al. (2005)	Benefits of sensor networks for RTE management	Case study	Various
Mollenkopt et al (2005)	Costs analysis of RTE versus disposable system	Quantitative study	Various
Breen (2006)	Customer compliance in RTE management	Qualitative survey	Various
Johansson and Hellström (2007)	Value of asset visibility for RTE management	Case study and simulation	Roll-containers
Martínez-Sala et al. (2009)	Benefits of active RFID for RTE operations	Qualitative study	Roll-containers
Hellstrom and Johansson (2010)	Decision models for efficient RTE control strategy	A Simulation Model	Various

Table 2.1 Presents RTE Based Literature (Adapted from Ilic et al. (2009))

The areas investigated focus on certain packaging areas, cost analysis/evaluation, environmental perspective or the location of depots. There is a lack of research investigating the potential of making decisions on an RTE level. Hellstrom and Johansson (2010) create decision models for the purposes of how to control the RTE generally and base it on cost, however there is a lack of development on critical decisions of the quantity of RTE needed at each location in order to make the whole distribution process more efficient. Other works, for example, Twede and Clarke (2005) and Angeles (2005) use case studies to shed light into certain aspects of RTE such as relationships and RFID applications but there is a lack of work investigating decision-making on an RTE level. In addition, much of the research is desk-based and lacks quantitative studies and models. This thesis addresses all these issues and fills this gap in the literature by investigating the ability to provide a model to enable effective decision-making on a RTE level. An overview of the problems are explored in the next section.

2.1.4 RTE Management Problem: Overview

The four main problems identified from the literature concerning RTE usage are costs, shrinkage, ownership and cycle times (Rosenau et al., 1996, France and partners, 2007, Johansson and Hellström, 2007, Ilic et al., 2009). Firstly, comparing RTE networks to one way-packing systems there are additional costs, such as RTE purchase, sorting and transportation returns, maintenance, repair, replacement and tracking (Rosenau et al., 1996). RTE is handled more than traditional one-way packaging: the cost of handling needs to be reduced in order to balance the costs in favour of a RTE system.

Secondly, shrinkage is another critical issue in RTE management (Johansson and Hellström, 2007). A case study of Arla Foods highlighted that 10% of their RTE fleet was lost annually costing 2 million Euros to replace. The causes of losses were due to the lack of information regarding the quantity and location of RTE in the pipeline and in stock. Bottani and Bertolini (2009) describe the pallet system in place at an Italian manufacturer, GranMilano, where most of the 300,000 pallets used by the company each year are damaged or the pallets collected belong to a different company, which does not conform to the requirements of GranMilano. Familia Sancela, a South American manufacturer, ships 250 pallets every day and losses are occurred due to: errors in the manual handling system; misplaced pallets at the plants and distribution centres and security problems, which lead to pallets of goods being stolen (Swedberg, 2009a). KiMs, a Danish manufacturer, experienced shrinkage of its 100,000 pallets used per year due to the inefficiency of the manual system in place. Where written lists could not provide the accuracy of information and level of visibility required to manage the pallets (Khan, 2004). As data is gathered and entered into the system manually at each point, the human errors led to poor information, increasing shrinkage levels.

Thirdly, the ownership of the RTE is problematic in terms of the distribution of costs and benefits (Rosenau et al., 1996). RTE is often supplied by LSPs where the equipment is pooled (LogicaCMG, 2004). By definition within a RTE pool the equipment is owned by the pool operator who has the responsibility to ensure that clients have the required RTE (France and partners, 2007). Often in RTE pools, the responsibility of maintaining and controlling the RTE in order for it to fulfil its function has been unclear (McKerrow, 1996).

Finally, without an efficient tracking system RTE lies idle at various points throughout the network creating bottlenecks, RTE shortages and inefficient asset utilisation (France and partners, 2007). RTE becomes a viable packaging option if it is used a certain number of times, otherwise under-utilised RTE becomes a greater burden on the environment, therefore the RTE rotation rate needs to be increased (Kroon and Vrijens, 1995, Ilic et al., 2009). RTE management is difficult to control and manage effectively due to the problems highlighted above and in order to increase the efficiency of RTE, effective tracking systems are required (Johansson and Hellström, 2007). DHL Exel Supply Chain is exploring the application of RFID technology to RTE in order to produce efficient and effective tracking systems throughout their logistics networks, which is discussed further in the case study in section 2.1.4.

2.1.5 **Auto-identification: RFID Technology**

Visibility is a necessity throughout the whole supply chain where identification of objects and flows are required. The process for identifying an object and aligning it with its corresponding information on a database is known as identification which can be done either manually or automatically (Ilie-Zudor et al., 2011). The "automated extraction of the identity of an object is referred to as Automated Identification (Auto-ID) (McFarlane and Sheffi, 2003). Figure 2.2 illustrates a number of technologies which can be implemented in order to automate the identification of objects. Each technology is described in great detail by Finkenzeller (2010) in his book titled "RFID Handbook". The most popular Auto-ID technologies used in logistics are RFID and Barcode.

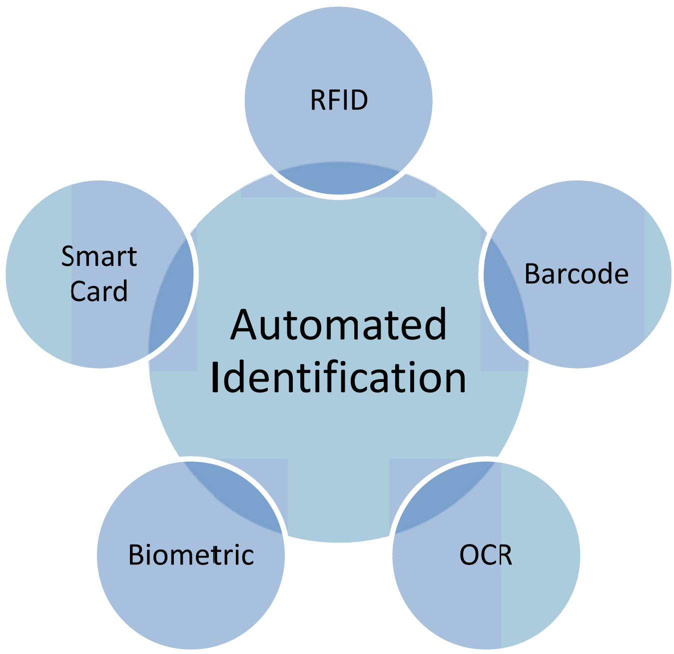


Figure 2.2 An Illustration of the Different Auto-ID Technologies

Although barcodes are extremely popular, when applied in the supply chain there are some limitations such as limited data capacity, requiring line of sight and lack of robustness. RFID is a wireless technology which uses radio frequency waves to identify objects and transmit data. The technology can overcome the limitations of the barcode by being read in any direction and in a stacked environment. The tags are more robust, and therefore can withstand the harshness of the environment. Also there is more data capacity on the tags so more information can be placed onto the tag.

RFID technology has a long history which can be traced back to 1906 when Ernst F.W. Alexanderson demonstrated how radio waves can be formed and the transmission of radio signals (Domdouzis et al., 2007). One of the first applications was developed during World War II with a system called Identity Friend or Foe. This system enabled the British to identify which aircraft were their enemies. The development of this system enabled the widespread application of RFID (Peppa and Moschuris, 2013). Further development of the technology was conducted during the late 1990s when "the researchers at the Massachusetts Institute of Technology (MIT) Auto-ID Center were investigating how RFID can be used as a new way to track and identify objects as they move between physical locations. The centre focussed on reducing the cost of manufacturing RFID tags, optimising data networks for storing and delivering larger amounts of data and developing open standards. When the Auto-ID Centre officially closed on 26th October 2003, the centre's task was taken over by EPC Global" (Holloway, 2009).

COMPONENTS

A RFID system has four main components to enable full application as illustrated in Figure 2.1. When requested, or if the tag is active, information is transferred from the tag to the reader via radio waves. The data is then transferred to middleware to be processed. Then the processed data is used in the application for which it is required for example, monitoring temperatures of chilled foods.

Tags

RFID tags have a microchip, which stores data, and an antenna, which transmits radio waves (Angeles, 2005). The tags can be as large as 2 square inches or as small as "tiny coded beads invisible to human eyes" (Domdouzis et al., 2007). They are also highly durable as they can be broken and can be bendable and still function (Xiao et al., 2007).

There are 2 main types of tags; Active and Passive as illustrated in Table 2.2. Passive tags are more widely used as they are cheaper and have a long usage time (Attaran, 2007). The tags can be as cheap as $0.05 with the potential to be $0.01 (Homs et al., 2004). An active tag on the other hand is more costly as it has a battery. This can limit the life of the tag and tags can cost up to $20 (Attaran, 2007). Table 2.2 illustrates the difference in the 2 main tags such as difference in frequencies, power, read range, speed and cost. These characteristics impact the type of tag used for each application.

TAGS		
Data Storage	Read Write	Read Only
Frequency	Low 125 KHz-134 KHz Range <0.5M Data transfer rate< 1 kbits/s	High 13.56 MHz Range<1.5M Data transfer < 25 kbits/s
	Utlra High 850 MHz–950 MHz Range<100M Data transfer rate <100 kbits/s	2.45 GHz Range<10M Data transfer rate <100 kbits/s
Power	*Passive* Current is induced in the tag's antenna via waves sent by a reader.	*Active* Battery power is used from the tag to offer continuous power to the chip.
Read Range	Short <0.5m	Up to 3m
Required Signal Strength (reader to tag)	High	Low
Available signal strength (tag to reader)	Low	High
Multi-tag reading rate	Scan 100s of tags within 3m from reader and 20 tags moving at 3m.p.h.	Scan 1000s of tags from one reader and 20 tags moving at 100 m.p.h.
Cost	$0.05 to $ 0.25	$4 to $20

Table 2.2 Table Presenting Different Characteristics of RFID Tags (Adapted from (Domdouzis et al., 2007))

Readers

Another key component is a reader which consists of antennas, a transceiver, and a processor. Readers energise the tags and send and receive signals. The transceiver and processor then encodes and decodes the data. "Examples of readers include retailer self-check-outs, library book sensors, security exit sensors, sorters, and portable sensors. A reader has a finite range called interrogation zone, and the size of the zone depends on applications. Passive tags, relying on outside power have weak signals so that they have smaller interrogation zones, whereas active tags have larger interrogation zones" (Xiao et al., 2007).

FUNCTIONALITY - BARCODE VS RFID

The barcode is the most widely used Auto-ID, however there are limitations such as it requires the line of sight to be read and it can be ineffective in certain environments (not robust) (Attaran, 2007). RFID can eliminate the need for human interaction in the process of automated identification and read the tags without the line of sight as well as through materials (useful characteristic in a stacked environment) (McFarlane and Sheffi, 2003). In addition, there is a limited amount of data which can be stored on a barcode whereas on a RFID tag this issue is alleviated (Xiao et al., 2007). All these characteristics make the application of RFID to logistics a very exciting one as illustrated in Figure 2.3.

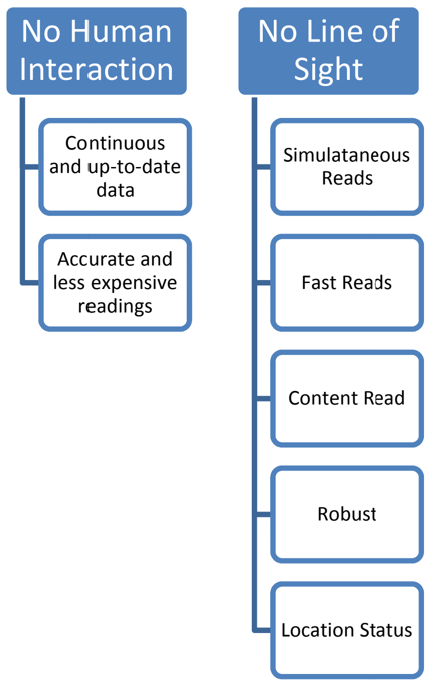


Figure 2.3 Illustration of the Benefits of using RFID in Logistics (Adapted from McFarlane and Sheffi (2003))

RFID CHALLENGES

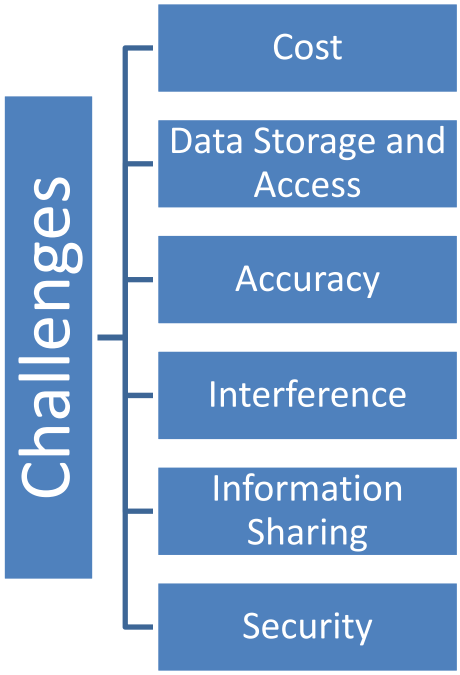


Figure 2.4 An Illustration of the Challenges of RFID Implementation (Adapted from McFarlane and Sheffi (2003))

There are various challenges (see Figure 2.4) in regards to the adoption of RFID however one of the main perceived challenges is the cost. It is essential to make a business case for any adoption of a technology. As the cost of tags is low and they can be re-used, there is "already feasible application cases for high-end goods and closed loop system for assets"(McFarlane and Sheffi, 2003).

The vast number of advantages makes the business case for using RFID in closed loop systems for asset management even more prominent. There are many layers of advantages. Firstly, RFID can add value to a business by giving visibility to all stakeholders in a supply chain of all the flows and enabling the installation of troubleshooting and continuous improvement strategies (Angeles, 2005, Zhu et al., 2012). Secondly there are a vast amount of advantages that can be achieved through various applications. (Attaran, 2007) puts together a comprehensive list of the advantages of using RFID in the supply chain:

- "enhanced visibility into customer needs;

- enhanced visibility along the supply chain;
- accurate and timely asset tracking;
- smart product recycling;
- streamlined or better managed business processes within the company;
- improved productivity by generating the fastest and lowest cost method of acquiring the data;
- improved velocity by responding to demand signals faster;
- better utilization of fixed assets, resulting in lowered capital asset requirements;
- reliable and accurate order forecasts;
- reduction in inventory costs including stock-out and holding costs;
- improved technology return on investment;
- improved accuracy by reducing the opportunity for human error;
- increased productivity and dramatically reduced operating costs;
- improved product quality and reliability including traceability;
- improved supply chain management by better tracking transportation and warehousing channels;
- improved counterfeiting identification, theft prediction, and faster recalls; and
- gaining favour with retailers to better position products on shelves".

Thirdly, RFID can be combined with other technologies such as GIS, GPRS and GPS to plan and schedule within the supply chain (Zhu et al., 2012) . This increases the data retrieval and can enhance the decision-making at each point in the supply chain.

BENEFITS OF RFID AND RTE APPLICATION

RTE is a closed loop system with assets which are ripe for RFID application. The problems with RTE are lack of visibility, missing items and poor rotation. The visibility gained through RFID application can potentially alleviate some of these very cost and inefficient issues. Figure 2.5 illustrates the benefits of RFID application to RTE.

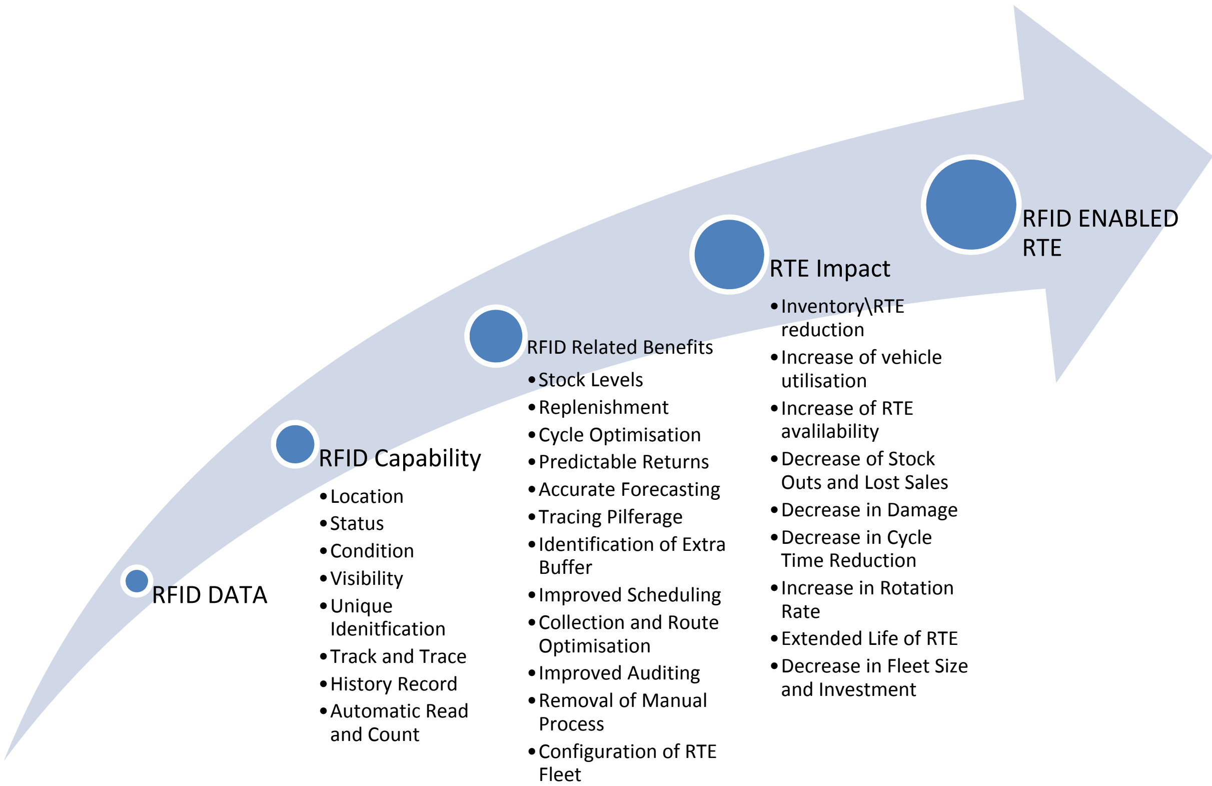


Figure 2.5 Illustration of the Benefits of RFID Data to Create RFID Enabled RTE (Adapted from (Demir, 2010)

There are various operational and economic challenges of RTE such as the cost of RTE, the lack of visibility, and the need to re-use as much as possible. The barcode is not a practical resource to use in order to offer full-time visibility as it cannot be read without the line of sight and it is not as robust. However, RFID can offer high levels of visibility with limited human interaction and the ability to read the RTE in any orientation.

This thesis utilises this functionality of RFID enabled RTE and develops on it to include decision-making elements. DHL Exel Supply Chain is exploring the application of RFID technology to RTE in order to produce efficient and effective tracking systems throughout their logistics networks, which is discussed further in the case study in section 2.1.4.

2.1.6 **Case Study: DHL Exel Supply Chain**

DHL Excel Supply Chain is interested in improving RTE management, particularly with the use of RFID technology. A presentation conducted by Marcel Schirmer, a senior project manager for DHL stresses the importance for one of the world's most expansive LSPs to research into RFID application. Schirmer focuses on the work supported by the DHL Innovation Centre in Germany; a copy of the presentation is in Appendix B-2. The centre in Germany has a comprehensive strategy in place in order to develop the technology which includes: introduction of pilot schemes, encouragement of projects with full implementation and software and hardware support. Clients have indicated that their interest in RFID application is increasing, particularly with the introduction of product regulations across various industries, however there are a few concerns: the cost of tags, the reliability of read rates and restrictions imposed when following RFID standards. Despite these concerns, it is clear that competing LSPs are engaging in RFID development by working with RFID experts in testing centres. DHL have followed suit in attempt to lead the field of RFID application in logistics by developing a research base in the DHL Innovation Centre.

The four key advantages that Schirmer advocates from RFID application are:

- Items can be scanned in any direction and position;
- Multiple tags are read simultaneously;
- Tags are highly durable particularly in dirty workplaces;
- Tags are inherently built with higher functionality levels than the barcode due to features such as re-writing and propensity to combine with sensors.

These advantages result in direct company benefits through: enhanced quality control, reduction of manual processes, reduction in the number of errors and increased visibility. DHL are utilising these advantages in order to promote RFID as a replacement for barcodes particularly to clients in the retail and electronic sectors.

Sony, an electronic company, is a client of a project where DHL are investigating RFID application to increase visibility. They plan to use this in order to better connect the product flow from the Sony Central Warehouse in Tilburg to the customer. Another project, which utilises the sensor component of RFID, is the application of Freshtime RFID tags to monitor the temperature of foodstuffs and analyse the data in order to determine the potential shelf-life of the food. DHL is also conducting a project with Metro France to apply RFID to a network of 1.3 million pallets per year in order to provide tracking and tracing and improve practices in the outbound section of the supply chain; from the distribution centres to the Metro retailers.

In 2005, DHL acquired Exel Plc who had completed tests and trials in RFID application, details of which are found in Appendix B-3. The read rates of a variety of tagged products were tested. For example, keg achieved a 100% read-rate whereas items such as cans had a much lower read-rate at 55%. In addition to conducting experiments on the application potential of the tags, multiple trials were implemented in the UK, for example at the House of Fraser. Garments were tagged at a supplier and monitored as they moved from Hong Kong to the UK. As garments left or entered facilities, the relevant databases were updated and more efficient stock-taking practices were introduced. The research highlighted that features to consider in the future are antenna design and spacing of items, particularly if tagging occurs on an item-level.

DHL Exel Supply Chain (UK) continue the development of RFID applications through pilot schemes and implementation, but this is not at the speed of the projects conducted at the innovation centre in Germany, and therefore much further research is required in the UK in conjunction with industry. During a site visit to a retail distribution centre (RDC) in Long Eaton, it was observed that Marks and Spencer uses RFID tags on a range of garments. Warehouse operators tag the most expensive items of clothing before they are shipped off to store. After meeting one of the managers of the RDC, it was stressed that the main reason Marks and Spencer requested RFID tags on these particular items was to aid in stock-takes in store. DHL, as a LSP interested in developing business opportunities, suggest exploration of RFID application to other areas. The potential is further highlighted when the size and throughput of the RDC is observed, details are available in Appendix B-4. The RDC is 250, 000 square feet and conducts up to 350 deliveries per week across 62 stores. RFID technology potentially aids in improving the processes for the millions of items, which move through the RDC including: put away, store, pick, load and despatch, going beyond the current application.

Another client with interest in RFID technology application is Homebase. A transcript, which is presented in Appendix B-1, details the problems surrounding Danish trolleys that the company uses to move flora through its networks. A manager stresses the following problem areas:

- No daily account of trolley stocks at sites;
- No account of the quantity of trolleys in use at each site;
- No record of which supplier each trolley belongs;
- No account of whether the right trolley is going back to its owner;
- No control of monitoring the type and quantity of trolleys which are taken by drivers;
- Poor tracking is leading to inefficient inbound and outbound processes and trolley losses of which for every 20 trolley bases lost results in approximately £1000 deficit;
- No real-time forecasting available on how many trolleys are required at the distribution centres each day;
- No confirmation from stores that they have received the trolley they require;
- No record of dates for which trolleys need to be returned;
- Lack of agility in the network to enable the movement of trolleys between stores and distribution centres;
- No accurate billing procedure, which is required to be sent to stores for lost trolleys;

This account gives valuable insight into the types of problems companies face with RTE management. At each stage, RFID technology application aids in alleviating or resolving these problems. An RFID tag placed on each trolley offers a unique ID to be assigned to every piece of equipment enabling full visibility through constant tracking and tracing. The advantageous attributes of RFID are encouraging companies to seek application advice from LSPs who must develop expertise in the potential of the technology. Therefore, further discussion and exploration is required as to the potential to use this technology in RTE management and the gain in value for the client and the LSPs, hence this project was developed in conjunction with the RTE department at DHL Exel Supply Chain (UK).

Due to personnel changes, the management at DHL in charge of the project were not able to contribute to the project as initially specified. Therefore, the project continues without any further contribution from the RTE Department from DHL Exel Supply Chain (UK) beyond the contribution specified in section 2.1.4. The knowledge gained from the initial stages of the research conducted at DHL is utilised by firstly analysing the information and then exploring the academic literature in order to place the industry experience gained with the current academic research in the remainder of the chapter.

2.1.7 Discussion

LSPs are required to develop their expertise in logistics and related IT capabilities in order to retain their clients and gain new business. A common problem experienced by clients is with their RTE networks. The RTE networks consist of multifarious equipment, which is used to move, store, handle and protect products. There are problems of underutilisation and losses where one study reported a loss of up to 50% of RTE, which creates further knock-on effects to the whole network until the RTE is replaced (Ilic et al., 2009). As RTE forms a closed loop due to their re-useable nature, efficient support systems are required in place to relay the information and conduct decisions as to their utilisation, which is lacking in industry application. A key technology to improve this area of logistics is RFID.

RFID technology was utilised during the second World War where it was used to identify between friendly and foe aircrafts (Spekman and Sweeney, 2006). However, despite decades of existent, costs and technical problems are cited as the dominant reasons for lethargic uptake of this technology in different application areas (Spekman and Sweeney, 2006). RFID application in the supply chain is relatively new, however interest amongst academics and industry practitioners is increasing by the virtue of technological potential (Mehrjerdi, 2008). RFID technology is a tool, which increases visibility and provides the information flow required to assist in the improvement of logistics operations (Delen et al., 2007, Melski et al., 2008, Zhou, 2009). The application of RFID to RTE gives complete and real-time network coverage of all the client's goods and LSP's assets, which can be used to enhance network synchronisation and efficiency.

Re-usability is also an important characteristic of RFID tags which complements the network feature of RTE (Swedberg, 2009c). The application of RFID to RTE has been addressed by a few authors in the literature (Chuang and Shaw, 2007, Johansson and Hellström, 2007, Hellstrom, 2009, Ilic et al., 2009, Martínez-Sala et al., 2009). All acknowledge that RFID visibility increase as the key attribute, which leads to significant RTE network management improvements. The applications in industry mirror these findings as companies such as Wal-Mart, Metro Group, Marks and Spencer, Volkswagen and Mercedes-Benz have observed significant improvements through the reduction of RTE losses and cost savings (Jonathan, 2005, Collins, 2006, Wasserman, 2010).

RFID has been used to make improvement in RTE networks in various ways for example, to reduce safety stock stocks and shrinkage, which is costly (Swedberg, 2009b). Hellstrom (2009) observed that tracking had enabled the removal of shrinkage at Arla Foods. Another critical area of RTE management is the reduction of cycle times in order to increase the rotation rates, which makes the RTE a more viable and cost efficient investment (Ilic et al., 2009). A study carried out by Ilic et al. (2009) observed a 16.5% increase in the amount of extra trips that each RTE conducted. In addition, another improvement highlighted the 14.1% reduction in the number of RTE required overall in the whole network, leading to a reduction of operating costs and the amount of investment required to implement a RTE system. This is particularly significant as bespoke RTE costs up to $2000 (Wasserman, 2010). Also, a study argued how RFID provides critical data, which helps to increase flow efficiency and improve the effectiveness of an array of processes, which leads to an overall system efficiency improvement of 18% (Mo et al., 2009).

After meetings and site visits with DHL Exel Supply Chain, the following observations were ascertained. Firstly, RTE and its management are critical areas for further research in the logistics and supply chain management area. There are many problems which effect both the equipment and the network within which the RTE moves. Secondly, the development of RFID technology does not only stem from technical developments but also the application potential. Expertise is required into the business based application of the technology and the benefits for supply chains needs to be clearly specified, especially in comparison to other auto-ID technologies, creating accessibility of the technology for clients. Thirdly, it is clear that the UK subsidiary of DHL Exel Supply Chain is lacking some expert knowledge in RFID applications in comparison to the Innovation Centre in Germany. The activity development at the Centre does not transfer through to the UK RTE department, and therefore it is necessary for the department to develop internal RFID based knowledge, hence the initialisation of this project.

From the perspective of LSPs, knowledge which is required in relation to RFID technology is how the technology is used to create value added streams to the RTE network. RFID offers the visibility but there needs to be further investigation as to the potential benefits RFID related technology brings to the LSPs, their clients and in encouraging RTE network improvement. This may aid a LSP in attaining and retaining clients where the LSP will provide robust logistics solutions with maximum customer service and cost reduction techniques. Finally, the research that needs to be explored is the potential to improve a network not only on an operational level, but also strategically. Therefore, the key is to improve the decision-making process. This requires extensive research in order to assess how to further utilise the key advantages that an RFID enabled system brings to a RTE network.

After the initial period of collaboration, personnel changes at DHL meant that they were unable to contribute significantly to the project. Information on the status of RFID in the supply chain has largely been gathered from secondary and published sources. An extensive literature review is conducted in order to establish research on RFID enabled RTE with a focus on how to go beyond the provision of an efficient tracking system to an effective autonomous decision support system.

2.2 Literature Review: RFID Application in Supply Chains

This section explores RFID application in the supply chain, focussing on the areas of: LSPs, outbound logistics, reverse logistics and decision support systems.

2.2.1 Logistics Service Providers (LSPs)

A logistics service provider (LSP) "manages, controls, and delivers logistics activities on behalf of a shipper" (Hertz and Alfredsson, 2003). This term is often used in exchange with third party logistics (3PL) provider where "third-party logistics are activities carried out by a logistics service provider on behalf of a shipper and consisting of at least management and execution of transportation and warehousing. In addition, other activities can be included, for example inventory management, information related activities, such as tracking and tracing, value added activities, such as secondary assembly and installation of products, or even supply chain management. Also, the contract is required to contain some management, analytical or design activities, and the length of the co-operation between shipper and provider to be at least one year, to distinguish third-party logistics from traditional "arm's length" sourcing of transportation and/or warehousing" (Berglund et al., 1999).

However, in this thesis it is argued that the term LSP is broader as there are many different types of providers of which one category is 3PL providers. One growing area is fourth party logistics (4PL).

(Bedeman and Gattorna, 2003) claim that a 4PL is "in effect the 'brains' or central nervous system of the participating organization”, and have revised Accenture's definition of 4PL as follows: A 4PL is an integrator that assembles the resources, capabilities, and technology of its own organization and other organizations to design, build and run comprehensive supply chain solutions and which have the cultural sensitivity, political and communication skills, and the commercial acumen, not only to find value, but to create motivating and sustainable deals that offer incentives to all the parties involved”.

The LSP industry is growing with revenues reaching $141 billion in 2011 (Min, 2013). More firms are utilising the expertise from external logistics service providers as a recent survey stressed that over 54% of shipper’s transportation costs and 39% of warehouse operation costs were outsourced (Langley, 2012). Two particular areas which are outsourced are in IT development and RTE control.

RTE has a range of associated issues such as losses and lack of visibility. The costly and devastating operational impact is driving firms to use LSPs in order to support the control of RTE (Carrasco-Gallego et al., 2012). One way the LSPs try to improve RTE management is through IT development.

“Revolution in computers and communication technology are indicated as the main forces causing TPL services to experience explosive growth. Lewis and Talalayevsky (2000) further addressed the implications of advancements in information technology (IT) for TPL development. They observed that the rapid progress in information processing and communication technology supports the outsourcing of logistics activities to third-party firms as it allows buyers and sellers of logistics services to communicate directly over data-rich, easy-to-use information channels, thereby reducing coordination costs and fostering strategic partnerships based on mutually agreed goals. Indeed, the ability to utilize IT has been increasingly considered one of the key determinants of success for these operators (e.g. Langley et al., 2002 and Langley Jr. et al., 2005), advantages derived from IT and the shortage of trained personnel still hinder full-scale IT implementation” (Marasco, 2008).

In this thesis the research addresses a gap in the literature in regards to how LSPs can use their IT and asset management services (RTE) to gain competitive advantage especially as the competition for this area is fierce. One way is by using advanced technology such as RFID.

Aggarwal and Fosso Wamba (In Preparation) (Appendix A-11) explore the multi-level benefits of RFID technology by investigating the possibilities for a LSP to benefit on three levels: operational, tactical and strategic. It is asserted that the demand of LSPs continues despite the recent recession, as recovery in the logistics sector has been swift with costs accounting for approximately 11.1 percent of the world's GDP and the LSP sector accruing a revenue of approximately 550 billion US dollars (Armstrong & Associates, 2010a). The increase in outsourcing is met with high levels of competency as a recent survey stated that 90% of shippers were satisfied with the function of their LSPs (Langley, 2010). The prominent task for a LSP is to add value to their client's business at a level that the client would be unable to achieve on their own, and in addition at a cost to value ratio, which beats their competitors (Berglund et al., 1999). Although a LSP is not perceived as a leader in IT solutions, in order for a LSP to be competitive it must be knowledgeable and develop logistics related IT capabilities and be proficient in offering services such as radio frequency identification (RFID) technology application in order to gain custom (Power et al., 2007, Coyle et al., 2003). RFID offers advanced information sharing capability, which a LSP exploits by providing the appropriate IT solutions and integration techniques in order to create a value added advantage (Lin, 2009).

Aggarwal and Fosso Wamba (In Preparation) stress that although research in the technical area of RFID is expanding through avenues such as antenna and chip development, there are wide gaps in the application area and as Spekman and Sweeney (2006) argue it is rather "anecdotal", hence it needs to be strengthened. As the potential application is so wide spread the research is often disjointed and sparse with limited references from the perspective of the LSP sector. Therefore, Aggarwal and Fosso Wamba (In Preparation) conduct further research into the adoption and implementation of RFID whereby the study draws upon research questions going beyond the research agenda set by Curtin et al. (2007) and applies them to the LSP sector. The questions explored in the paper are:

- What are the drivers for RFID adoption by LSPs?
- What are specific RFID related benefits for LSPs?
- What are the methods to exploiting higher value added benefits at minimum costs?
- What are the critical activities, on a multi-dimensional level (operational, tactical and strategic levels), requiring transformation from a LSP's perspective?

Each question is analysed in detail in Appendix A-11; the main findings from the literature lead to the following assertions. Firstly, one of the main drivers for LSPs to use RFID technology is to meet customer's expectations and remain competitive, retaining market share (Power et al., 2007, Coyle et al., 2003). Although a LSP may be pushed into exploring RFID applications there are a variety of benefits which can be gathered and utilised for further gain. The main benefit is increased data access, which enhances network visibility, enabling LSPs to offer a higher quality of service. The level and range of benefits will depend on the actions LSPs commit with the data. The potential list of benefits as described by Aggarwal and Fosso Wamba (In Preparation), pulls a LSP into exploring RFID application.

Aggarwal and Fosso Wamba (In Preparation) highlight that the two main reasons which hinder application are RFID standards and the projection of implementation costs. For each of these issues a LSP plays a pivotal role in offering solutions. The potential of RFID application in a supply chain format led to the creation of standards, which are being formed by two separate organisations, EPCglobal and ISO, causing discrepancies and confusion, creating a hindrance to apply RFID globally on a large scale (EPCglobal, 2012, Journal, 2005). However, recently the organisations have been working more closely together in a bid to develop standards, which align creating a more supportive environment for RFID practitioners and the encouragement of adoption. This has led to the application of RFID across an array of different industries such as manufacturing, retail, transportation, health care, life sciences, pharmaceuticals, and government where it is expected that the popularity of the technology will continue to grow with a compound growth rate of 21.5 % between 2007 and 2013, which illustrates that RFID application is moving beyond the initial hype and into useful applications (Bacheldor, 2007, Banks et al., 2007, Spekman and Sweeney, 2006). The RFID market is further being supported by IT development with firms such as Microsoft, IBM and Intel providing services and products to complement this growth (Spekman and Sweeney, 2006). LSPs further support this work through testing centres such as the DHL Innovation Centre, to support EPCglobal and ISO in producing harmonised standards and test applications across industries.

The second potential hindrance is the implementation cost of RFID technology (Huber et al., 2007). Aggarwal and Fosso Wamba (In Preparation) assert that a LSP needs to consider several points in relation to acknowledging and quantifying the benefit amplification they offer to their clients. Firstly, a LSP needs to utilise its extensive network coverage and expertise portfolio to lower implementation costs in comparison to a client implementing an in-house RFID system. During the initial start-up, on hardware and systems development and integration, LSPs use their contacts to drive down the costs of installing RFID systems by negotiating with suppliers, a feature which is difficult for a single firm to achieve. Therefore, the initial start-up costs will be lower for the LSP, as with their multiple clients, the system can be installed at a cheaper rate as illustrated by Aggarwal and Fosso Wamba (In Preparation) in Figure 2.6. Although the initial costs are high for RFID implementation this is because the benefits (and hence the savings) are one-dimensional, and only exist on an operational level, therefore more investment is required to move to tactical level of benefits. As the curve illustrates the amount of benefits that are gained when moving to a strategic level is higher than the amount of investment required. In order to gain higher levels of benefits, more work needs to be conducted into utilising economies of scale available to LSPs and developing RFID applications on multi-dimensional levels.

Secondly, it is in the LSPs interest to experiment and conduct research to ascertain the benefits, which are obtained for different types of RFID application. In order to fully gain the benefits of RFID application, it must go beyond a point solution and be applied across a whole supply chain (Spekman and Sweeney, 2006). When RFID is applied in supply chains the application on each level, operational, tactical and strategic, is then fed into each other causing a knock-on effect of encouraging the attainment of more benefits on a multi-dimensional level as illustrated in Figure 2.7. RFID provides the real-time data to enable proficient information sharing on operational, tactical and strategic levels throughout the supply chain (Zelbst et al., 2010).

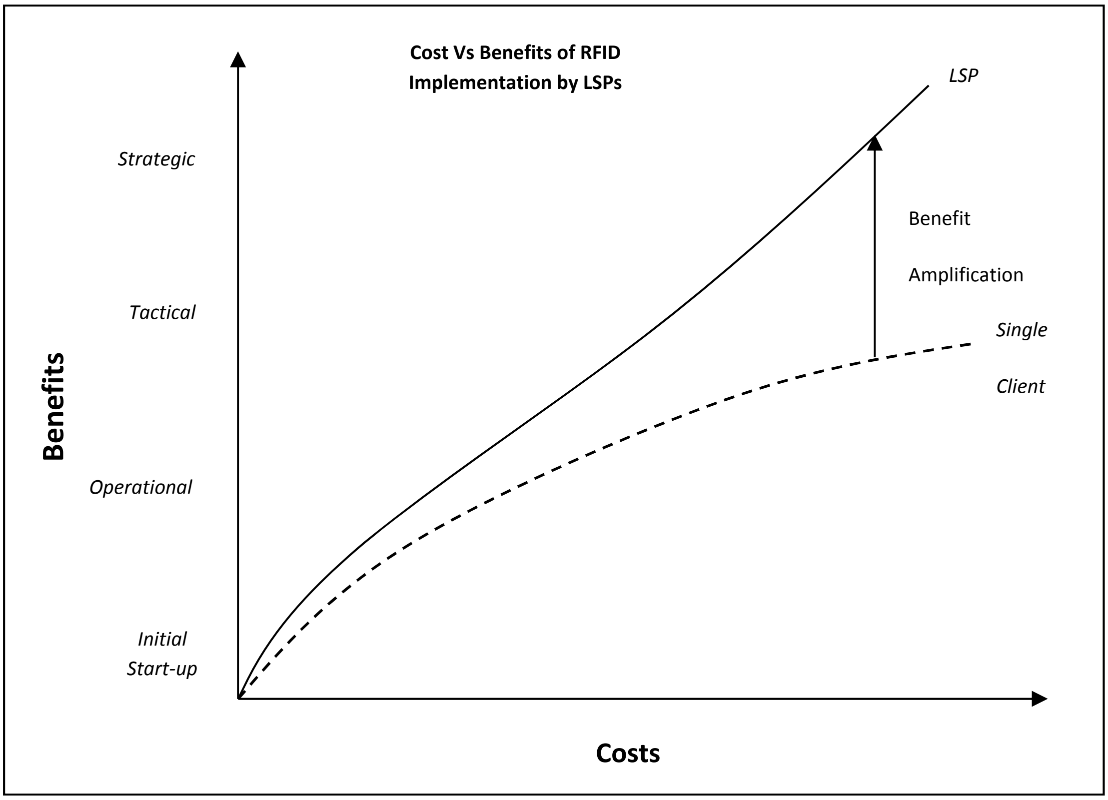


Figure 2.6 The Cost Versus Benefits of RFID Implementation by LSPs

The value that RFID brings is best assessed from a "bottom-up approach", starting on an operational level (Lee and Ozer, 2007). Decisions on an operational level are short-term and concern everyday problems such as correcting human errors or solving the fallout from disruptions. A strong operational base feeds into improving tactical planning, which refers to decisions conducted a few weeks or months in advance, such as transport planning, and demand forecasting (Ivanov, 2010). Strategic advantage concerns creating competitive advantage in order to put the company in an competitive position, such as modifying distribution strategy (Ivanov, 2010). This creates a cycle of amplification of benefits. Another important factor to consider is the technical developments of RFID technology as these will feed directly into the amount of benefits which are retrieved at each level as illustrated by Aggarwal and Fosso Wamba (In Preparation) in Figure 2.7. The higher the quantity of technical advancements made, for example larger storage capabilities, combination with other technologies or longer read ranges, then more proficient applications will be implemented in each area and higher levels of benefits gained. Therefore, it is critical that LSPs not only keep abreast of the technical aspects but work closely with the RFID manufacturers discussing current applications and the technical capability required to bring these into fruition.

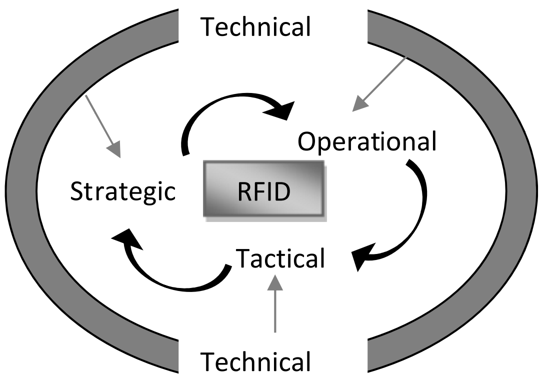


Figure 2.7 The Amplification of Benefits through Multiple Levels and Technical Feeds

Thirdly, in order for LSPs to gain from the amplification of benefits, the benefits must be acknowledged and quantified. As benefits span across many levels and are interwoven, it is challenging to quantify the benefits, therefore Aggarwal and Fosso Wamba (In Preparation) present a tool, which a LSP may use to analyse logistics activities and determine the benefits gained on operational, tactical and strategic levels. Specific issues related to RFID technology such as cost, security, privacy, technical and greening the supply chain, must be researched by LSPs as they have the understanding and expertise in RFID applications and are best placed to solve these concerns. It is concluded that the combination of outsourcing and RFID enabled technologies are critical to achieving advanced levels of logistical efficiency at low costs. To explore the application of RFID further, Aggarwal and Lim (In Preparation) (Appendix A-10) conduct a review of RFID application in outbound logistics, which is discussed further in section 2.2.2.

2.2.2 **Outbound Logistics**

The scope of potential impact of RFID application in logistics, specifically outbound logistics is reviewed by Aggarwal and Lim (In Preparation) (Appendix A-10), where the role of RFID technology in outbound logistics is explored.

Aggarwal and Lim (In Preparation) explore current research of RFID application in outbound logistics, analysing future trends based on industry application potential, creating a framework of critical areas of innovative research streams, encouraging adoption within industry. Aggarwal and Lim (In Preparation) assert that the literature reviews of RFID application are very broad (Ngai et al., 2008, Irani et al., 2010, Chao et al., 2007, Liao et al., 2011, Sarac et al., 2010). Although recently, RFID application in supply chain management has grown, there has not been an attempt to analyse the impact of RFID within different areas of the supply chain in detail, despite the popularity of RFID application within areas such as manufacturing and logistics (Liao et al., 2011, Ngai et al., 2008). Therefore, due to the scattered and inconsistent RFID related literature within the outbound logistics paradigm and the potential for application due to the bountiful activities such as transportation, warehousing, material handling and consolidation, an in depth examination is conducted by Aggarwal and Lim (In Preparation). The paper offers a point of reference for readers, closing the gap between the research being conducted and the industry requirements.

The review is classified into four main categories which are: RFID based technological development, RFID enabled activity support, RFID embedded network integration and RFID empowered value creation as illustrated in Figure 2.8. Each of the categories is described in further detail in Appendix

A-10. Aggarwal and Lim (In Preparation) identify a gap between the academic research being conducted and industry requirements, which results in a lack of focus in the outbound logistics paradigm. Therefore, a framework as illustrated in Figure 2.9 is presented in order to encourage more fruitful RFID exploitation in the outbound logistics paradigm.

The framework presented by Aggarwal and Lim (In Preparation) encourages the reader to focus on four critical areas in regards to RFID application in outbound logistics which are: distribution facilities, transportation, RTE & goods and network flows. Each area has a specific RFID based application of focus. Firstly, the large quantity of activities conducted within distribution facilities, many of which are highly labour intensive, require a focus of automation. In order to fully embrace automation, proficient decision support systems need to be developed. It is beneficial in a large network to decentralise the decision-making in order to achieve optimal solutions and outbound logistics is a prime case scenario for decentralised application research (Lee and Whang, 1999). Agent based modelling provides the means to developing the software capability to automate decision-making to an object level, whilst still maintaining global goals (Hülsmann et al., 2008).

Secondly, data capture and utilisation is focussed on transportation activities. This refers to utilising the opportunities to combine RFID technology with other technologies (such as sensors and GPS) in order to gather valuable data. RFID technology is exploited even further when combined with web-applications via the internet, which enables real-time tracking, encouraging the expansion of possible value-added streams (Violino, 2011, Violino, 2003). Thirdly, for RTE and goods the focus is on an object orientated approach, where RFID application on a RTE level is explored by considering autonomous logistics research (Windt et al., 2008).

Finally, the focus is on utilising the real-time visibility which RFID brings to analyse network flows. The outbound network is a very dynamic environment, which sits within an aggressive supply chain creating many challenges. Real-time visibility is a critical tool to ensure competitive advantage through responsiveness, whilst maintaining lean policies. However, unless the real-time visibility is used in a timely manner then there will be a limit to the level of business value gains for each outbound process. Information through network flows are revolutionised with the introduction of software such as artificial intelligence, as real-time and prompt decision-making functionality utilises increased visibility. The visibility creates a circular motion of potential value creation around the components of outbound logistics as illustrated in Figure 2.9. The framework developed will assist academics in the areas of research requiring further exploration and assist industry practitioners to identify the benefits of RFID and how best to exploit the technology in order to create further value. The focus of this review has been for outbound logistics; however it is also important to consider reverse flows and their integration with forward flows. Section 2.2.3 explores RFID application to reverse logistics.

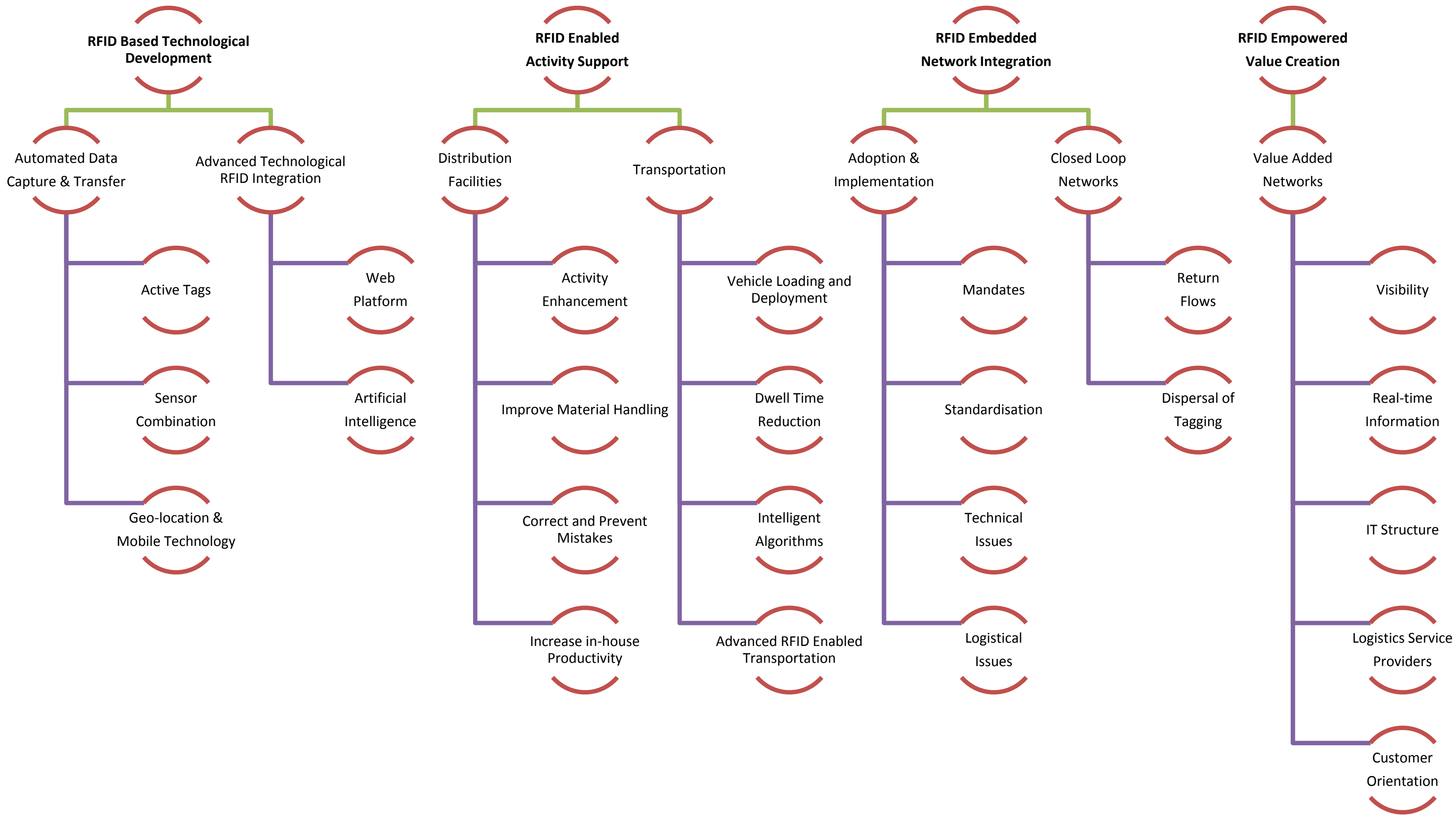


Figure 2.8 Classification of RFID based Literature for Outbound Logistics

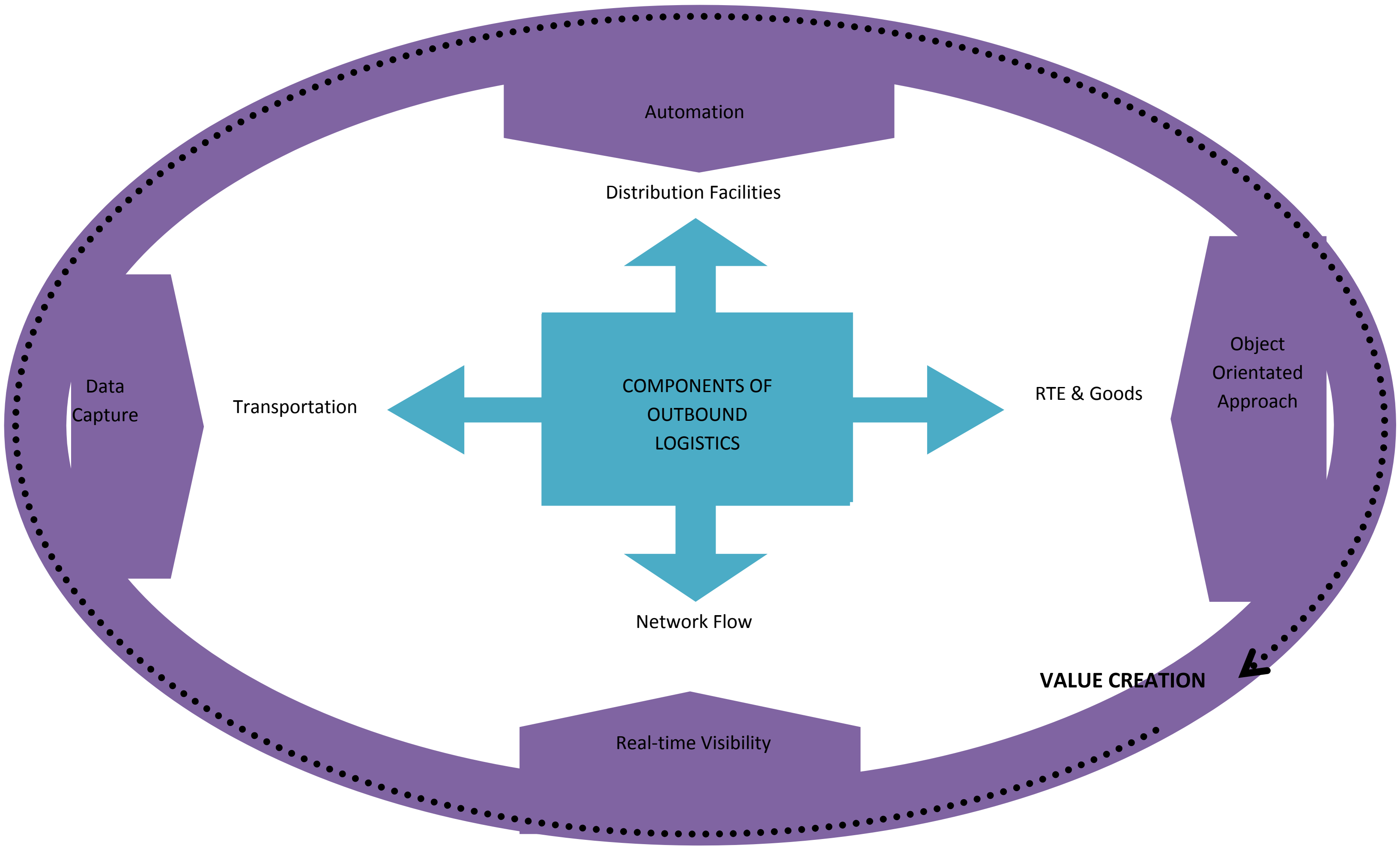


Figure 2.9 Value Creation within Outbound Logistics via RFID Technology Implementation

2.2.3 **Reverse Logistics**

Reverse logistics (RL) is an area, which in the past decade has received growing interest. RL is an extensive area most commonly associated with all the activities that are carried out once a product has been sold to the consumer. RL activities permeate through manufacturing, warehousing and distribution processes where forecasts and goods in the pipeline need to be identified in order to allocate the appropriate quantity of logistics resources. One of the key issues within this area is the lack of available accurate and up-to-date information. This is mainly because once a product is sold the information is usually not updated or it is lost (Visich et al., 2007). If the sales and customer demand figures are inaccurate then all the RL activities would be affected and the inaccuracies would be further amplified as the information goes through the RL network. Due to a lack of product information, the element of decision-making becomes difficult leading to further inaccuracies (Ergen et al., 2007). Once RL activities are integrated upstream the inaccuracies become amplified, which increases the bullwhip effect within the supply chain, and hence decreases the efficiency and effectiveness. Therefore, the bullwhip effect occurs in both directions from the middle of the supply chain due to inaccuracies generated as illustrated by Aggarwal and Lim (2009) in Figure 2.10. In addition, as RL products and information are fed back into the forward logistics supply chain, the inaccuracies are carried through and further amplified. Aggarwal and Lim (2009) (Appendix A-9) assert that reverse logistics is a growing area of the supply chain sector and the exploration of RFID application will give valuable insight into whether the bullwhip effect is reduced as reverse flows are integrated into the supply chain.

Aggarwal and Lim (2009) assert that within the forward logistics paradigm, RFID brings many benefits to the flow of the information in terms of speed, accuracy, efficiency and security which has resulted in the re-design of supply chain processes (Shad, 2000, Li et al., 2006). However, reverse logistics is a different set of processes from forward flows as it extends forward logistics forming a closed loop (Lee and Chan, 2009). Depending on the item in the return flow, it is difficult to predict which path the product will take until it is returned back into the system. Therefore, it is difficult to forecast and have all the appropriate resources in place. RFID application offers a level of visibility enabling the creation of agility in the network allowing the appropriate signals to be sent to the correct facilities in a bid to reduce product dwell time (Lee and Chan, 2009). RFID also reduces the number of inaccuracies downstream, as up-to-date information of quantities in stock and within the RL pipeline are fed into forecast scenarios and information systems upstream, which assists in the decision making process for return centres, leading to smoother operations (Langer et al., 2007). In conclusion, RFID enables more accurate forecasting and higher levels of agility in order to reduce the bullwhip effect created by the integration of reverse flows. The reduction of inaccuracies enables more effective decision-making and the impact of RFID application on decision support systems is explored further in section 2.2.4 on a waste collection system.

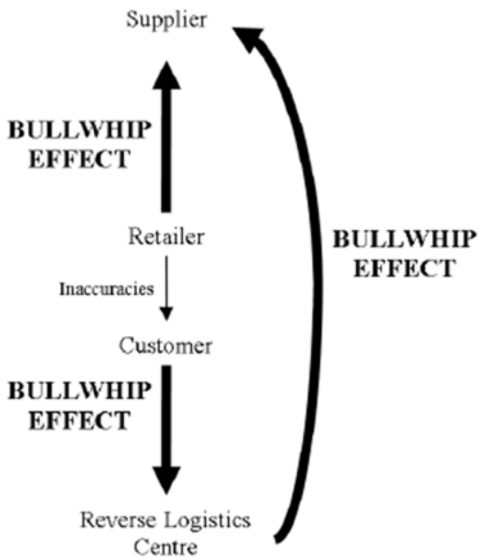


Figure 2.10 : The Bullwhip Effect in a Closed-loop Supply Chain

2.2.4 **Decision Support Systems**

In addition to forward flows there are a few studies supporting the use of RFID technology in order to enhance reverse flows and assist with product renewal and disposal activities, particularly within warehousing (Nielsen et al., 2012) (Appendix A-5). Lee and Chan (2009) state that by using RFID, signals are sent instantly to the sorting centre, which then updates the vehicle scheduling systems thus, reducing dwell time of the product and enhancing efficiency of the distribution network. Langer et al. (2007) observe that the real-time information that RFID provides, assists in the assessment of claims made for returns within centres to facilitate smoother and quicker operations. Jayaraman et al. (2008) observe that switching from barcodes to RFID within reverse channels reduces the average cost of material handling by 30% because the need for manual handling is removed. Literature relating to the use of RFID within reverse logistics to aid in decision-making is rare, however Trappey et al. (2010) propose a genetic algorithmic forecasting approach to assist in the decision support of safety stock and loss rates at different points throughout the reverse chain. Furthermore, research into optimising the transportation routes of reverse flows is mainly limited to developing a vehicle routing model with RFID applications in order to assist authorities in utilising labour and assets during waste collection (Ustundag and Cevikcan, 2007). RFID technology is applied in order to give the real-time location and accurate weight readings of each bin so that the most optimal routes are calculated. However, this study was applied on a small scale of just one vehicle and 15 bins.

The paper presented in Appendix A-5 considers the utilisation of RFID to update and validate information in a continuous manner for a decision support system in order to bring value-added benefits to the waste collection business (Nielsen et al., 2012). RFID enables an enhancement of the decision support system through the following features: being a real-time tracking tool, providing an accurate record of information, accurate billing, improved truckloads and optimised delivery routes and ability to cope with unanticipated demands.

2.2.5 Discussion

Section 2.2 explored the multi-faceted benefits of RFID application to LSPs, within outbound logistics, in reverse flows and the impact to decision support systems. RFID application in the supply chain is relatively new, however interest amongst academics and industry practitioners is increasing by the virtue of the technological potential (Mehrjerdi, 2008). A principle advantage is no line of sight whereby tags are read through materials and in any orientation, which is beneficial in the stacked outbound logistics environment (Finkenzeller, 2010). Another advantage is the function of multiple scans, simultaneously, a time saving attribute, which is favourable for each outbound logistics process (Liao et al., 2011). There is a large list of benefits that RFID application has throughout the supply chain, such as process re-engineering, competitive advantage gain, aligning supply and demand, offering robust and durable tags, increased business intelligence, easier transferral of information, improved traceability and bull-whip effect reduction but to name a few (Ngai et al., 2008, Liao et al., 2011, Chao et al., 2007, Li et al., 2006, Sarac et al., 2010, Mehrjerdi, 2008). In addition, there are some attributes of RFID implementation, which are particularly beneficial for outbound logistics such as reduction in the cost of handling goods, information visibility, tracking large volumes with ease, accurate levels of authentication, speeding up identification, increasing material flows, reduction of out-of-stocks, reduction in shrinkage, reduction of labour costs, reduction of inventory costs, reduction in human errors and improved inventory management (Ngai et al., 2008, Liao et al., 2011, Chao et al., 2007, Li et al., 2006, Sarac et al., 2010, Mehrjerdi, 2008).

However, despite decades of existence, costs and technical problems are cited as the dominant reasons for lethargic uptake of this technology in different application areas (Spekman and Sweeney, 2006). Aggarwal and Fosso Wamba (In Preparation) (Appendix A-11) argue that drawbacks such as standards and costs can be eliminated or alleviated by outsourcing to LSPs. LSPs need to continue to work with key RFID industry players in order to develop the technical aspects of RFID technology so that benefits on operational, tactical and strategic levels are captured.

RFID application research requires greater focus of application in areas of: data capture functions, automation potential, object orientated approaches and real-time visibility utilisation as asserted by Aggarwal and Lim (In Preparation) (Appendix A-10). In addition, there is limited research on reverse flow integration and the impact of RFID application to decision support systems despite RFID offering a real-time tracking tool and accurate information (Aggarwal and Lim, 2009, Nielsen et al., 2012).

Therefore, section 2.3 explores these concepts further through the application of RFID to RTE and assesses its impact as an automation tool and with an object orientated decision support system.

2.3 Literature Review: RFID enabled RTE

RFID technology application offers many advantages throughout the supply chain as discussed in section 2.2. An application area where further research is required is to RTE, which is affirmed by DHL Exel Supply Chain in section 2.1.4. This section explores the concept of RFID enabled RTE from the perspective of gaining competitive advantage through a LSP with a focus on the possibilities of achieving automation and autonomy in the supply chain.

2.3.1 Supply Chain Tool

Aggarwal and Lim (2010) Appendix A-8) highlight that RTE usage is continuing to rise, however research into the potential of RTE within the realm of supply chain management and logistics is limited (Twede and Clarke, 2005). The use of RTE has been highlighted as a cost effective option for the movement and storage of goods, particularly within the retailing sector in the UK and the automotive industry in the US (Twede and Clarke, 2005).

However, one main problem has been the management and control of the RTE (LogicaCMG, 2004). An effective solution of resolving problems and finding better supply chain solutions is to give control of the RTE and related activities to a LSP (Twede and Clarke, 2005, McKerrow, 1996). The use of LSPs entices companies to invest in RTE networks as the financial requirements with installing such a system is spread, and therefore the risk reduced for the clients (Twede and Clarke, 2005). Although the area of LSPs is growing, research from the perspective of the provider is very limited (Power et al., 2007, Berglund et al., 1999).

In some cases it has been reported that companies lose from about 10%-15% of their RTE fleet annually (Johansson and Hellström, 2007), which decreases the amount of RTE available to utilise, and therefore increases replacement costs. RFID is an effective tool to create visibility and assist with efficient tracking (Lee and Park, 2008). The application of RFID to RTE is at an early stage and further research is required (Hellstrom, 2009).

Aggarwal and Lim (2010) (Appendix A-8) observe that in the literature it is insisted that although there are models illustrating the investment potential and gains of a RTE system, there is very little research investigating the cost and benefits of RFID enabled RTE (Hellstrom, 2009, Twede and Clarke, 2005). Additionally, there is even less research investigating the efficiency and effectiveness gains that a LSP brings to a RFID enabled RTE supply chain network. Therefore, further research is required of a LSP managed RFID enabled RTE network from the perspectives of both the LSP and their clients. Aggarwal and Lim (2010) begin to fill this gap by defining RTE and describing its function and associated networks. Particular focus is assigned to the LSP pooling network difficulties such as balancing supply and demand, where demand uncertainty of RTE is a problem (Schmidt and Wilhelm, 2000). The application of RFID is proposed as a value-adding solution to assist with the smooth flow of RTE, providing supply chain efficiency and effectiveness benefiting both the LSP and their clients.

Aggarwal and Fosso Wamba (In Preparation), recommend industry practitioners and academics to build RFID based benefits from an operational level through to a strategic level. Aggarwal and Lim (2010) suggest a top down approach should be deployed when attempting to add value streams through RFID application throughout the supply chain. The prospect of adding value should influence the strategy whereby the RTE network and the goods being carried are considered, in addition to the clients and the LSP itself. One strategic value adding feature may be to increase supply chain agility. On a tactical level this would result in smaller batch sizes required from the client and possibly the addition of more cross-docking facilities by the LSP. On a operational level it would require information of each RTE and goods, at each exchange point. By building value streams from the top-down, the benefits obtained on each level are identified from the bottom-up. The value adding strategy from RFID enabled RTE is discussed further by Aggarwal and Lim (2010) in Appendix A-8, where it is observed that further research is required into competitive advantage theory of RFID enabled RTE from the perspective of a LSP.

2.3.2 **Competitive Advantage for Logistics Service Providers**

Firms are utilising the expertise of LSPs, in particular the development and application of innovative technologies, in order to increase automation on the endeavour to enhance competitiveness (Peppa and Moschuris, 2013). However, LSPs themselves struggle with achieving competitive advantage (Min and Joo, 2006). Langley and Capgemini (2007) asserted in a study that customers of LSPs were dissatisfied with the service they were receiving due to, amongst other reasons, a small range of services and lack of advanced information technology. This stresses the need for LSPs to investigate their competitive advantage and explore the development of information technology and related innovative technologies and services.

Barney (1991) states that competitive advantage is when a firm "*is implementing a value creating strategy not simultaneously being implemented by any current or potential competitors. A firm is said to have a sustained competitive advantage when it is implementing a value creating strategy not simultaneously being implemented by any current or potential competitors and when these other firms are unable to duplicate the benefits of this strategy*".

This section investigates competitive advantage theory in logistics with a focus on LSPs. Resource based view theory and Porter's theory are considered and evaluated in the context of RFID enabled RTE with Christopher's (2005) theory explored in the section to follow.

2.3.2.1 Resource Based View vs Porter's Theory

Penrose (1995) was one of the first scholars to assert the importance of considering resources in relation to a firm's competitiveness where it was stated that a firm is "a collection of resources". It was asserted that a firm's potential for growth is limited by their resource portfolio and that the success of the firm was directly related to the firm's ability to make resources available and utilise them fully. For further detailed development and evidence of application of RBV, refer to (Olavarrieta and Ellinger, 1997). RBV considers resources at the forefront of competitive advantage, and therefore within the firm (Barney, 1991).

Firm's resources are required to have the following attributes (Barney, 1991):

- "it must be valuable";
- "rare among a firm's current and potential competition";
- "imperfectly imitable";
- "cannot be strategically equivalent substitutes for this resource that are valuable".

This will determine the propensity of firm's resource to enhance competitive advantage, however these cannot be applied to IT systems and technological implementations as it would under sell the application potential of these value adding opportunities. The "definition of resources does not tie the existence of a resource to the existence of advantage" (Hunt and Davis, 2008).

Other issues are that RBV does not consider the importance of external impacts, value chains and processes. Process development and value are particularly important when considering IT and technological applications as both external and internal processes within a firm need to be integrated (Lewis and Talalayevsky, 2000). Therefore, for processes there is a disconnection in regards to IT implementation between the resources and the utilisation of resources through the multiple internal and external processes. This can be captured more significantly with value chains. RBV lacks a theory related to value (Bowman and Toms, 2010). Value exploration in competitive advantage is essential, particularly the exploration of the impact of process development when considering the implementation of technology. It is essential to look beyond the firm in order to develop competitive advantage (Hunt and Davis, 2008).

Aggarwal and Lim (2013b) explore the concept of competitive advantage theory with Porter's (1985) concept of value chain management, where all activities conducted by a firm are split into five primary activities and four support activities. The primary activities are inbound logistics, operations, outbound logistics, marketing & sales and service. The support activities are firm infrastructure, human resource management, technology development and procurement. In order for a firm to be competitive it must be a market leader in each of these areas or it is advised to outsource. Leadership is seized from the aspects of cost or service. A firm is a cost leader if they carry out all the activities in the value chain at the lowest cost in comparison to their competitors. Where a firm strives to become a service leader the aim would be to differentiate their products and provide unique services to customers in order gain competitive advantage.

Activities are considered the focus of competitive advantage with Porter's theory as it integrates the environment beyond the firm. Porter considers more the value that a firm can gain from assembling and exploiting their resources, therefore taking into consideration the industry and external influences. This theory develops barriers of entry in order to protect advantage and an industry orientated approach is essential within the competitive nature of LSPs, particularly with the recent recession.

The main issue with using RBV in the context of this research is that it is difficult to define the resources and determine whether they are truly unique, which would not be the case with RFID enabled RTE as they are not rare (Thomas et al., 1999). However, using Porter's theory it is possible to identify characteristics which will focus on what will be significant within the LSP industry and focus on innovative technologies. By being industry orientated it is possible to create an effective value chain and set of processes in order to create barriers to entry and protect competitive advantage.

2.3.2.2 Competitive Advantage Theory Application to Logistics and LSPs

Porter (1985) insists that a company should aim to excel at either becoming a cost leader or a service leader as trying to do both will lead to companies being “stuck in the middle”, and therefore not achieving competitive advantage. However, Christopher (2005) contradicts this by observing that logistics costs represent a high proportion of a firms overall costs and that through the efficient management of these activities, firms reduce costs and differentiate their services in order to become cost and service leaders, putting them in the strongest possible position in the market.

Based on the work presented by Aggarwal and Fosso Wamba (In Preparation) on the impact of LSPs on their clients, it is argued in this thesis that the value-added opportunity of using a LSP to the client is only realised if the LSP can provide a lower cost solution, and therefore increase the cost to value ratio. The concept of adding value for competitive advantage is well explored by Christopher (2005) with work on cost leadership. As presented by Lewis and Talalayevsky (2000), RBV does not consider the “value chain and processes”, particularly with the role of technology application; and hence in the context of LSPs, a Resource based view is less aligned in explaining LSP's competitive advantage. Although Porter’s theory is more aligned, in the context of logistics and providing a service with technological innovation, Christopher (2005) has identified incompleteness of Porter’s theory with respect to competitive advantage. Keeping these views on suitability of other theories, this thesis aligns to the competitive advantage theory of Christopher (2005), which is further elaborated by Aggarwal and Lim (2013b).

In order to achieve competitive advantage for the LSP and the client, the value chain is broken down into cost advantage and value advantage. Cost advantage is where the firm focuses on operating at the lowest cost in comparison to its competitors. Value advantage refers to the uniqueness of the product or service in the market place. Christopher (2005) further categorises how firms gain cost and value advantage through logistics activities.

Aggarwal and Lim (2011) (Appendix A-6) investigate the opportunities for LSPs to gain competitive advantage with RFID enabled RTE. This area of research is explored further by Aggarwal and Lim (2013b) (Appendix A-1) where competitive advantage theory is applied to LSPs and a framework to encourage cost and value advantage through four elements: asset reduction, increase of asset utilisation, RTE cycle-time reduction and utilisation of real-time data.

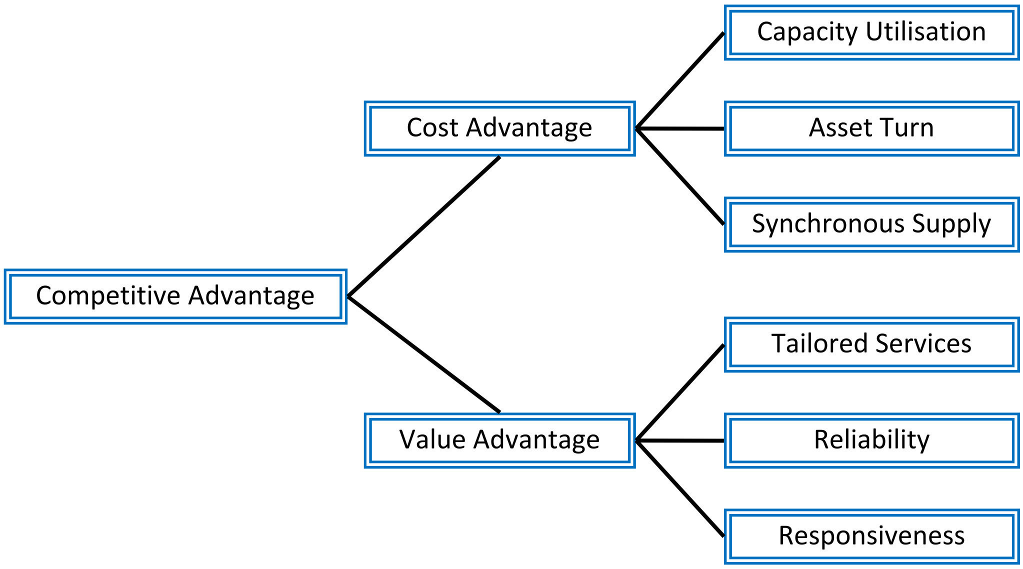


Figure 2.11 Competitive Advantage Theory Adapted from Christopher (2005)

Christopher (2005) insists, cost advantage is achieved through capacity utilisation, asset turn and a synchronous supply chain. Aggarwal and Lim (2013b) suggest that firstly the capacities of the storage facilities, fleet and RTE need to be utilised. The asset costs are reduced by the LSP through economies of scale by leveraging the size of their multi-client network. The LSP keeps the costs low and passes these savings onto the clients at lower rates than the clients would be able to achieve in-house (Damme and Ploos van Amstel, 1996, Sohail et al., 2006). Secondly, often companies outsource high asset activities such as fleet management, warehouse management and shipment consolidation (Dapiran et al., 1996). With the addition of RTE, these assets incur high operating costs and LSPs use their expertise to reduce assets, increase asset utilisation and reduce other high costs such as labour (Bardi and Tracey, 1993). Asset reduction and utilisation occurs through the consolidation of storage facilities, increasing the interchangeable rate of the RTE and sharing the fleet capacity. This increases the network capacity and return on investment of each asset. Finally, synchronicity of the supply chain is only achieved if there is the free flowing exchange of accurate information in a timely manner. RFID readers are installed throughout the networks at any points where the data would be valuable, for example at the suppliers, cross-docking gates and retailers. RFID technology application introduces real-time continuous data points specifying the location of each RTE, and furthermore assists in the optimisation of the fleet and facilities, which improves real-time scheduling.

Christopher (2005) asserts that value advantage is gained through tailored services, reliability and responsiveness. Firstly, LSPs gain value advantage through offering a high number of tailored services and adding value to their client's supply chains (Rajesh et al., 2010). Offering more specialised and tailored services in comparison to competitors, increases value advantage. LSPs tailor services to each client through unique product tracking with the introduction of RFID enabled RTE, even when sharing and consolidating assets. In addition, LSPs have expertise in reducing order cycle-times and lead-times, improving the efficiency of client's supply chains (Bhatnagar et al., 1999, Bask, 2001). Another incentive to offer clients service differentiation is due to the fact that it is five times more costly to attain a new client than to keep an existing client (Knemeyer and Murphy, 2005). Therefore, it is in the LSP's interest in terms of cost saving to have value advantage in comparison to other competitors to maintain business. Secondly, reliability is crucial in order to achieve on time deliveries, in full orders and with no defects in the products. With a time or quantity based consolidation approach the LSP offers reliable lead-times and delivery dates to its clients. The real-time tracking and tracing of RFID enabled RTE gives LSPs the ability to monitor their client's goods. They ensure that items are moving on the right routes and any deviations are resolved quickly and the clients are informed immediately of any delays. In addition, any special requirements, for example temperature control items such as food and pharmaceutical products are handled. Finally, responsiveness is achieved through the introduction of visibility. This aids in the reduction of the RTE cycle time, which further improves client's product lead-times. With a cross-docking consolidation centre, a LSP re-arranges the fleet inbound and outbound at any point. This increases responsiveness and agility as the provider's cycle time is reduced. The level of RTE reliability is improved, encouraging the reverse flows to be as robust as the forward. All the factors considered in this section are aspects of lean and agile theories which are explored further in the next section.

2.3.2.3 Lean and Agile

In this section, the concepts of lean and agile strategy are explored as they have a significant impact on the competitiveness of a firm. Firstly, the concept of lean is described, which is followed by an investigation into agility. Then follows a discussion on the impact of lean and agile practices to supply chain strategies, and the requirements for RTE management, and hence the need for both lean and agile practices for the context of this research.

LEAN

"Lean is rooted in the early work of Frederick W. Taylor's The Principles of Scientific Management" where the idea was to find "the optimal way of optimising production" (Arlbjørn and Freytag, 2013). It was later popularised by (Womack et al., 1990) with the book titled "The Machine that Changed the World". The material presented was based on the practice conducted at Toyota from their production system (Schonberger, 2007). The key focus for this was the elimination of waste, which led to the formation of the seven wastes (overproduction, transportation, motion, waiting, processing, inventory and defects) (Hines et al., 2004)

Lean has now been applied to various other areas in business going beyond the original application of production. With this enlargement of scope, the concept and definition of lean has come under scrutiny as the original "narrow definition" is deemed not applicable in many application areas beyond production (Hines et al., 2004, Arlbjørn and Freytag, 2013). Womack et al. (1990) present an apt definition: "lean production is lean because it uses less of everything compared with mass production – half the human effort in the factory, half the manufacturing space, half the investment in tools, half the engineering hours to develop a new product in half the time".

However, Hines et al. (2004) assert that that the concept goes beyond waste reduction as it includes value creation, which must be considered when applying the concept as this enhances competitiveness. Other conditions to consider are: "production of standard goods/services; large volume (not mass production, but production in smaller series, but still with a large total volume); and relatively long product lifecycle" (Arlbjørn and Freytag, 2013). All these factors are applicable to RTE management as there are standard items with large volumes and lifecycles. However, the implementation of lean cannot account for demand variability, which is critical in RTE management (Hines et al., 2004). Therefore, the concept of agility is explored.

AGLIE

Agility is "the ability of an organisation to thrive in a continuously changing, unpredictable business environment. Simply put, an agile firm has designed its organisation, processes and products such that it can respond to changes in a useful time frame" (Forum, 1994). Christopher (2000) expands this by stressing that it is the firm's responsiveness in regards to volume and variety. In order to achieve this, a critical requirement is the need for full and effective information flow (Mason-Jones et al., 2000). The information flow is just as critical as the material flow in regards to supply chain strategy and the timely retrieval of information is of paramount importance as "once information ages, it loses value".

LEAGILE

Supply chain strategy is critical when firms are seeking to enhance their competitive advantage (Qrunfleh and Tarafdar, 2013). Supply chain strategy is defined "as a set of approaches utilized to integrate suppliers, manufacturing, warehouses, and stores so that merchandise is produced and distributed at the right quantities, to the right location, at the right time, in order to minimize system-wide costs while satisfying service level requirements" (Qrunfleh and Tarafdar, 2013).

The implementation of agile/lean strategies will have a different impact on the supply chain strategies implemented. “An agile supply chain strategy is aimed at being flexible by adapting quickly and effectively to rapidly changing customer needs” whereas a lean strategy is focused on getting it just-in-time with a focus on cost and waste reduction (Qrunfleh and Tarafdar, 2013). “Leanness and agility can sometimes be combined with the strategic use of a decoupling point, thereby capitalizing on the benefits of both paradigms” (Qrunfleh and Tarafdar, 2013). “Leagile is the combination of the lean and agile paradigms within a total supply chain strategy by positioning the decoupling point so as to best suit the need for responding to a volatile demand downstream yet providing level scheduling upstream from the marketplace” (Naylor et al. 1997).

RTE management requires both agile and lean strategy as not only do the waste and costs need to be reduced, which adds value, but also the network needs to be responsive in order to account for the variability in demand for each piece of equipment, across all the facilities. One of the critical requirements in order to facilitate both lean and agile practices is the need of full network visibility. This can be achieved with the application of RFID, which is explored further in the next section.

Aggarwal and Lim (2011) (Appendix A-6) investigate the opportunities for LSPs to gain competitive advantage with RFID enabled RTE. This area of research is explored further by Aggarwal and Lim (2013b) (Appendix A-1) where competitive advantage theory is applied to LSPs and a framework to encourage cost and value advantage through four elements: asset reduction, increase of asset utilisation, RTE cycle-time reduction and utilisation of real-time data, is presented.

2.3.2.4 RFID enabled RTE

Aggarwal and Lim (2013b) (Appendix A-1) highlight that the reusable nature of RFID tags makes them an effective companion for RTE application (Swedberg, 2009b). As the tags are reused this increases the return on investment (ROI) of both the tags and the RTE tracking system. Furthermore, in comparison to the barcode advantages such as no line of sight and robustness, encourage applicability to RTE. The implementation cost of RFID technology is a significant barrier in the adoption of the technology for general supply chain application (Huber et al., 2007). However, this is not the case for RTE as a healthy payback period is achieved, making RFID enabled RTE a very cost effective tool within the supply chain (Hellstrom, 2009). Ikea and Arla Foods achieved payback periods for RFID attached to RTE in 24 and 14 months respectively. The advantages of RFID are crucial as it helps to reduce the problems associated with RTE.

A key attribute that RFID technology brings to the RTE network is visibility (Chuang and Shaw, 2007, Hellstrom and Saghir, 2007, Hellstrom, 2009, Ilic et al., 2009, Martínez-Sala et al., 2009). Through the enhancement of network visibility, the problems of RTE costs, shrinkage, ownership and cycle times as described in section 2.1.3 and the DHL Exel Supply Chain case study in section 2.1.4, are improved. Costs associated with RTE, such as tracking are decreased through the reduction of handling costs. RFID technology when applied to RTE removes manual handling and improves the efficiency of automatic identification as no line of sight is required. Another problem, shrinkage is reduced or even completely eliminated with RFID technology due to constant monitoring capability as stated in a study of Arla Foods (Hellstrom, 2009). Ownership problems in a LSP pool environment where the RTE are shared amongst clients is aided with the rewritable options of the RFID tags. Each tag attached to the RTE is updated every time there is a change or update. This increases the interchangeable rate of RTE between clients and the ownership of RTE is clarified throughout the supply chain. Finally, cycle times are reduced with increased visibility, reducing bottlenecks and increasing RTE velocity, which reduces RTE cycle costs by up to 22% and assist in the reduction of supply chain costs overall (Hellstrom, 2009, Ilic et al., 2009, Martínez-Sala et al., 2009).

The increase of benefits and extended capabilities of RFID enabled RTE has encouraged the development of a multi-faceted piece of equipment, where its functionality amplifies as it becomes a mutli-purpose advantage tool. Amongst others, a continually growing list, it acts as a security system for goods (Swedberg, 2011c); when attached with sensors becomes a product monitoring tool for chilled foods or pharmaceutical items (Martínez-Sala et al., 2009); a packaging device acting as a display unit (Martínez-Sala et al., 2009); automation tool for processes such as picking and receiving-in (Swedberg, 2009b, O'Connor, 2009); as part of a carbon footprint reduction system (Swedberg, 2010); and goods and equipment tracking device (Collins, 2006, Wasserman, 2010). This demonstrates that RFID application to RTE, forms a dual-purpose piece of equipment, where both the RTE and the goods are monitored.

The benefits of RFID application to RTE have increased its popularity for use of moving goods within different types of supply chains. There are a variety of companies using RFID enabled RTE with great interest in the automotive and retail sector. Also, LSP's interest in the potential of the technology continues to increase. In the automotive sector, companies such as Volkswagen and Mercedes-Benz have utilised the application of RFID to RTE in order to reduce the RTE losses, which cost the automotive sector $750 million dollars annually (Wasserman, 2010). It is highlighted that the application of RFID reduces the number of RTE needed in the system by 30%, which is significant as one piece of RTE costs up to $2000 dollars. Furthermore, it is highlighted that with RFID enabled RTE, particular processes such as receiving-in goods, have a labour reduction of up to 80%.

Retailing giants such as Wal-mart, Marks and Spencer and Tesco are trialling and implementing RFID enabled RTE to enhance their supply chains (Collins, 2005a, Collins, 2006). Other retailers such as Argentinean medicine wholesaler, Monroe Americana attached RFID tags to 25,000 containers in order to reduce losses and increase efficiency of processes such as unloading and loading (Swedberg, 2011a). A major retailer, Metro Group are collaborating with DHL to tag 1.3 million pallets per year to cater for delivery of goods to all 89 of their stores in France in order to increase the speed of delivery and obtain more accurate and reliable data (Wessel, 2008). The implementation of RFID enabled RTE in sectors described above, offer identification and network improvement, however LSPs are seeking further value-adding applications of RFID enabled RTE.

CHEP, a RTE rental company applies RFID to many of its RTE. In one example, RFID application to crates for use in the Australian automotive industry enables CHEP to keep track of which customers have which crates and allows the customers to monitor the flow of goods, which are carried by the crates (Bacheldor, 2006). Another company, Packaging Logistics Services (PLS),which supplies pallets and containers has utilised the tracking function of RFID enabled RTE and is seeking to apply the technology to 2.5 million RTE, which are then leased. The aim is to reduce the safety stock, whilst still maintaining RTE availability in the system (Swedberg, 2009b). Another LSP based in Toronto, JD Smith & Sons is trialling an RFID enabled pallet system in order to better track pallets, improve accuracy, reduce manual labour, reduce errors, offer tracking and tracing and remove manual errors from processes (Swedberg, 2010).

Although there are many advantages and opportunities for gaining network efficiency with RFID enabled RTE, there is a lack of framework within the literature of how RFID enabled RTE assists in gaining competitive advantage for the LSP and enhance their client's supply chain. As the use of RFID enabled RTE continues to grow in the LSP market, the next section explores competitive advantage and its application to the LSP sector.

2.3.2.5 LSP's RFID enabled RTE Competitive Advantage Model

After applying competitive advantage theory to LSPs, Aggarwal and Lim (2013b) presents a framework which suggests a focus on four areas to enhance competitiveness, which is illustrated in Figure 2.12. The competitive advantage model is discussed in detail by Aggarwal and Lim (2013b) in Appendix A-1 where it is emphasised that a LSP achieves cost advantage through the introduction of consolidation techniques and the reduction and further utilisation of assets. Simultaneously, value advantage is gained through the introduction of a cross-docking system in conjunction with introducing real-time data updates and the reduction of the RTE cycle time. There is an extended discussion in the paper of how RFID enabled RTE improves competitive advantage at each level of the model. Some benefits highlighted include the increase of throughout rates of products, the availability of unique track and trace facilities and the development of efficient maintenance systems in order to introduce a preventative strategy for RTE management.

After the research conducted by Aggarwal and Lim (2013b), it was questioned whether the function of RFID enabled RTE goes beyond a tracking tool, particularly when it is combined with decision-making capability. Therefore, the automation and possibilities of autonomy are explored.

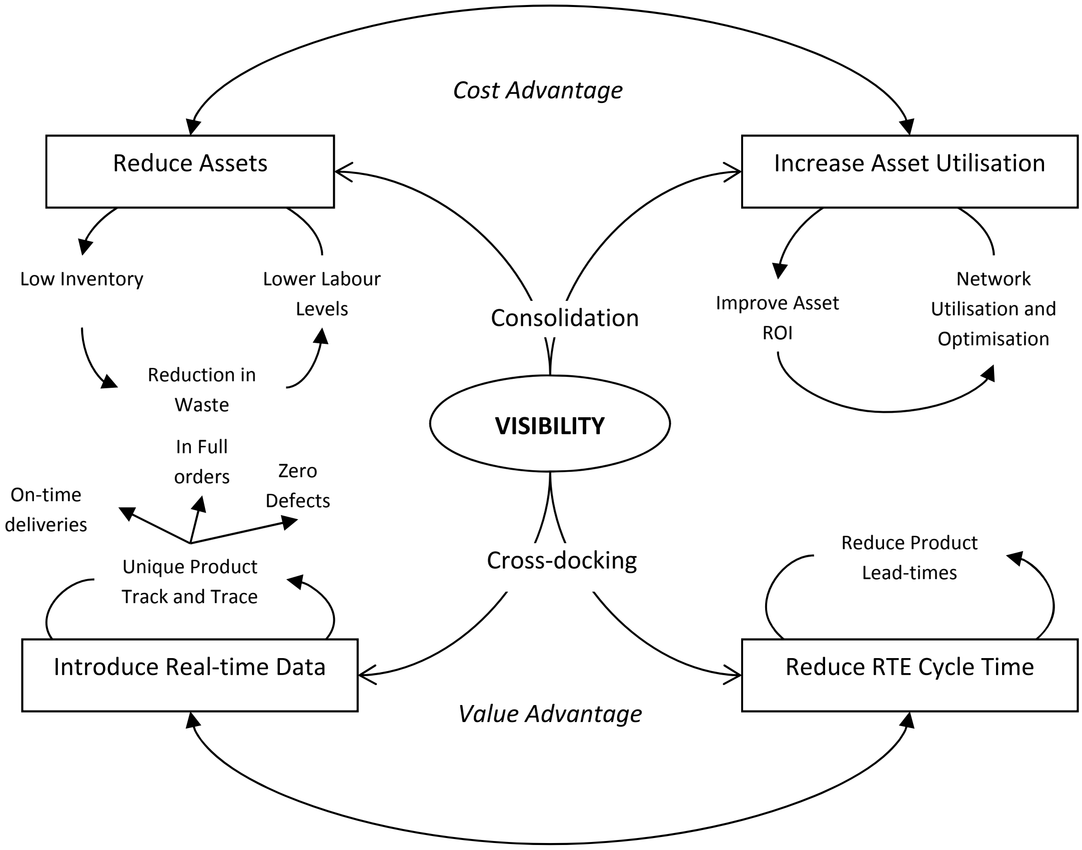


Figure 2.12 A RFID enabled RTE Competitive Advantage Model for LSPs

2.3.3 Automation

Today, the question of how to successfully reduce supply chain costs, whilst increasing customer satisfaction continues to be the focus for many firms. Aggarwal et al. (2011) (Appendix A-7) highlight that the introduction of automation may increase flexibility, whilst reducing inefficiencies, however despite the dynamic and process driven environment of logistics, there is the absence of a cohesive automation approach to guide companies in improving network competitiveness. Aggarwal et al. (2011) explore the potential of automation in the supply chain through RFID application addressing the gap in the literature by developing a three-level framework automation application approach with the assistance of RFID enabled RTE.

Aggarwal et al. (2011) observe that although the use of RFID is prevalent in many industries in conjunction with an array of different functions, one area of continued interest is the combination with automation (Swedberg, 2011b, Wasserman, 2007, McFarlane et al., 2003). Within the distribution network there is a move towards re-configuring processes with a heightened role for automation. One area where there is a lot of potential for automation integration is the RTE network. RTE is an important and essential part of the supply chain, however inaccurate data and poor management has led to the cause of network delays, excess inventory, asset damage and inefficient utilisation (France and partners, 2007, Meng et al., 2010). RFID makes assets more manageable and assists in advanced levels of network automation, improving competitiveness. Aggarwal et al. (2011) present a three-level framework to consider automation in relation to data retrieval, physical processes and of a cognitive nature as illustrated in Figure 2.13.

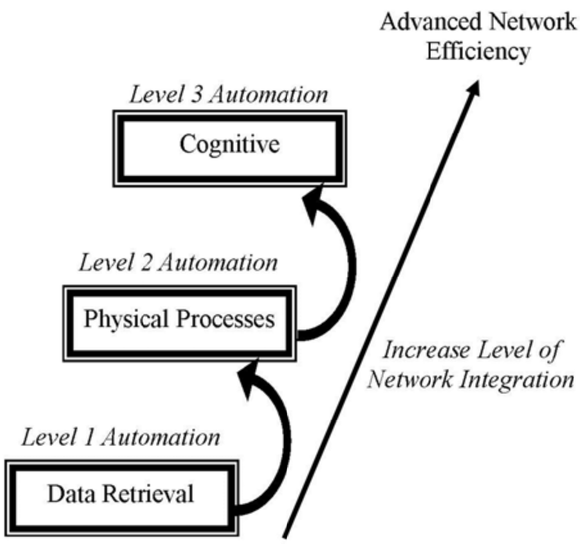


Figure 2.13 A Three-Level Framework of Automation Integration

2.3.3.1 Data Retrieval

The level of data automation from an RFID enabled RTE is higher than from a barcode as illustrated by Aggarwal et al. (2011) in Figure 2.14. The graph highlights the advanced level of data automation that RFID brings, with increased amounts of benefits in comparison to the barcode. Utilisation of RFID technology increases the level of data accuracy, enhances the implementation of automation systems and provides unique product identification (McFarlane et al., 2003). RFID technology enhances the capability potential of the RTE as it offers continuous packaging in the cycle acting as display units, and in addition makes storage and transport networks more efficient (Martínez-Sala et al., 2009). Also, RFID is embedded into the RTE with sensors, allowing the continuous monitoring for items that need to be kept within certain temperatures (Martínez-Sala et al., 2009).

When RFID is combined with GPS, higher levels of visibility on a global level are achieved. Introducing RFID triggered alarm management on an automated asset management system, creates the possibility to increase data quality in the system and the rate of work carried out on each process. The increased visibility leads to improvements in efficiency and reduction in labour costs (Meng et al., 2010, Hellstrom, 2009, McFarlane and Sheffi, 2003). RFID technology provides advanced levels of data retrieval automation, but further advances in efficiency require the re-configuration of physical process.

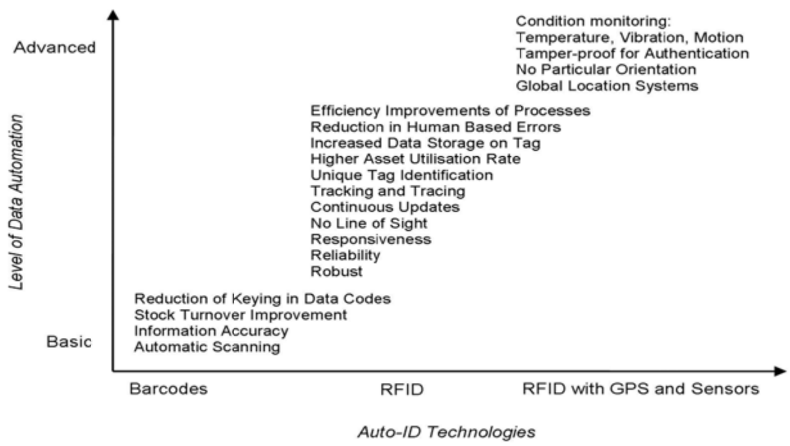


Figure 2.14 Benefits of Advanced Automation with Auto-ID Technologies

2.3.3.2 Physical Processes

On the second level of the framework, Aggarwal et al. (2011) consider the reconfiguration of physical processes throughout warehousing and transportation in regards to automation. Reconfiguring distribution processes brings significant savings, particularly in distribution facilities as labour is up to 80% of the total costs, and therefore there is capacity for cost reduction (Bose and Pal, 2005). Labour reduction is achieved through the automation of processes such as unloading, loading and picking. DHL use automation techniques to improve such processes (DHL, 2013). Robot arms are used to optimise parcel flow movements of unloading and loading. One facility that DHL opened was fully automated and designed to process 17,500 CDs or DVDs per hour where they are sorted, labelled, packed and made ready for shipping (DHL, 2013). In addition to labour reduction, other benefits include improvements in accuracy, reduction of inventory and increase in asset utilisation (Bose and Pal, 2005). The advantages of the implementation of robots, automated conveyors and RFID technology to RTE in order to increase process efficiencies are also discussed. As the labour is reduced with the introduction of RFID enabled robots, the balance between automation and labour is discussed with a review of a case study of KH Lloreda (Swedberg, 2011b).

2.3.3.3 Cognitive

Aggarwal et al. (2011) assert that the role of the human is crucial in decision-making. A cognitive automation approach refers to where the decision-making element (which is normally conducted by humans) is decentralised and moved to the objects within the network (Scholz-Reiter et al., 2004). Using software agents, objects are programmed with rules and objectives, where each object guides its own way through the system in real-time, utilising the data gathered via attached RFID tags. The real-time data communication capability of RFID technology has made the potential implementation of software agents more prolific within logistics (Scholz-Reiter et al., 2004). The level of autonomy of each object is further increased when combined with technology such as RFID, GPS and WLAN as this combination increases data accuracy on which decisions are made (Hülsmann et al., 2007). By decentralising the decision-making to an object level, systems are able to cope with unexpected changes better with the increase in flexibility, which is crucial in the distribution environment (Mes et al., 2007). One agent based system developed encouraged the optimisation of truck usage, whilst increasing customer service levels by allowing each vehicle the opportunity to swap jobs (Mes et al., 2007). One area where object-level autonomy needs to be further applied, is to the RTE network where it has to plan its own route and maintain optimal movement for itself and the goods it is carrying. The concept of autonomy in regards to RFID enabled RTE is explored further in the next section.

2.3.4 **Autonomy**

Aggarwal and Lim (2012b) explore the concept of autonomy in a paper titled "Improving Returnable Transport Equipment Management with an Autonomous Approach" in Appendix A-4. They investigate the nature of RTE network management, building on current research and practices and filling a gap in the literature through the investigation of a product-centric approach, where the paradigms of "intelligent products" and "autonomous objects" are explored. A paradigm shift of RTE management research is proposed where RTE is viewed as a network optimising tool rather than just a waste product. Advanced research development is encouraged in order to align academic research with problem areas in industry. The paper advocates the use of RFID technology and agent based software in order to develop an information-centric approach with efficient and effective decision-making.

2.3.4.1 Product Centric Approach

Aggarwal and Lim (2012b) observe that in a bid to improve flows, research in product-centric approaches in supply chain applications have been increasing (Rönkkö et al., 2007, Baïna et al., 2009, Bardaki et al., 2011). Baïna et al. (2009) use the approach to resolve product based data inaccuracies, which cripple the supply chain, and stress the importance of having full and complete information throughout the network in order to increase intelligence. Bardaki et al. (2011) emphasise the significance of RFID technology in object tracking as the information retrieved offers full visibility, initialising a product-centric approach. Although the amount of data related inefficiencies are amplified due to the increased quantity of data transactions, LSPs are encouraged to move away from the current location based model to a more product-centric approach increasing responsiveness and customisability (Rönkkö et al., 2007, Holmström et al., 2009). Aggarwal and Lim (2012b) insist that there are currently two prominent paradigms within the area of product-centric approaches to supply chain management improvements: intelligent products, which focus on the information side, and autonomous objects, which explore the decision-making elements. Both areas are explored further below.

2.3.4.2 Intelligent Products

Increasing the "intelligence" of items through a product-centric approach within supply chain management centres on product information management and tracking with full visibility (Holmström et al., 2009, Yang et al., 2009, Valckenaers et al., 2009, Kiritsis, 2011, Meyer et al., 2009, Kärkkäinen et al., 2003). Holmström et al. (2009) focus their attention on creating change by implementing systems based on continuous monitoring of products throughout the supply chain rather than being scanned at set locations. RFID provides the capability needed to retrieve and transfer the relevant information, whilst updating its own status. Kiritsis (2011) builds product based intelligence via four levels: combination with sensors, higher memory and provide advanced communication for integration with other technologies where the use of RFID is essential. Other literature within the paradigm of "intelligence" has taken the definition further to consider agent application (Meyer et al., 2009, Yang et al., 2009, Valckenaers et al., 2009). Valckenaers et al. (2009) recognise that in order to encompass the notion of intelligence, there needs to be a decision-making element. They state that an intelligent product needs to be combined with an intelligent agent. Yang et al. (2009) embody intelligent product theory by attaching "intelligent data units" to each product. The purpose of the unit is to provide lifecycle data where each unit is packed with sensors, memory and communication features. The information is then transferred to a software agent in order to process the relevant information for lifecycle management. The intelligence of the product is created through a data information retrieval function and a software agent for data analysis. Aggarwal and Lim (2012b) assert that a product-centric approach requires an information-centric environment, which is enabled by RFID, and intelligent agents, to represent each unit to foster decision-making functionality. The concept of decision-making is explored further in the literature under autonomous objects.

2.3.4.3 Autonomous Objects

Autonomous control is defined by Windt et al. (2008) as “the ability of logistic objects to process information, to render and to execute decision on their own”. The literature around intelligence focuses mainly on the collection, process and transferral of information with the notion of decision-making, however the thread of autonomy takes this further, although it is in early stages. This moves beyond “intelligent” products as they make their own decisions based on the relevant information collected from the environment and critically the learning functions, which are implemented. A key attribute of autonomous control is the decentralisation of decision-making to the products. Windt et al. (2008) explores decentralised decision-making, which improves efficiency of production logistics, enhancing responsiveness of machines and improving system flexibility. This highlights the potential of autonomous objects in the supply chain and the ability to enable them to act independently. Currently much of the research is conceptual and in early development. Further research is required into the potential within this area in preparation for applications of these concepts to real world cases. After reviewing the literature of autonomous and intelligent objects, the two key requirements that need to be added to the RTE network in order to facilitate network improvement are decision-making functionality and information-centric RTE, as illustrated in Figure 2.15. The relationship between the two areas is discussed further in the next section.

2.3.4.4 Network Optimising Tool

Aggarwal and Lim (2012b) assert that RTE plays a prominent and pivotal role within logistics networks, providing a constant contact point to the items being handled. To transform RTE into a network optimising tool, improving product and RTE flows, invoking the concepts of “intelligence” and “autonomy”, two approaches need to be considered simultaneously; information-centric RTE and software agents with a high level of decision-making capability as illustrated by in Figure 2.15. Both functions are intertwined, affecting each other’s ability and potential. The quality of decision-making is directly related to the information fed into the environment, therefore both are important in relation to real-time RTE management and network optimisation.

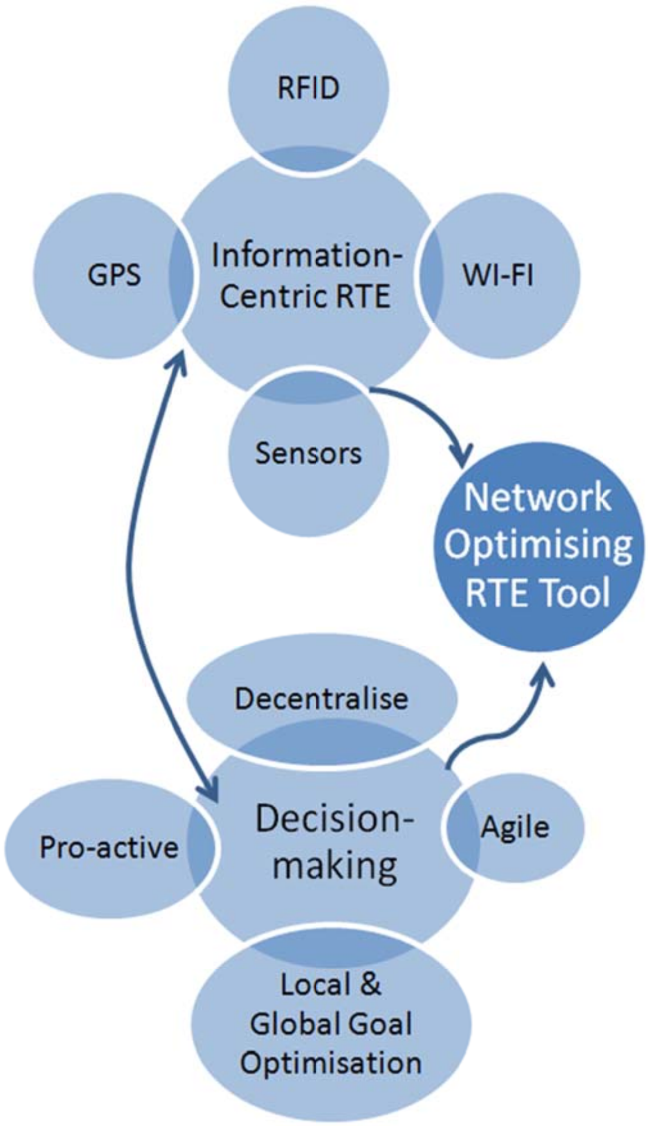


Figure 2.15 Diagram Illustrating the Essentials for an RTE Optimising Tool

The complexities of supply chains continue to increase challenging optimisation, particularly in a centralised decision-making format. The decision-making process tends to be centralised in order for management to remain in control and save costs. However, the dynamic, distributed and intricate nature of supply chains lends to a structure which favours a more decentralised approach enhancing agility. Agent based models are structured with the ability to cope with the dynamic and complex nature of supply chains (Ying and Dayong, 2005, Davidsson et al., 2005). Especially when software agents are combined with RFID technology, "intelligence" is increased and the data, information exchange and communication between the agents and environment, are subsequently all improved (Dias et al., 2009). Hence the combination of RFID and distributed agents is an effective way of initialising a product-centric approach (Rönkkö et al., 2007).

Aggarwal and Lim (2012b) assert that a RTE-centric approach to supply chain management means that each RTE has an effective stake in the optimality of the supply chain. Therefore, every decision carried out by each RTE would either hinder or improve the network. A positive outcome is reliant on the representation of each RTE and the capability of the decision support functions. These may be contradictory, and therefore agents need to be packed with negotiating ammunition in order to resolve any conflicts for RTE scheduling, which may arise. Information of the product and its environment is just as important as the decision-making support functions of the agents. The agents only utilise their decision-making capabilities based on the information fed into them. As Figure 2.15 illustrates, in order to achieve a network optimising RTE tool, where both RTE management problems are reduced and the wider network improved and optimised, data needs to be fed from information-centric RTE. This level of information is used during the decision-making process to introduce agility and create a pro-active network. Agents are built with learning functions to process the other agent's behaviour and the interactions with the environment (Panait and Luke, 2005). Logistically, activities such as scheduling are observed and decisions are made by other agents proactively. In addition, if there are unexpected changes, knowledge gained from the learning functions enable less damage to be caused to the outcome. Supply chains and RTE networks need to be agile in order to respond to unexpected changes. A multi-agent system offers a path of agility for supply chains through the high-level coordination, which is programmed for agent interactions (Lou et al., 2004). In addition, the combination of real-time data gathered from RFID enabled RTE with efficient decision-making of software agents on a RTE level enables increased network agility. The combination of information-centric RTE and intelligent agents programmed with rules and communication capability, instils network flexibility leading to the creation of autonomous RTE.

The agent methodology is investigated and evaluated in chapter 3 in relation to RTE application. However, before the introduction of the discussion of agent applicability, the nature of decentralised decision-making, which differs from the current centralised approach is discussed..

2.3.5 Decentralised Decision-Making

Aggarwal and Lim (2012a) assert that in the literature, supply chains have been modelled using centralised decision-making formats despite the networks being of a distributed nature (Lee and Whang, 1999). A centralised decision-making approach leads to decreases of efficiency and a lack of competitiveness, therefore a move towards a decentralised decision-making format is being pursued (Lu et al., 2012, Schneeweiss, 2003). Aggarwal and Lim (2012a) present a paper titled "Advances in Decision Support Systems on the Endeavour for Supply Chain Optimality" in Appendix A-3, which describes that nature of supply chains and their natural alignment with a decentralised decision-making format.

The decentralisation of decision-making in supply chains is an all-consuming area which spans economics, mathematics and computer science (Schneeweiss, 2003). The expansion of decentralised decision-making (DDM) in such vast areas has led to an array of different modelling methodologies and techniques in order to optimise the supply chain. The supply chain has many attributes as illustrated by Aggarwal and Lim (2012a) in Figure 2.16, which led to difficulties in modelling.

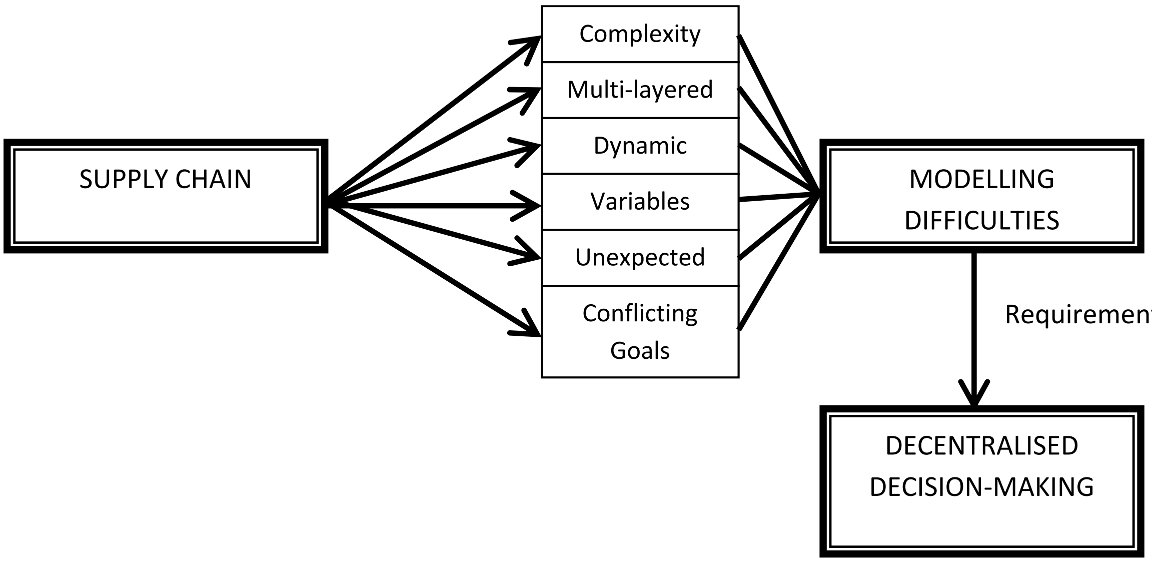


Figure 2.16 Modelling Difficulties within the Supply Chain

Throughout the literature there is an array of different ways described to model supply chains and engineer the networks to make the better decisions. However, the solutions are affected by some key supply chain attributes. The multi-layered and multi-dimensional nature of the supply chain coupled by many variables and unexpected events induces high levels of dynamism, which makes the level of realism in the model difficult (Sarimveis et al., 2008). As the supply chain complexity are not well captured, the decision-making implemented may often not produce better solutions (Sarimveis et al., 2008). The traditional ways of modelling supply chains such as “deterministic models where all parameters are known”, “stochastic models where at least one parameter is known but follows a probabilistic distribution”, “economic game-theoretic models” and models based on simulation, which evaluate the performance of various supply chain strategies, fail to adequately represent the dynamism of supply chains producing inadequate results such as late deliveries (Sarimveis et al., 2008). This has lead to a shift of modelling the dynamics of supply chains with “continuous-time differential equation models, discrete-time difference models, discrete event models and classical operational research methods” (Sarimveis et al., 2008). Although methods such as systems dynamics and discrete event simulation are popular, there are many assumptions and boundaries in place, for example fixed inventory and lead-times (Tako and Robinson, 2012). Therefore, recently there are researchers looking into applying techniques from artificial intelligence such as agent based modelling in order to represent the dynamism of logistics networks more appropriately. In addition, a technique like agent based modelling (ABM) is used in order to overcome difficulties, solve conflicts and conduct efficient trade-offs.

The six supply chain attributes described in Figure 2.16, make the implementation of effective decentralised decision-making (DDM) challenging. Aggarwal and Lim (2012a) identify six important elements to consider when implementing DDM as illustrated in Figure 2.17. To achieve effective DDM, the requirement is to meet global goals, whilst making decisions on a local level, which requires system coordination (Lu et al., 2012). The supply chain is large with a variety of areas spanning from purchasing to distribution whereby each area would have its own objectives, which could potentially conflict with each other, affecting global optimal goals. Therefore, the logic in the model needs to be developed so that the localised decision-making improves the global goals and not hinders them. This needs to be instilled through coordinating a decentralised supply chain, which is difficult to achieve (Schneeweiss, 2003). The coordination is affected by the complexities in the supply chain, the sheer number of decisions being conducted and different locations of decision-making. Therefore, strong connectivity through full visibility and efficient knowledge transfer systems are required. This will allow sound representation of the supply chain, which is not too abstract and theoretical of nature. The visibility, coordination and systems of knowledge transfer are also critical to ensure that decisions are carried out in a timely manner. Late decisions have a large negative impact on key performance indicators such as costs, on time deliveries, inventory levels and customer satisfaction levels.

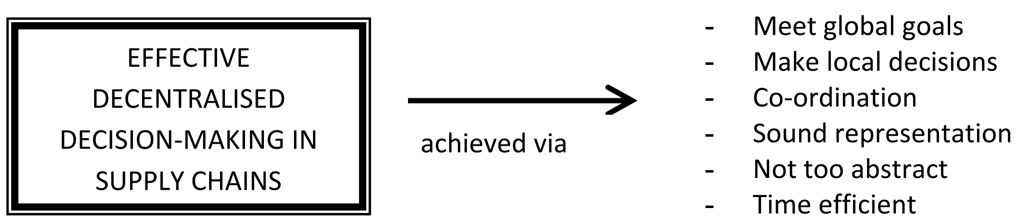


Figure 2.17 Effective Decentralised Decision-making

2.3.6 **Discussion**

RFID enabled RTE is a potential value adding tool in the supply chain (Aggarwal and Lim, 2010). The tool is utilised by LSPs to gain competitive advantage through cost reduction and service differentiation as asserted by Aggarwal and Lim (2013b). Although in the DHL Exel Supply Chain case study presented in section 2.1.4, RFID is applied to enable tracking of RTE, in the literature it is clear that functionality of RTE moves it from a tracking tool to a network optimising tool with the addition of automation and autonomy (Aggarwal et al., 2011, Aggarwal and Lim, 2012b). Any application of RFID enabled RTE requires a lean approach, which includes a focus on RTE reduction and utilisation, whilst the agile approach includes the utilisation of real-time data to improve the reaction to unexpected events, creating service differentiation. This strategy is further amplified with the formation of autonomous RTE units throughout the network where decisions are carried out on a decentralised level (Aggarwal and Lim, 2012a, Aggarwal and Lim, 2012b). This is pivotal on the endeavour to developing agile support systems, fully utilising an information-centric environment and encouraging RTE to be viewed as critical network optimising tools rather than costly waste.

2.4 **Conclusion**

RTE management is a challenging but important area to improve and is often outsourced to LSPs. The utilisation of RFID technology can enable the reduction of losses, improve process efficiency and offer effective tracking to increase RTE utilisation. In order for LSPs like DHL Exel Supply Chain to be competitive and retain clients, investigation of RFID application to RTE is essential.

RFID enhanced RTE becomes a multi-faceted piece of equipment as a: security tool; sensory product monitor; packaging device; display unit; automation tool; carbon footprint reduction tool and tracking device. However, the increased visibility and capability of RFID enabled RTE requires further investigation in regards to decision-making.

The distributed nature of the RTE network would favour a decentralised decision-making format, which requires an object level focus and an environment of real-time data in order to enable accurate decision-making. Agents are an effective way to represent objects from the bottom-up, enabling localised decision-making and sound representation of the network enabling the behaviour to be captured.

Chapter 3 explores the application of multi-agent systems to RTE networks and evaluates the problem and criteria, which need to be considered when applying this technology.

Chapter 3

Methodology

The problems associated with returnable transport equipment (RTE) management were presented in chapter 2, where the difficulties of representing and solving multi-objective supply chains were discussed. The need for a paradigm shift from conventional modelling techniques currently used to the implementation of distributed decision support systems was asserted. The literature review affirms that a decentralised approach to the RTE management problem domain encourages agility and fosters leaner networks, both of which are characteristics lacking in current RTE practices.

This chapter will investigate the potential application of multi-agent systems (MAS) to facilitate distributed decision-making in reference to the RTE network management problem domain. The field of MAS is an exciting and rapidly expanding area of research and development, as the characteristics of agents can offer a more realistic representation of physical entities in comparison to traditional modelling approaches such as discrete event simulation and systems dynamics (Borshchev and Filippov, 2004). Interest in the application of MAS to supply chains is increasing considerably and the decentralised nature of logistics networks presents an environment ripe for distributed artificial intelligence (DAI) application (Swaminathan et al., 1998, Fox et al., 2000, Aggarwal and Lim, 2013a). Aggarwal and Lim (2013a) (Appendix A-2) assert that RFID technology enhances MAS application to the RTE network by providing connectivity between the virtual and physical spheres. This improves RTE practices by increasing network agility, invigorating customisation of logistical services.

The objective of this chapter is to evaluate whether the design and deployment of MAS is congruous to the RTE management problem domain. To establish the objective, this chapter has been structured in the following way:

- The relationship between artificial intelligence (AI), distributed artificial intelligence (DAI), agents and MAS is discussed in section 3.1. An agent is defined by exploring its properties, architectures and behaviours, which is followed by an assertion of the importance of environment and society design when developing MAS.
- Section 3.2 explores the applicability of MAS to the RTE network problem by reviewing the literature for applications in logistics. The application of MAS for decentralised decision-making is explored by Aggarwal and Lim (2012a) (Appendix A-3). The application of RFID to enhance MAS is investigated by Aggarwal and Lim (2013a) (Appendix A-2). MAS application is then compared with other techniques in order to establish its merit followed by a discussion of the possible solutions to overcome the disadvantages of MAS application.
- Section 3.3 features a review of platforms that can be used in order to develop agents with a justification for the use of Eclipse and Java to build a bespoke agent package for this particular problem.
- Finally, section 3.4 concludes the chapter with a summary, highlighting the merit of the methodology.

3.1 General framework – Artificial Intelligence and Distributed Artificial Intelligence

The field of artificial intelligence (AI) has evolved over the decades; one definitive statement cannot capture the complex constituents of AI research. The term artificial intelligence which was coined by John McCarthy (McCarthy, 1979), was in reference to a workshop held in 1956 and as McCarthy states "he had to call it something" (Nilsson, 2012). The term did not originate with a concrete definition, however Luger (2005) expresses an apt and broad characterisation where AI is defined as "the branch of computer science that is concerned with the automation of intelligent behaviour" (Nilsson, 2012). Nwana and Ndumu (1999) assert the significance of the ground-breaking literature from the 1970s and 1980s as it is still drawn upon today for artificial intelligence theory. Figure 3.1, which illustrates the relationship between AI, distributed artificial intelligence (DAI), agents and multi-agent systems (MAS), begins with a description of McCarthy's (1979) original AI based characteristics, which include intentions, free will, understanding and creativity.

Figure 3.1 illustrates the founding AI principles as the backbone of various facets within AI, which include cognitive sciences, machine learning, robotics and logic. To assess which facets an agent should be programmed with requires a focus on the description of the subject or situation to be modelled and whether the intelligence is applied as a human-centric approach (agent thinking) or a rational approach (agent behaviour); both sides are illustrated in Figure 3.1 and are influenced by McCarthy's (1979) AI principles (Russell and Norvig, 2003). Modelling from a human-centric approach requires the object to be built so that it can think and act like humans (Russell and Norvig, 2003). This draws on the cognitive sciences whereby a model is developed and based purely on the human brain. On the opposite end of the spectrum, artificial intelligence (AI) can also be implemented from a standard where logic and rationality is modelled so that the agent behaviour can lead to the best outcome.

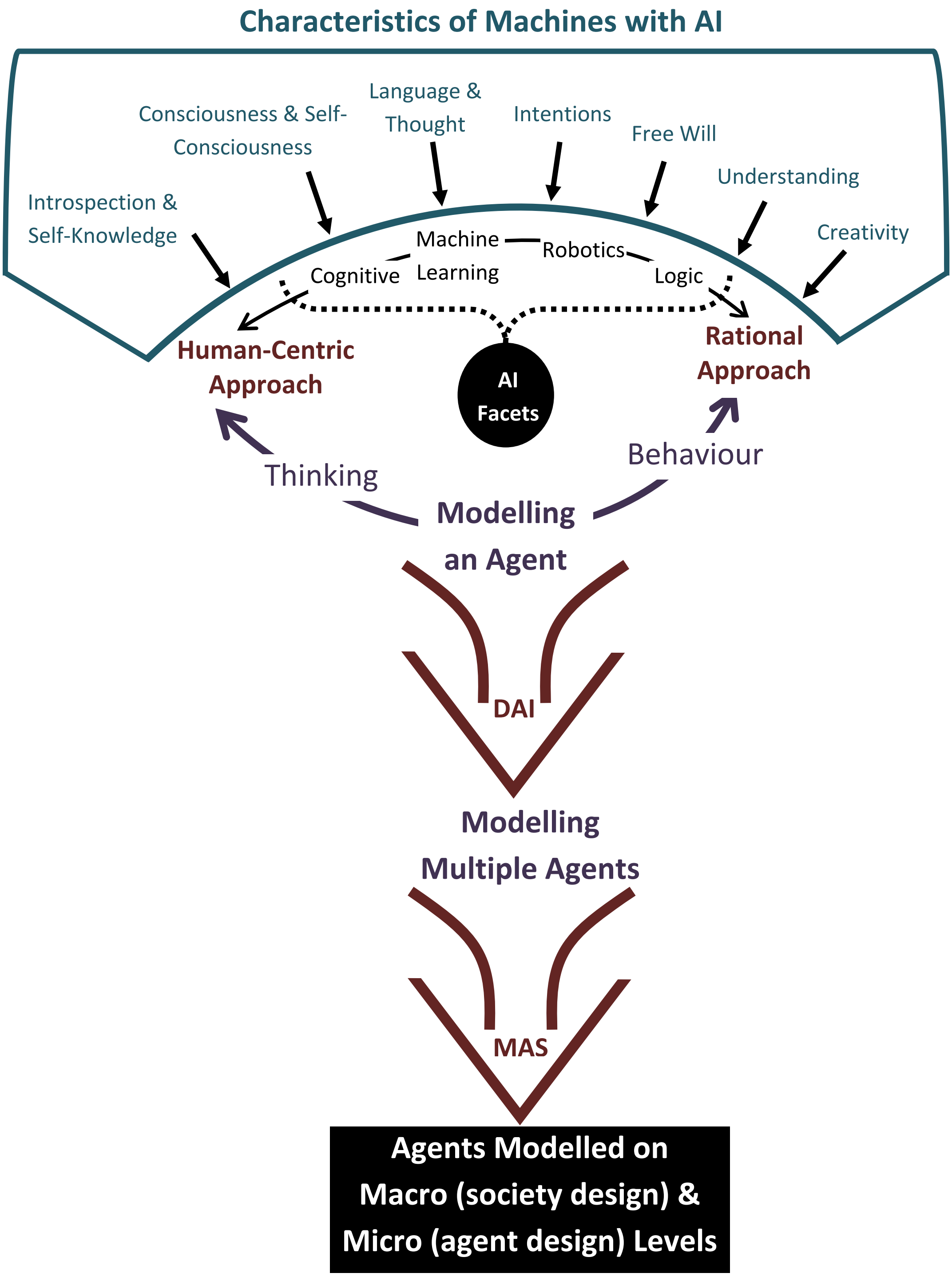


Figure 3.1 Illustration of the Relationship between AI, DAI, Agents and MAS

Traditional AI application encompasses a single entity (an agent), which makes decisions in enclosed environments. However, distributed artificial intelligence (DAI), a sub-field and extension of AI is where multiple intelligent agents with their own goals, interact with each other as well as the environment (Chaib-Draa et al., 1992). Therefore, as illustrated in Figure 3.1, DAI moves from modelling with one agent to multiple agents. The focus of programming moves from human reasoning to building a system of intelligent agents with rational behaviour (Russell and Norvig, 2003). The focus is on cooperation, competition and various other ways of increasing connectivity and interactions between agents, which forms the growing research area of multi-agent systems (Chaib-Draa et al., 1992).

Although MAS have threads of AI, Etzioni (1996) argues that only 1% of an agent is built using AI techniques, the rest will be based on computer science application and software engineering development. In fact, distributing AI by breaking down a larger problem, contributes to the understanding of AI and the cognitive sciences (Chaib-Draa et al., 1992). As Ferber (1999) asserts "in contrast to the classic conception of artificial intelligence, which postulates that the manifestation of intelligence rests on logical reasoning, multi-agent systems take as their point of departure not reasoning, but the behaviour of agents, the actions they perform in the world and the interactions which flow from them". As the problem is divided across multiple entities, the focus of MAS is on two levels, first macro-level, which concerns the design of the society and then on a micro-level, which is concerned with the agent design (Wooldridge, 2009). The society design and its interaction with the environment have been critical, and the areas which have been lacking focus in previous DAI structures (Michel et al., 2009). The ability to break down large complex and interactive systems with MAS lends itself well to decentralised decision-making (DDM) in supply chain applications (T Moyaux et al., 2006). Therefore, this methodology requires further exploration into its potential for solving associated problems, which have been raised in the literature review (chapter 2). The following section explores the definition of agents in more detail followed by the importance of their interactions in order to build coherent agent systems with adequate communication links.

3.1.1 **Agent Definition**

Agent is a widely used term with an expanding area of application research. Conceptually agents live in a virtual world and often in application they are representations of entities from the real-world, making a standardised, succinct and all-encompassing definition difficult, leading to a continued debate (Franklin and Graesser, 1997, Jennings, 2000). However, in order to avoid misuse and overuse, it is important to explore the essence of an agent and develop a core understanding (Wooldridge and Jennings, 1995).

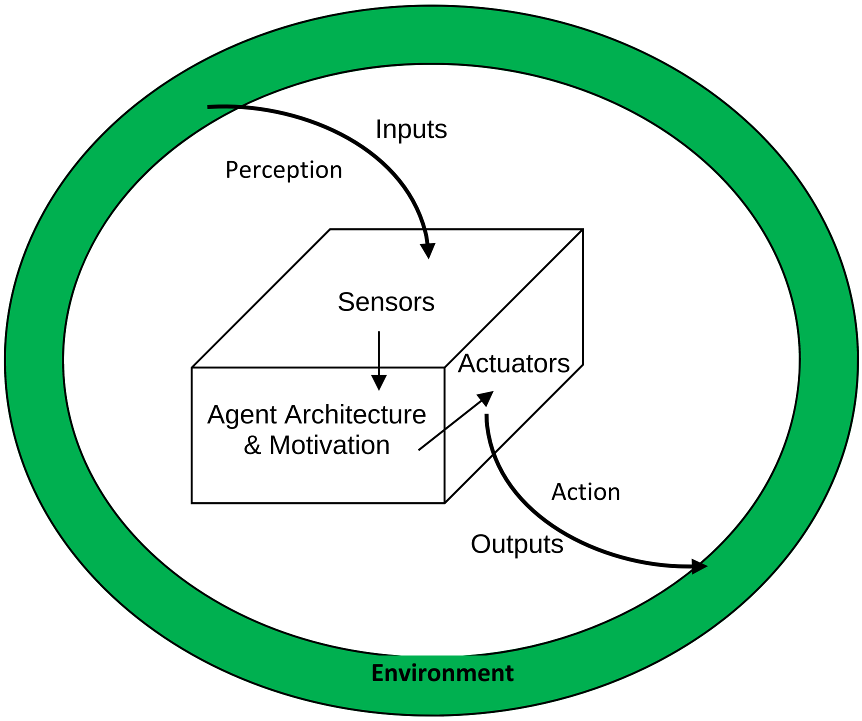


Figure 3.2 Illustration of Agent Functionality (Adapted from Russell and Norvig, 2003)

Shoham (1993) stresses that the outcome of an agent is dependent on what the programmer is aiming to achieve, however the literature refers to three main features that an agent must possess: an autonomous internal core, ability to perceive its environment and the ability to carry out actions. These three features are illustrated in Figure 3.2, where an agent is programmed with certain motivations and has a perception of the environment from where it receives inputs. It then uses its knowledge base, located within the agent architecture, and actuators to develop an action and feed this into the environment. The first of the three features, the autonomous internal core, should have an agent built with "beliefs, capabilities, choice and commitments", features which will tend towards instilling autonomy (Shoham, 1993). However, autonomy is challenging to achieve and is discussed further in section 3.1.1.1. Secondly, an agent must be able to perceive its environment so that it can then respond appropriately (Russell and Norvig, 2003). Finally, the agent must be able to carry out actions, these actions will then modify the environment, and therefore alter what it will "sense in the future" (Franklin and Graesser, 1997, Russell and Norvig, 2003). This feature distinguishes it from previous AI systems, which were developed with the purpose of instilling reasoning (Ferber, 1999).

Wooldridge (2009) summarises the perspectives above and describes an agent as a "computer system that is situated in some environment, and that is capable of autonomous action in this environment in order to meet its delegated objectives". The core features of the agent definition align with some of the original AI principles set by McCarthy as illustrated in Figure 3.1. Free will, which is a variant of autonomy, and also intentions, which is a feature of goal-orientation, is both key features of the agent definition. The difference is that the original AI characteristics have been modified in order to "reflect computing reality" of today (Jennings, 2000). A general overview of an agent has been provided, however selecting and building the "right" type of agent is very challenging and there are numerous ways to classify agents (Muller, 1999, Franklin and Graesser, 1997).

In this section, properties, architectures (internal structures) and behaviours (external structures) illustrated in *Figure 3.3* will be explored in order to increase understanding. The architecture refers to internal structures of the agents, which are programmed in order to influence the behaviour displayed by the agent, the external output. The behaviours and architectures selected affects each other, which in turn influences which properties are developed. The development of expertise in each of these areas determines the level of autonomy achieved within the agent.

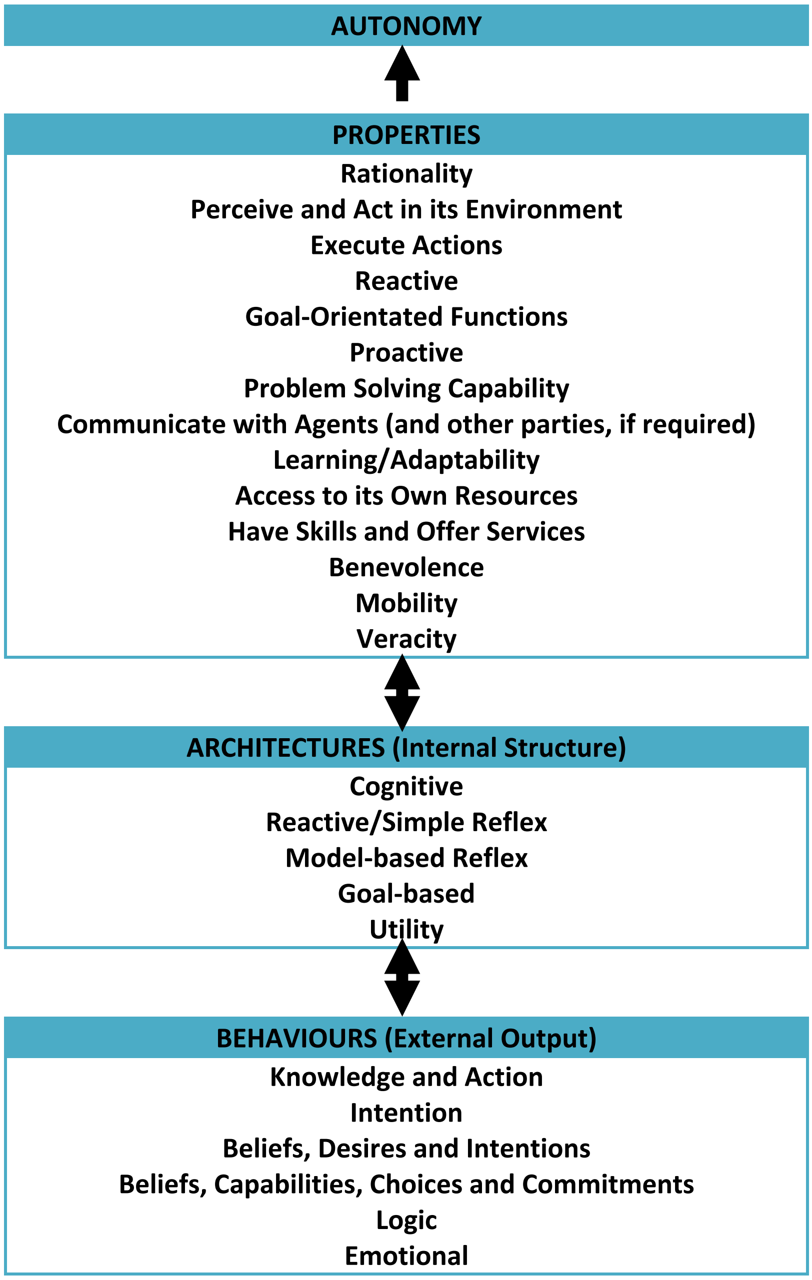


Figure 3.3 Agent Features and their Influence on Autonomy

3.1.1.1 Agent Properties

Autonomy is the most essential criterion where some authors stress that it is not even mentioned that the agent needs to be autonomous as it is assumed that agents have their own agenda and the ability to act in their favour (Russell and Norvig, 2003, Shoham, 1993, Franklin and Graesser, 1997).

AGENT PROPERTIES

Autonomous

Autonomy is the ability to be independent, make your own decisions and have control over your own actions. This means that "agents' activities do not require constant human guidance or intervention" (Shoham, 1993).

Rationality

A rational agent has the ability to pick the option which best serves it purpose to achieve its goal, maximising its result (Russell and Norvig, 2003).

Perceive and Act in its Environment

An agent can use sensors to pick up relevant information from the environment. The inputs can be used to evaluate the state of the environment and the agent can use the information for itself and then create an appropriate action (Ferber, 1999) (Jennings, 2000).

Execute Actions

An agent with the ability to execute actions, moves it beyond the traditional AI systems where reasoning was the focus (Ferber, 1999). The execution of an action will change the environment and then this will potentially impact any decisions which are made in the future.

Reactive

The reactivity of an agent is the ability to absorb inputs from the environment, construct an appropriate response and carry out the action, whilst meeting its objective in a timely manner (Wooldridge, 2009, Jennings, 2000, Franklin and Graesser, 1997).

Goal-Orientated Functions

An agent must have goals to achieve based on satisfaction or survival in order to possess behaviour. Therefore, the agent is influenced by its own behaviour, in addition to its environment making it proactive (Franklin and Graesser, 1997).

Proactive

A proactive agent takes "the initiative" in order to make decisions to achieve their goals (Jennings, 2000, Wooldridge, 2009).

Communicate with Agents (and other parties, if required)

Cooperation, negotiation and sharing of goals can assist an agent with achieving its goals (Wooldridge, 2009). Therefore, an agent must have the functionality to communicate with other agents, objects and players in the system (Ferber, 1999).

Learning/Adaptability

Learning is where the agent has the ability to change its behaviour based on its past experiences (Franklin and Graesser, 1997). It is possible to make an agent adaptable for the short-term through feedback loops, however it is a challenge to achieve long term adaptability (Muller, 1999).

Problem Solving Capability

Agents must have "well-defined boundaries and interfaces" in order to have the ability to begin flexible problem solving (Jennings, 2000). This requires an agent to have access to its own resources and a skills-set in order to evaluate offers.

Access to its Own Resources

An agent must have its own resources that it can use in negotiations in order to achieve its goals (Ferber, 1999).

Have Skills and Offer Services

An agent needs skills to be able to accept or reject offers and offer its services (Ferber, 1999).

Mobility

A mobile agent can move within its virtual environment and also move from machine to machine if required (Franklin and Graesser, 1997).

Veracity and Benevolence

An agent is believed to always do what it is required to do. It will stay true to its programmed objectives and always act in truthful way without communicating false information.

Figure 3.4 Descriptions of Agent Properties

Autonomy is the ability to be independent and make your own decisions whereby the "agents' activities do not require constant human guidance or intervention" (Shoham, 1993). Autonomy is an expansive ideology to instil into agents however; many of the properties discussed in this section encourage higher levels of autonomy, increasing the level of intelligence in the system. Learning is an example of another property, where an agent uses past events to influence future decisions, increasing the level of autonomy (Russell and Norvig, 2003). There are many other properties, which are described in further detail in Figure 3.4 (Ferber, 1999, Wooldridge, 2009, Wooldridge and Jennings, 1995). Agents will possess a range of properties as described in Figure 3.4 and these can then be used to help build different types of agent architectures. Depending on the architecture, some properties may be more important than others, and in turn the level of importance of each activity can determine which architecture is possible. Agent architectures are discussed in further detail.

3.1.1.2 Types of Architectures of an Agent

The architecture of an agent refers to its internal structure whereby there are different types of "design methodologies", which can enable the agent to achieve its goal(s) (Bryson, 2000, Ferber, 1999). Each type of architecture represents the programmers notion of how to construct intelligence into a system and the changing nature of a system over time can induce the architecture of an agent to "evolve" (Bryson, 2000). Each agent in a system may have different architectures and the variety of the different types used is proportional to the level of complexity of the system to design (T Moyaux et al., 2006). A system which requires multiple different agents may require the use of different types of architectures (Bryson, 2000). The most popular architectures as described in the literature are illustrated in Table 3.1 and explored further in Figure 3.5.

Throughout the literature there are core architectures, which are programmed into agents and these will be described in Figure 3.5. Although an agent may have certain architecture, this is determined by the programmer who initialises the internal structure and relationships of the agent. The behaviour of an agent, on the other hand is an external output, which when combined with the architecture, enables the agent to achieve its specification (Ferber, 1999).

Author and Reference	Type of Architecture
(Maes, 1994)	Interface
(Ferber, 1999)	Cognitive and Reactive
(Muller, 1999)	Reactive, Deliberative, interacting, Layered
(Russell and Norvig, 2003)	Simple Reflex, Model-based Reflex, Goal-based, Utility, Learning
(Wooldridge, 2009)	Purely Reactive, With State, With Utility Functions, Deductive Reasoning, Practical Reasoning, Reactive, Hybrid

Table 3.1 A Selection of Agent Architectures from the Literature

ARCHITECTURES (Internal Structures)

Cognitive

Cognitive agents have high levels of intelligence, which stems from their knowledge base (Ferber, 1999). The agents are self-sufficient and they utilise their knowledge in order to cooperate and coordinate, whilst on the endeavour to execute their plans to achieve their goals. However, constructing knowledge reasoning on a cognitive level is highly intensive and costly in terms of programming.

Reactive/Simple-reflex

Reactive/simple-reflex agents make decisions and carry out actions based on current available information, by using simple situation-action rules, and therefore do not need to be intelligent for the system overall to foster intelligence (Muller, 1999, Ferber, 1999, Russell and Norvig, 2003). These agents are not self-sufficient as they do not take into account the future when making a decision or have access to previous history, therefore access to the current state of the environment is crucial (Russell and Norvig, 2003).

Model-based Reflex

A model-based reflex agent identifies the current state of the environment and then keeps track via an internal model, which is then used by the agent to aid in decision-making (Russell and Norvig, 2003).

Goal-based

An agent which is goal-based considers the future when making a decision and permeates through multiple what if scenarios, picking the most appropriate option in order to achieve its goal (Russell and Norvig, 2003).

Utility

An agent with utility can quantify a state, which it can then use to maximise or minimise its options in order to better achieve its objectives (Russell and Norvig, 2003).

Figure 3.5 Descriptions of Agent Architectures

3.1.1.3 Agent Behaviour

There are a variety of internal components, which influence the behaviour of an agent and are developed over the life of an agent (Wooldridge and Jennings, 1995). Various agents' behaviours as described in the literature are listed in Table 3.2.

Author and Reference	Agent Behaviour (External Output)
Moore (1984)	Knowledge and Action
Cohen and Levesque (1990)	Intention
Shoham (1993)	Beliefs, Capabilities, Choices and Commitments
Bates (1994)	Emotional
Rao and Georgeff (1995)	Beliefs, Desires and Intentions (BDI)
Wooldridge (2009)	Logic

Table 3.2 A Selection of Agent Behaviours from the Literature

The literature supported by Moore's (1984) thesis on knowledge and action and Cohen and Levesque's (1990) work on intention, moves towards the most common form of behaviour for modelling agents, which is beliefs, desires and intentions (BDI) (Rao and Georgeff, 1995, D'Inverno et al., 2004). Many applications of behaviour asserted after the introduction of BDI are stemmed from this particular behaviour such as Shoham's (1993) extension. BDI behaviour is based on "mentalistic notions" where belief refers to the knowledge an agent has about the environment (Shoham, 1993). Desires are where the agent has preferred states within the environment and commits resources to achieve all the tasks or as many as those which are assigned to it. Intention refers to the desires, which the agent has selected to get and will achieve until completion or it is no longer possible (the tasks the agent is trying to achieve). When agents are developed these types of behavioural nuances can enable an agent to achieve autonomy. Works by Bates (1994) and Wooldridge (2009) attempts to introduce different aspects of behaviour, which an agent may display more prominently such as emotion or logic, however the BDI often remains as the base of an agent's behaviour.

This section has asserted that an agent is intelligent if it can interact, adapt and act within its environment, whilst attempting to achieve a goal. The intelligence of an agent is now stemmed in more practical applications, such as pattern recognition, learning, deduction, optimisation, environment interaction etc., which is a development from the more human based characteristics, such as understanding and consciousness as illustrated in Figure 3.1 (McCarthy, 1979). However, other characteristics such as free will (autonomy) dominates agent property. Autonomy influences the internal structures and external outputs of the agents; however agents can also be influenced by the environment in which they are embedded. This becomes even more important when the environment contains different types of agents. This creates a macro-level approach of thinking as the design of the society becomes critical. For distributed problems, such as supply chain management, it is clear that multiple agents are needed in order to represent the "multiple parts of control, perspectives and competing interests" (Jennings, 2000). This is now explored further under the umbrella of MAS.

3.1.2 **MAS**

A MAS is constructed with multiple agents, which implements a DAI approach of modelling and simulation. MAS is one of the rare methods that can be used solely to build a complex system (Weiming, 2002). Recently, MAS have received increasing interest as it can portray and simulate multi-dimensional systems with high levels of complexity better that more traditional approaches such as discrete-event simulation and systems dynamics (Heath et al., 2009). One of the key differences of MAS is that outcomes of the whole network cannot be determined as the individual components within the network are defined and so a "bottom-up" approach is conducted (Nikolai and Madey, 2009). Therefore, agents can be used to model supply chains with a distributed approach, which aligns with their business structure and geographical positions, whereby there are a multitude of actors with different requirements. As more agents are developed in the system this impacts the agents and the environment with which they are embedded, if the society is designed appropriately.

Figure 3.6 illustrates the introduction of another agent into the environment where the interactions and communications increase, and therefore these features are important to consider when developing agents. The society design (interactions and communications) will be discussed in section 3.1.2.2, first the environment of the agent is explored further.

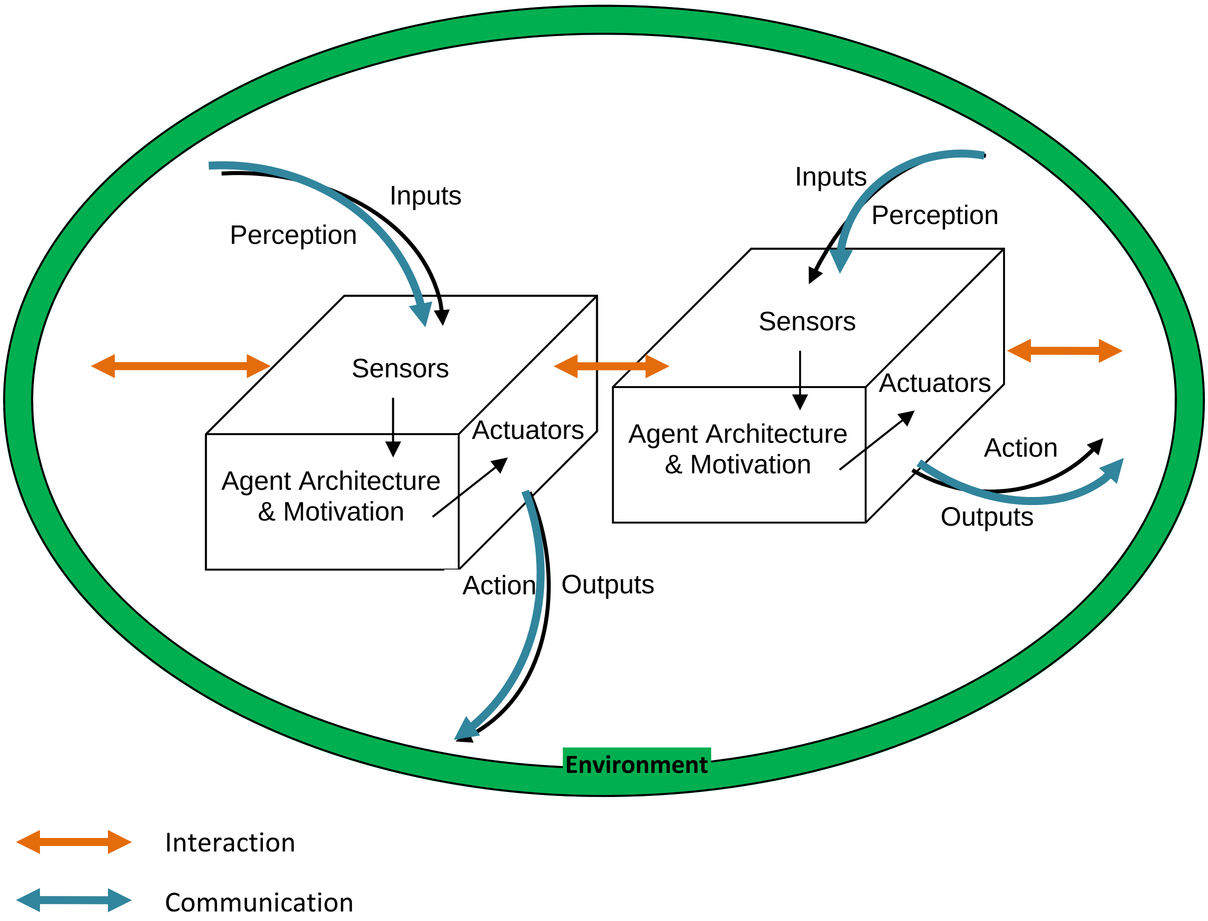


Figure 3.6 Illustration of the Functionality of a Multi-agent System

3.1.2.1 Environment

Every agent must be embedded in an environment, which is "everything outside the individual agent's boundary" (Parunak et al., 2004). The environment can include other agents, human stakeholders, system designers or conventional computer systems. Agent's decisions can change the environment over time, which makes it a "dynamical system" (Franklin and Graesser, 1997). Figure 3.6 illustrates two agents. An agent cannot predict the other agent's decisions. Therefore, it cannot know the state of the environment before the other agent has made its decisions. Therefore, the agent is influenced by the other agent due to its potential contribution to the environment. As the quantity of the type of agents embedded into the system increases, a higher level of unpredictable changeability occurs in the environment. The dynamical nature of an environment makes it difficult to categorise but there are key criteria, which are listed in Table 3.3 (Russell and Norvig, 2003).

ENVIRONMENT VARIABLES		
Fully Observable	Vs	Partially Observable
Deterministic	Vs	Stochastic
Episodic	Vs	Sequential
Static	Vs	Dynamic
Discrete	Vs	Continuous
Single Agent	Vs	Multi-Agent

Table 3.3 A List of Different Criteria for an Agent-based Environment (Adapted from Russell and Norvig (2003))

An environment is considered fully observable if the agent can perceive the state of the whole environment otherwise it will only have partial visibility. If an environment is solely determined by the current state and the actions of the agents then it is deterministic. On the other hand, a stochastic environment is one where the random variables alter the state of the environment. An episode refers to how an agent perceives and acts. When the perceptions and actuators are defined and set then this is a discrete environment. There is a big difference between a single agent and multi-agent environment. Agents are affected by the presence of other agents either directly or by changes to the environment, increasing the interactions within the system through communication as illustrated in Figure 3.6. Therefore, the interactions and communications need to be explored and defined when constructing multiple agents. This is discussed further in section 3.1.2.2.

3.1.2.2 Society Design

An agent acts on behalf of an entity and as a part of a society where high-level interactions and organisations of relationships are critical. An agent is trying to achieve its own goals and when it is part of a society, for example a supply chain, interaction with the environment and other agents can assist in goal achievement (Jennings, 2000).

Figure 3.6 illustrates the interactions between the agents and their environment. The orange arrows indicate two levels of interactions: between the agents and between the environment. The blue arrows indicate the four aspects of communication, each of which involves the environment and enables the two levels of interaction to occur.

3.1.2.3 Interactions

Studies assert that if animals and humans do not interact with each other then they are not able to reach their full potential and remain in a "primitive state" (Ferber, 1999). This observation influences the development of MAS whereby interactions are essential in order to represent macro-level behaviour of a system (Jennings, 2000). Interactions can either be conducted from each agent's knowledge level (agents interact based on their goals) or with flexibility where unpredicted interactions occur during the run-time (Jennings, 2000).

Interactions can improve current models based on algorithms as the output from an algorithm is dependent on the inputs, however with an interactive system it is possible to have components that use historic events to learn and interact, improving their adaptive nature (Wegner, 1997). An object is classed as intelligent not because of its algorithm but the way the object interacts with the environment. Algorithms perform tasks based on the rules programmed and whereas agent based modelling takes into consideration the interactions between objects, which improves the results (Wegner, 1997). Interaction is the key to scaling up larger problems rather that developing more complex algorithms. Interaction aids in the delegation of control through enhanced coordination between the objects. Table 3.4 describes various interaction situations which agents may conduct (Ferber, 1999).

Type of interaction situation	Goals	Resources	Skills	Category
Independence: Agents work independently without any interactions with other agents.	Compatible	Sufficient	Sufficient	Indifference
Simple Collaboration: The interaction is expressed “in the form of the allocation of tasks and the sharing of knowledge”.	Compatible	Sufficient	Insufficient	
Obstruction: The agents do not need each other to accomplish their goals but they “get in each other’s ways”	Compatible	Insufficient	Sufficient	Cooperation
Coordinated Collaboration: Agents “have to work together in order to pool their skills and take advantage”.	Compatible	Insufficient	Insufficient	
Pure Individual Competition: Agents “have conflicting goals and so they have to barter and negotiate in order to come to a resolution”.	Incompatible	Sufficient	Sufficient	
Pure Collective Competition: “If an agent is lacking skills then they form a group in order to coordinate with people who can help them to achieve their goals.”	Incompatible	Sufficient	Insufficient	Antagonism
Individual Conflict Over Resources: “If resources cannot not be shared, the agent tries to acquire the resources for itself.”	Incompatible	Insufficient	Sufficient	
Collective Conflict Over Resources: “This is when coalitions barter with each other in order to take full control”.	Incompatible	Insufficient	Insufficient	

Table 3.4 Describing Different Interaction Situations (Adapted from Ferber (1999, pp.66))

Table 3.4 describes eight possible situations of interaction, which may occur between agents. Half of the interactions favour agents with compatible goals, interactions such as “coordinated collaboration”. On the other hand, if there are conflicting goals between agents then a “pure individual competition type” of interaction would suit. This may be the case when building supply chain agents, for example a truck agent, which wants to achieve a full truck load versus a customer order agent, which needs to get an item delivered as soon as possible. The resources of an agent will determine if it has its own ability to achieve its goals and how much help is required from other agents. If it requires resources to complete its goals then it will use the skills of bartering and coalition forming. If an agent has insufficient skills, then just like resources it will be required to join forces with another agent with the skills to enable the completion of its goals. Finally, the agents may be indifferent to each other, have high levels of cooperation or be antagonistic, whichever way they all require a form of interaction.

Building agents with complex interactions is challenging as it is difficult to model and simulate real-world interactions and the input data can be inaccurate and inadequate, however this is crucial in order to develop agents with autonomy and represent real-world problems (Muller, 1999). This becomes even more prominent when trying to solve distributed problems in areas such as the supply chain. For example, with problems such as resource allocation, routing and scheduling, where the model is often a far abstraction from the real-world situation (Muller, 1999). The physical and virtual world can be aligned using RFID technology and this is explored further in section 3.2.1.1. In order to facilitate actions and create interactive behaviour, robust communication techniques are necessary.

3.1.2.3.1 Communication

Communication is essential to enable interactions and the type of communication will differ based on the interaction required (Ferber, 1999). A cognitive system would rely on messages where as a reactive system can use signals as the base for communication. As asserted in the previous section, interactions are the backbone of MAS and communication enables an agent to move from being a solo entity and into a part of a coordinated, connected and cooperative society. The main types of communication are either by message or through signals (Ferber, 1999).

Communication by message

A message is sent to an agent directly and the logic is stemmed from speech act theory (Austin, 1962, Searle, 1969). There are several types of different speech acts where it is stated that "communication is action" and the actions are carried out intentionally (Wooldridge, 2009, Ferber, 1999). Based on this theory a US project called Defense Advanced Research Projects Agency (DARPA) developed a high level communication standard to allow cognitive agents to cooperate (Ferber, 1999). They developed two languages: Knowledge Query and Manipulation Language (KQML) and Knowledge Interchange Format (KIF). KQML is where the agent states the purpose of the message but not the content. KIF represents the knowledge of the speech. However, there are certain draw-backs with these languages such as "ambiguity, imprecision, useless and incoherent performatives and missing performatives" (Ferber, 1999). A non-profit organisation, Foundation for Intelligent Physical Agents (FIPA) developed the Agent Communication Language (ACL) with a more rigorous semantic formula in order to try and offer a standard.

Communication by signals

Signals are a primitive type of communication as they are conducted through marks or traces, which are left behind often inspired by biology, for example animals leaving pheromones: as an agent comes across the mark they can then interpret it (Ferber, 1999). Signals are laid down based on their own principles, and therefore the agent must have the capability to sense the signals, and then also interpret them converting that signal to an action. A signal can act as either a stimulus, which produces a certain agent behaviour or as a signal of "meaning" as it travels straight to the agent's cognitive core (Ferber, 1999). Communication via a signal is different to communication via message, as it is incidental and often increases and moves in a natural way in the environment unlike a message, which is direct.

3.1.3 **Discussion**

Agents offer a different approach to modelling systems where the key is to instil autonomy and increase intelligence. The intelligence of a human is much about social environment and reacting with other humans as it is about genetics, and this philosophy is crucial in agent application. The knowledge base possessed by each agent does not equal to the sum total of the knowledge in the whole system, as the agents acquire and share knowledge in order to collaborate in common tasks, and therefore adding further intelligence to the system. Interaction ensures that the complexities of systems are captured from the "bottom-up" with changes and inputs occurring, whilst algorithms are running, leading to a paradigm shift in software engineering "from logic to interactive (agent based) models" (Wegner, 1997). Therefore, MAS highly increase the intelligence of a system overall. The autonomous property of an agent allows it to perceive and monitor the environment and negotiate for services to solve complex problems without intervention by other parties. However, the utilisation of agents must be in the right context.

MAS are currently being applied to problems which are not appropriate and this is leading to agents being used to formulate mathematical formulas where it is not advantageous (Nwana and Ndumu, 1999). The representation of agents should enhance the understanding of an entity, for example if an agent represents a machine it can be used to understand the structure, its past and future behaviour or how to repair or improve it (McCarthy, 1979, Shoham, 1993). In addition, the application of MAS should add further value to the system representation in comparison to other methods currently being utilised so a "requirements-driven" philosophy as opposed to a technology-driven one is essential (Muller, 1999, Nwana and Ndumu, 1999).

The thesis explores the improvement of RTE management with MAS and RFID technology. The most apt and challenging case for MAS application is a partially observable, stochastic, sequential, dynamic, continuous and multi-agent" environment, which are all the qualities a RTE network possesses (Russell and Norvig, 2003). The motivation and viability of MAS with RFID technology to the RTE management problem is evaluated further in section 3.2.

3.2 Applicability of MAS in Logistics

Early literature on agent-based modelling (ABM) in supply chains (SCs) comes from pivotal analysis by Swaminathan et al. (1998) and Fox et al. (2000), where the natural synergy between the application of agent technology to supply chain representation, and in particular their complexities enabling customisable, interactive and agile decision support functions are asserted. The characteristics of agents enable them to go beyond traditional modelling approaches such as discrete event simulation and systems dynamics as more realistic representations of physical entities can be gained (Borshchev and Filippov, 2004). ABM is particularly useful in the SC as agents can better represent active objects with individual behaviour patterns, which is more difficult to achieve with other modelling techniques on an item level scale. Once behaviours are captured of different entities and parts of the SC, local goals can be programmed and agents left to act in their own best interests. These agents can also feed into an overall system ensuring that certain limits such as time frames and quantities are not exceeded and global company goals and objectives are achieved for example, lead-time reduction of items.

ABM has been used to facilitate information retrieval and transfer, which can enhance SC agility. Moyaux et al. (2007) deploy agents to represent parts of the SC in order to share information to simulate the bullwhip effect. With information transferral through agents they were able to demonstrate a reduction of the bullwhip effect improving supply chain performance. For further discussion into the bullwhip effect the please refer to (Aggarwal and Lim, 2009) in Appendix A-9. García-Flores (2000) creates a network of agents representing different parts of the SC to facilitate a flow of information, enabling the integration of different business functions and improving relationships through building a cohesive SC. Jung et al. (2008) develop an agent based decision making tool to enable manufacturers and logistics service providers (LSPs) to construct cohesive production and delivery plans, whilst maintaining information confidentiality. This is a key characteristic of ABM, which encourages supply chain application as there are many parties involved who want data protection, and therefore information sharing can be difficult. Using agents alleviates this issue.

There are many more studies which take advantage of the potential decision-making capabilities of agents in various SC situations. Wang et al. (2009) use the distributed nature of agents to align downstream requirements with upstream resources, forming agile supply chains. Monteiro et al. (2007) improve SC performance through the formation of planner and negotiator agents making decisions on a local level, whilst mediator agents tend to global solution construction. Forget et al.(2009) overcome difficulties in centralised decision-making by decentralising the function to an array of agents capable of different behavioural patterns of planning, depending on their environment, leading to overall efficiency in the supply chain. Labarthe et al.(2007) propose an agent based system to overcome the complexities of providing mass customisation solutions by solidifying connections between different parts of the SC and being responsive to customer demands. Connecting agents in real-time with the physical entities can be an effective way of conducting and improving SC practices. The role of ABM in the SC application, and particularly in encouraging SC agility and decentralised decision-making (DDM) is explored further.

3.2.1 Enhancing Supply Chain Agility and Improving Decentralised Decision-making

Aggarwal and Lim (2012a) (Appendix A-3) investigate the reasons for the shift to DDM and evaluate the methods of ABM and hybrid modelling in terms of potentially achieving better solutions, in the supply chain. The accuracy of data is crucial, and therefore (Aggarwal and Lim, 2013b) (Appendix A-1) explore the use of RFID and MAS application to improve RTE practices by increasing network agility in order to invigorate customisation of logistical services. They assert that RFID technology enhances MAS application to the RTE network by providing connectivity between the virtual and physical spheres.

3.2.1.1 Improving Decentralised Decision Making with MAS

SCs are an evolving entity embedded with high levels of complexity. The multi-layered and global networks, which are scattered with various actors, creates autonomous functioning units throughout. These units often have differing goals to each other and conflicts can result in trade-offs, where the logic of each decision-making unit is critical. Effective decision-making is essential in order to achieve efficient and optimal SCs, however modelling these networks is extremely challenging due to SC attributes such as complexity, dynamism and layered structures (Wernz and Deshmukh, 2010). Difficulties of modelling the SC are further heightened with the constant stream of uncertainties and unexpected events, which can occur (Ko et al., 2006).

In the literature, SCs have been modelled using centralised decision-making formats despite the networks being of a distributed nature (Lee and Whang, 1999). A centralised decision-making approach leads to decreases of efficiency and a lack of competitiveness, therefore a move towards a DDM format is being pursued (Chow et al., 2007, Lu et al., 2012). The decentralisation of decision-making in SCs is an all-consuming area which spans economics, mathematics and computer science (Schneeweiss, 2003). The expansion of DDM in such vast areas has led to an array of different modelling methodologies and techniques in order to optimise the SC. Prior to further investigation of these options, the nature of supply chains is discussed and the requirements for effective DDM explored.

There are three main aspects to consider in the application of ABM; global optimisation, environment representation and the mechanism (for further discussion see Appendix A-3). Firstly, the balance between conducting local decisions and global goal optimisation; agents can achieve their own local goals and then achieve the global goals through targets (Wernz and Deshmukh, 2010). A structure is developed where local agents make decisions on a local level, whilst considering the impact on overall global supply chain objectives. The quality of global and local optimisation is heavily affected by the environment of the agents and the mechanism developed.

Secondly, the environment created in the virtual world has to represent the dynamic nature of SCs in the real-world, and also any unexpected events, which may occur throughout the network such as traffic, cancelled orders, shrinkage, human error, addition of resources and loss of fleet. There needs to be an efficient way to introduce connectivity between the real-world and the virtual world. Visibility can be created to capture data of stochastic events via technologies such as radio-frequency identification (RFID), GPS and GSM. RFID technology can provide a continuous system of data, enabling full visibility and constant track and trace. The use of these technologies can be time efficient, encouraging higher throughput and productivity. In addition, access to real-time information can lead to higher levels of efficiency with scheduling and customisation. The technology can be used to monitor the environment whereby the information can be fed into the agents. Once the data is in motion then it needs to be monitored and analysed so that the agents become an interactive tool with links between the real, digital and virtual worlds.

Thirdly, agent based systems need rich knowledge, coordination and powerful reasoning mechanisms to aid with the decision-making. Knowledge is a critical requirement for agents in order to make their own decisions. As the knowledge becomes more sophisticated, the agent can use the information to make enhanced decisions enabling better outcomes. Coordination is critical in order to foster agents with the ability to acquire, process and send back information. The advanced levels of coordination will determine the amount of knowledge, which can be used and the quality of the DDM mechanisms. The coordination needs to be created on a communication level, and also enable the seamless integration of a variety of logic, algorithms and heuristics, which becomes more challenging when striving for a proactive approach (Schneeweiss, 2003). Advanced levels of coordination will increase the flexibility, which can be represented and align the model with the real-world.

Negotiation is essential to enable agents to make their own decisions and this is also another level of coordination, which needs to be implemented. On each local level there can be different objectives which may be of conflict, for example the requirement to use one resource for multiple tasks at the same time. Therefore the agents need to have the ability to form a negotiation and barter. In the literature, the creation of flexibility and responsiveness by some researchers has been through implementing bidding and market-based approaches (Weiming, 2002). However, this area can be further improved with the addition of learning functions.

The main shortcoming of ABM in SC application is simplification, which plagues all modelling techniques and this in part is due to the feasibility of developing the mechanism as discussed above. Firstly, as with all modelling techniques, the problem is broken down into smaller ones and then the solutions coordinated, which leads to simplification of the real-world problems (Schneeweiss, 2003). In addition, high levels of complexity can be difficult to represent through modelling but it is less effected through ABM than applying OR techniques or mathematics. In addition, to encourage representation of increased layers of complexity, there must be the transfer of information, closing the gap between the virtual and real-world through advance communication devices such as RFID. Each part of the mechanism requires much further research in application in the SC. In addition to this under-developed area, feasibility of using ABM needs to be explored. The costs associated with programming the code and computing the algorithms needs to be calculated especially as bespoke systems are often used as different supply chains have different goals and purposes (Lu et al., 2012). The need to use bespoke systems is reflected in the limited number of domains and toolkits available in ABM, which can produce poor results when applied in a generic fashion (Nikolai and Madey, 2009). In addition to costs, the time span to produce solutions in real-time is of paramount importance. The application of ABM in supply chains is still developing and more research is required to making improvements and achieve the potential of ABM (Heath et al., 2009). To overcome the shortcomings in the different modelling approaches, hybrid modelling is explored (Aggarwal and Lim, 2012a) (Appendix A-3).

Hybrid modelling in SCs takes advantage of different techniques such as ABM, operational research, mathematical techniques and optimisation algorithms in order to better represent the dynamic nature of the SC. There are various meta-heuristics, which are complementary for different parts of the SC, for example for scheduling problems there is a preference on genetic algorithms, simulated annealing and variable neighbourhood searches. Solutions from these types of heuristics can be furthermore improved when combined with ABM and simulation.

Supply chains are evolving networks with varied structures in a dynamic environment filled with uncertainties, making representation via mathematical modelling very challenging. The difficulty is further amplified due to the different types of variables: continuous and discrete, which are required for representation. It is observed that with simulation it is possible to represent different variables and with an array of scenarios, leading to the research of a hybrid of mathematical modelling and simulation to create optimisation (Acar et al., 2009). Simulation is used with other modelling techniques to strengthen and improve the algorithms or logic until the virtual and real-worlds are aligned. Therefore, although the model may start very simplified, through the use of simulation, performance analysis is conducted, leading to the introduction of higher levels of complexity.

Aggarwal and Lim (2012a) (Appendix A-3) propose that the lines of modelling in SCs are becoming more blurred and just like SCs the representation of the models are becoming more complex and dynamic. Further development of hybrid modelling in the supply chain is required in order to produce agile and effective decision support systems in the SC. In summary, this paper explores the development of DDM in the SC and discovers the challenges in modelling the dynamic and unexpected events, which can occur. Recently, there has been increased interest in academia for the implementation of hybrid modelling, for example combining ABM and simulation. ABM can provide a tool to enable effective DDM as it lends itself well to distributed systems. Simulation is used to strengthen and improve the model. However, in order to overcome the current shortcomings in modelling supply chains further development and application of ABM to supply chains is required in order to get the best qualities of the modelling technique. Section 3.2.1.2 explores the use of RFID technology to enhance supply chain agility when coupled with MAS, improving the potential for application.

3.2.1.2 RFID Enabled MAS: Enhancing Supply Chain Agility

Today, competent SCs seek competitiveness through personalised customer-orientated services, which can be reactive to any unexpected changes, whilst simultaneously compressing costs (Cao et al., 2006, Christopher, 2005, Swafford et al., 2008). Aggarwal and Lim (2013a) explore the creation of agile and responsive RTE networks to enable customisation as RTE become a prominent necessity in logistics networks and effective management of them is crucial in order to increase SC agility.

Improving agility requires the ability to react to unexpected changes, which in turn is dependent on high quality and accessible information (Swafford et al., 2008). As RTE networks expand; handling, monitoring and tracking becomes more challenging. A solution is to use RFID where pilots have shown promising results in providing the level of information visibility required to improve RTE management (Ilic et al., 2009, Daniel, 2009, Martínez-Sala et al., 2009). RFID enabled RTE can be monitored constantly in the system and not just at checkpoints, i.e. barcodes. If objects move then the corresponding agent can be updated with the information and re-scheduling decisions conducted. If the RTE are lost or move in the wrong direction then RFID can enable a trigger to be sent to alert the agent, which can then make the appropriate decision. Aggarwal and Lim (2013a) also explore the benefits of DDM to RTE management in order to enhance the agility, especially when combined with RFID.

Through agent application, a virtual sphere can be created to represent physical entities within SCs such as trucks, storage facilities and RTE. The virtual sphere can only function effectively if information regarding the physical entity and the environment is available. RFID can aid in the automation of data retrieval and transferral. The decision-making functions can be programmed within each agent and RFID can enhance the interaction with the environment and the communication with other agents.

The majority of the literature which apply agents utilising the value-added features of RFID technology is within the realm of manufacturing, and particularly production and assembly. Chen and Tu (2009) utilise the real-time function of RFID to enable agents to conduct responsive decision-making of products and the overall flow of the network more dynamic, whilst under JIT conditions. Trappey et al. (2009) develop an agent based system to assist in the selection of which parties should make moulds during production. RFID boosts the collaborative approach between different partners in the SC through the real-time visibility. Similarly, Zhang et al. (2011) utilises the real-time and wireless capabilities of RFID technology by attaching the tags to objects within manufacturing, which are represented by agents who make decisions on their usage based on the data collected. Zaeh et al. (2010) create flexibility in production by developing an agent based system supported with RFID to reduce the unexpectedness on the shop floor by providing up-to-date accurate data of every part of the production process. This approach resulted in a reduction of the time resources spent idle by 41%. Wang et al. (2009) stress the importance of RFID technology within providing flexible manufacturing systems due to the ability of receiving in and sending out constant status signals and requests with active RFID tags. This crucial information can be linked directly to the manufacturing planning systems where agents can schedule optimal plans.

Aside from manufacturing, the application of ABM and RFID technology in other areas of the supply chain is sparse and disjointed. Dias et al. (2009) capture the complexities of a global transportation system with a combination of RFID technology and GPS to enable agents to carry out decisions autonomously, ensuring responsiveness. Chow et al. (2007) develop an RFID based agent system to improve network processes and resource utilisation by reducing the number of pallets needed and reducing shipping errors, which enhances customer service. Yeh et al. (2011) use RFID to reduce stock-out problems by integrating shelf availability with agents to provide accurate stock availability encouraging better decision making. Jedermann et al. (2006) uses sensors and RFID to monitor gas, which is accompanied by an agent through its journey enabling a direct response if any changes are detected. Bose et al. (2009) apply RFID technology and ABM techniques to automobile logistics, reducing lead-times and achieving higher levels of on-time deliveries. As the literature demonstrates, benefits and improvements are evident but there are many more areas to investigate. Despite the dynamic and complex natures of RTE networks, research is limited and the application of RFID and ABM has not been fully explored. Aggarwal and Lim (2013a) (Appendix A-2) fill a gap in the literature by developing an ABM utilising the benefits of RFID to improve the RTE network and create logistical service agility.

In conclusion, DDM with an ABM approach requires accurate and real-time data with which RFID can assist. Agents can become informed, and therefore more reactive within the environment aiding in decision-making. This inclusion of real-time data can increase the intelligence of the agents. Aggarwal and Lim (2013a) propose the increase of RTE network agility through agents, which are enhanced by RFID technology. By creating an agile information and decision-making system with automation, a truly agile supply chain is in the process of being achieved. The paper has addressed the gap by setting out the architecture of an ABM to enable agile RTE networks and the ability to allow LSPs to action customised services.

3.2.2 **Motivation for MAS Usage - Discussion**

3.2.2.1 Comparison to Objects

Shoham (1993) introduced agent orientated programming (AOP) as an alternative approach to object orientated programming (OOP). Table 3.5 contrasts the two methods, where the ability of the agent to have beliefs and commitments followed with other attributes, makes it more advantageous than OOP. An object is encapsulated and will always do what it is programmed to do if the method is called, however an agent may not do what it is meant to as it will depend on the environment and other agents. The inclusion of autonomy sets agents apart as “objects do it for free; agents do it because they want to” (Wooldridge, 2009).

	OOP	**AOP**
Basic Unit	Object	Agent
Parameters Defining State of Basic Unit	Unconstrained	Beliefs, commitments, capabilities, choices…
Process of Computation	Message passing and response methods	Message passing and response methods
Types of Message	Unconstrained	Inform, request, offer, promise, decline…
Constraints on Methods	None	Honesty, consistency…

Table 3.5 Comparing Object Orientated Programming and Agent Orientated Programming (Adapted from Shoham, 1993)

3.2.2.2 Properties of agents in comparison to other techniques/heuristics (motivation for use)

Firstly, it is important to establish whether agent technology is applicable to industry in order to be compared with more traditional heuristics. Jennings (2000) asserts two reasons for the potential wide adoption of agents in the mainstream.

- It is logical to view the world as having many autonomous objects that perform actions and have actions performed on them. Therefore, within the software engineering domain, applications of agents are seen as the next logical step for exploration.
- Agents have a thread of control, which allows the action selection to be encapsulated, interactions to be re-used and also legacy software to be easily integrated.

Secondly, properties of agents such as flexibility, robustness, a distributed nature, and reduced problem solving due to localisation, makes them ideal for application in open-networked systems such as SCs. Although it is difficult to quantify the benefits of an agent-based approach in comparison to a conventional system, particularly in terms of “productivity, software reliability, system maintainability”, further benefits are explored through qualitative methods (Jennings, 2000).

Addressing the issues favouring conventional systems as described in Table 3.6, agents cannot reach the theoretical optima, and also there are lower levels of computational stability. However, conventional systems have a lower match to the real-world situation due to the high levels of abstraction of the models, which can make it invalid. The advantages of autonomous agents as described in Table 3.6 are discussed further in relation to RTE management application.

Issues	Autonomous Agents	Conventional Systems
Model	**Economics, biology**	**Military**
Issues favouring conventional system		
Theoretical Optima?	No	Yes
Level of Prediction	Aggregate	Individual
Computational Stability	Low	High
Issues favouring autonomous agents		
Match to Reality	High	Low
Requires Central Data	No	Yes
Response to Change	Robust	Fragile
System Configurability	Easy	Hard
Nature of Software	Short, Simple	Lengthy, Complex
Time Required To Schedule	Real time	Lengthy, Complex, Slow

Table 3.6 Comparing the Application of Autonomous Agents with Conventional Systems (Adapted from Parunak, 1996)

The RTE network consists of autonomous entities where decisions are made on a decentralised level. Agents are touted as the next generation of software modelling for complex and distributed systems with a natural application to SCs, introducing much needed flexibility (Huhns and Stephens, 1999, Jennings, 2000). Agents can be used to represent autonomous units within the supply chain on a micro-level and then the interactions between those units (i.e. the organisational relationships) are modelled on a macro-level within the construction of the society design (T Moyaux et al., 2006). Agents are specifically programmed to be able to interact during runtime and respond to unanticipated requests and generate the response whenever required. Therefore, the overall goal is not programmed on a micro-level but is reached during run-time (Jennings, 2000).

The RTE network has conflicting trade-offs, and therefore reaching one solution through a centralised format is inadequate. Agents can develop better solutions, building on current algorithms, but use tools such as negotiation and bargaining through their interactions (Huhns and Stephens, 1999). The decisions need to be coordinated from various factions within the RTE networks, where coordination can only be made possible by information sharing and a robust & interactive IT system to capture and support unpredictability (Um et al., 2010). Gathering, disseminating, analysing and utilising data can significantly enhance competitive advantage for the logistics service provider. Agents potentially offer better solutions and improvements to current data-based systems such as Enterprise Resource Planning (ERP), e-commerce and Advanced Planning and Scheduling (APS) (T Moyaux et al., 2006)

Finally, agents can be built individually and then added to a system and developed ad hoc (Jennings, 2000). So if things change or there are additions, then an agent can be developed to represent it and an extension is possible. For example, if there are changes to the RTE network such as the addition of facilities or customers, then agents can be constructed and added into the current system. Agents have the capability of flexibly forming, disbanding and maintaining organisations, which is critical for RTE network representation (Jennings, 2000).

3.2.3 Negatives of agent use

There are four main areas of agent application concern: information retrieval and monitoring, legacy and protocols, unpredictability of MAS performance & results and agent misuse.

Information Retrieval and Monitoring

An agent needs access to the right information from its environment and other agents. Nwana and Ndumu (1999) stress that the ability to design an agent, which can gather the right information from the appropriate sources would require hard programming and as changes occur constantly this would need to be conducted manually. Therefore, the system becomes a closed loop and limits the performance and capability of the agent. RFID technology can enable an agent to have constant monitoring of the environment, creating an open system, enabling the agent to access the right information and react to constant changes in information.

Unpredictability of MAS Performance and Results

Jennings (2000) identifies two drawbacks to modelling complex systems using agents. Firstly, the interactions and results are unpredictable during runtime, in addition the agent has a choice as to whether to accept or reject at all stages, therefore predictions are challenging. Secondly, it is difficult to predict how the system will perform overall if it is represented from the "bottom up" as there can be "emergent behaviour". If predictability is required of a system being developed, then there will be an endeavour to minimise the quantity and level of unpredictability. Therefore, the interaction protocols should be robust and the organisational structures rigidly defined to limit the scope of the agent.

Agent Misuse

Recently, there has been an increase in the application of agents to problems of the wrong context (Parunak et al., 2004). An agent should be applied in order to: reduce complexity through increased modularity, enhance speed and reliability, improve efficiency and encourage flexibility. If a problem is too large for a single agent then it should be split for multiple agents. The dissemination of MAS in the RTE problem domain is yet to be explored:

- RTE networks have multiple autonomous units, which are inter-connected. MAS enhanced with RFID can provide connectivity between the virtual and physical spheres.
- As agents are distributed with autonomous experts it enables decision-making to be decentralised across the RTE network.
- RTE, trucks and warehouses require constant information and knowledge updates much of which will be gathered through collaboration. Agents can enable this representation as they provide cooperative information systems.

- Agents can enable the resource allocation problem and scheduling of the RTE, trucks and warehouse capacity.
- MAS can aid in the achievement of both short term and long term goals of the RTE and of increasing company profitability and strategic competitive advantage.
- Agents can be built to create flexible interactions with any parties including customer, suppliers and other distributors.

3.2.4 Discussion

A MAS is applicable to a problem for modelling and simulation when it either improves understanding or produces solutions better than other heuristics (Nwana and Ndumu, 1999). In order to assess whether a problem is ripe for MAS application, a requirements-driven approach should be conducted. The capricious nature of agents offers a critical modelling tool to represent unpredictability, which occurs in the RTE management problem domain. As the level of control is dispersed throughout the network, the "bottom-up" approach can reduce the level of RTE network complexity. The distributed control leads to decentralised decision-making, which occurs on a local level with greater efficiency as an agent does not need to be distracted by other tasks given by external entities and can focus on the actual state of affairs locally.

The improvements in RTE network management would also require integration of advanced auto-ID technologies such as RFID technology. The retrieval and automation of data collection throughout the RTE network leads to increased innovative potential with MAS application. As data is attached to each object this can provide the ideal data support for its corresponding agent. The internal workings of an agent can be robust, however if the agent does not receive the right data from the environment or other agents to make decisions, then the performance of the system is limited and the agent capability only part-fulfilled (Nwana and Ndumu, 1999). Therefore, the information systems need to be designed in order to accompany the development of MAS applications in supply chains. RFID gathers data quickly and efficiently for each entity represented by an agent increasing the connectivity between the virtual and physical spheres. The combination of MAS and RFID provides a modelling and simulation tool, which is appropriate for RTE network application, and therefore software options to develop and simulate the agents are explored.

3.3 Agent Software Development

The development of software is a “significant barrier” when it comes to multi-agent systems (MAS) implementation to real-world applications (Railsback et al., 2006). The application of MAS to supply chains offers many benefits due to the ability to implement autonomy, and therefore conduct decentralised decision-making. This mirrors the distributed nature of the RTE network. However, due to the complexity and challenge of constructing MAS, many of the agent based benefits have not come to fruition. This section discusses the options of developing software agents by considering the available software packages, toolkits, languages, environments and the section is concluded with a discussion.

3.3.1 Software Packages

A software package such as Any Logic, is a java-based modelling platform with agent based capability, in addition to the ability to implement discrete event simulation and systems dynamics. Another software package, Mat lab has its own language and can be used to model a wide range of systems. However, the package is not built specifically for modelling agents and requires an extension in order to program a system from an object-orientation. With software packages such as these, they are user-friendly but the functions are generalised and built in order to solve a range of popular problems and systems, rather than specific problems such as in the RTE network management problem domain. In addition, they are costly, and therefore anything built would not be freely accessible.

3.3.2 **Agent Building Toolkits**

Toolkits can be made for various reasons and different types of users, for example scientists with little programming skills, computer scientists who wish to explore the back system or teachers who want a package that is easy to teach students (Nikolai and Madey, 2009). Railsback et al. (2006) compare the platforms; Net Logo, Mason and Swarm, which are described in Table **3**.**7**. However, they are rated based on factors such as file input and statistical capabilities instead of their applicability to certain problems. Nikolai and Madey (2009) fill the gap in the literature by expanding their survey of platforms and considering which would be most appropriate for particular problem domains.

Although toolkits can provide guidance to agent development, they often simplify the capability of the agent. Each toolkit has been built for a specific reason, and therefore application to other problem domains creates further challenges. Net logo was developed for social scientists and with nature systems as many of the base examples. In addition, the toolkits can have different languages and protocols, which could make extension of the agent capability difficult. Also, if the agent capability and interactions need to be programmed, then this will often require hard coding, and therefore although the user may not initially need programming skills, eventually programming skills will need to be developed. In addition, the toolkits need to have adequate debugging systems (especially if extensions occur) and often these are unsatisfactory (Nwana and Ndumu, 1999). Bespoke agent development can provide a more appropriate and flexible agent based model development environment.

3.3.3 **Programming Languages and Environments**

Java allows for the creation of the software quickly and robustly (Garcia-Flores, Wang, 2000). It is accessible as Java is available in multiple platforms including Windows, Mac, Linux and Unix. Java has strong typing as the compiler must be informed of each object's type and this allows the identification of errors which is crucial (Railsback et al., 2006). Therefore in comparison to languages such as C**, more assistance is provided to fix run-time errors. In addition, unused objects and code are removed, which improves the elegance of the code making it more readable for others and improves run times. In addition Java has excellent network communication facilities and a rich set of class libraries.

PLATFORM	WEBSITE
Any Logic A java-based modelling platform that supports discrete event, agent based and systems dynamics simulation	http://www.anylogic.com/overview
Jade "JADE (Java Agent DEvelopment Framework) is a software framework fully implemented in Java language. It provides a middle-ware that complies with the FIPA specifications and through a set of tools that supports the debugging and deployment phases. The agent platform can be distributed across machines (which not even need to share the same OS) and even during run-time".	http://jade.tilab.com/
MASON "MASON is a fast discrete-event multi-agent simulation library core in Java, designed to be the foundation for large custom-purpose Java simulations, and also to provide more than enough functionality for many lightweight simulation needs".	http://cs.gmu.edu/~eclab/projects/mason/
NETLOGO "Net Logo is a multi-agent programmable modelling environment".	http://ccl.northwestern.edu/netlogo/
SWARM "Swarm is a platform for which provides a conceptual framework for designing, describing, and conducting experiments on ABMs".	http://www.swarm.org/index.php/Main_Page
Agent Builder "AgentBuilder provides tools for managing the agent-based software development process, analysing the domain of agent operations, designing and developing networks of communicating agents, defining behaviours of individual agents, and debugging and testing agent software. The Run-Time System includes an agent engine that provides an environment for execution of agent software".	http://www.agentbuilder.com/index.html

Table 3.7 A Selection of Agent-based Modelling Platforms

A proven Java based IDE is Eclipse, which is an open source platform and can be used to build, deploy and manage software across its lifecycle (Um et al., 2010). It has extensible frameworks, tools and runtimes. It is advantageous as it has features such as "instant identification and help with syntax errors, menu tools for mundane tasks such as adding getters and setters, and the debugger" – not only for testing code but also for understanding how an ABM executes (Railsback et al., 2006). Eclipse is a multi-language tool, however many of the packages and plug-ins are java-based and so this offers more opportunity when extending agents and system capability, particularly in the application and maintenance stages of software development.

3.3.4 Evaluation

The aspects to consider when selecting how to build, test and deploy agents are: cost, ease of use, reliability, programming experience and efficiency. Firstly, many software packages and toolkits require licenses or particular operating systems to run on. Therefore, this incurs a cost when building the agents and also when trying to make the software available for others due to the barriers of purchasing licenses or different operating systems. This is not favoured in the development of the agent based system in this project. An IDE such as Eclipse and a language such as Java are free of charge and accessible to all, allowing transferability and application.

Secondly, although the software packages and toolkits are packed with easy to use GUIs, they lack the precision of custom-built software. The easy-to-use packages and toolkits are made to meet the expectations of various system requirements and although a custom-built agent system is more laborious to build, it provides for the specific needs of this problem. In addition, certain toolkits, such as Swarm, use a particular language objective C and a certain system Linux, which makes distribution of the software more difficult. Java is an object-orientated programming language, which has many support functions, whilst implementing the software. Eclipse is a well-developed IDE with a thriving online community and constant updates and plug-ins, which assists in the development of the agents and the simulation environment.

Thirdly, in regards to reliability, tool-kits encase blocks of code, which are often inaccessible, leading to inflexibility and if there are any discrepancies, then changes will be difficult or impossible to implement. As software packages are developed to cater for many problems, they are not proficient in modelling extreme cases for example, Any Logic has a tendency to crash with more complex models and Net Logo is very slow at running simulations of complex problems. In addition there are many errors when programming large parts of code, which is caused by poor debugging features and a lack of software maturity. These specific agent based environments are weak when building code in comparison to object-orientated pieces of software, which are more developed (Bordini et al., 2006).

Fourthly, the level of programming experience required is essential and the accessibility of programming agents for non-programmers. Although the tool-kits and software packages offer GUIs, when changing features for more complex systems, the code behind each method will need to be accessed and then modified. Therefore, a specific language will need to be learnt, for example in Mat lab. Once software packages are not being used, then a language to implement the system is required. Java has an extensive library and is a solid starting point to learning an object orientated programming language, and is also widely used, therefore easier to distribute.

Finally, a pre-built tool is a general instrument with a satisfying overall rate of efficiency, yet not particularly good in solving a specific type of issue with special requirements like in the RTE problem domain. However, a custom made piece of software will adhere to any extreme scenarios or test cases as it is built specifically for a particular design protocol. It is a big effort to make a bespoke system as it is very challenging and time consuming but it offers transparency and a platform in the RTE management problem domain, setting the foundation for agent application.

3.3.5 Conclusions

The construction and development of software for application in industry is a highly challenging task to undertake (Jennings, 2000). Although software is developed to reduce complexity of a problem; the construction of software is an arduous task in itself. Therefore, the time constraints of this project will put a limit to the application of the software in industry. When software is developed it can be difficult but external factors such as commercial availability, availability of tools and development environments and compliance with internal information structures can determine which software development environment is used (Muller, 1999).

The costs associated with programming the code and computing the algorithms needs to be calculated especially as bespoke systems are often used, as different supply chains have different goals and purposes (Lu et al., 2012). The need to use bespoke systems is reflected in the limited number of domains and toolkits available in ABM, which produces poor results when applied in a generic fashion (Nikolai and Madey, 2009). A bespoke system built through a powerful IDE such as Eclipse with Java, creates a system which can be tailored to the problem and simple to read and understand. The IDE can also be used to run simulations, which will form the basis of the analysis of RTE system behaviour.

Even though there are many languages, development environments and toolkits available, building and implementing agents is often a challenging task. Many of the toolkits lack maturity and the environments are not developed to cater for agent programming (Bordini et al., 2006). A custom-built agent based platform developed in Eclipse using Java, will provide a flexible transparent robust architecture, and therefore in the long run encourages flexibility for application rather than going for short-term efficiency of commercial solutions.

This section discussed various options of building an agent based system for the RTE problem domain addressed. The software needs to be developed in order to solve a specific problem with a degree of accuracy, which would be difficult to achieve by the toolkits and software packages discussed. Custom-built software enables refinement as a more realistic version of the RTE network is captured and modelled rather than applying a software package, which has a fixed amount of tools and capability no matter the range of complexity of the problem. It is a big effort to make a bespoke system as it is very challenging and time consuming but it offers transparency and a platform in the RTE management problem domain, setting the foundation for agent application.

3.4 Summary

Artificial intelligence applications to real-world problems have become more prominent over the decades, and particularly MAS application to supply chain management problems. In this chapter, an introduction to the relationship between AI, DAI, MAS and agents was given, followed by a discussion of the concept of agents where the properties, architectures and behaviours were explored. Autonomy is the key agent property, which distinguishes it from other modelling techniques and heuristics. This was followed by a description of developing MAS where interaction and communication with other agents and the environment are of paramount importance in order to represent the system to be modelled. However, often ABM is misused, and therefore it is critical to identify whether the RTE management problem will benefit from MAS application. RTE management is complex, with distributed control, decentralised decision-making, unpredictable and in real-time, therefore appropriate for MAS application. The real-time factor requires a robust and interactive information system to support the MAS developed, and this can be constructed with the integration of RFID technology. RTE network agents should be developed with a bespoke system using a powerful IDE, such as Eclipse and in Java language as it offers advantages over standardised and expensive toolkits. In addition, the simulation functions can enable the level of analysis required for this project in order to evaluate the merit of MAS application to the RFID enabled RTE network domain.

Based on the findings, in the project a custom-built agent model creates a modular system of representation of the RTE network. It provides flexible, transparent and robust architecture, and therefore in the long run encourages flexibility for application rather than the short-term efficiency of commercial solutions. Chapter 4 introduces the agent based architecture, which will be implemented to represent the RFID enabled RTE network problem.

Chapter 4

A New Approach to RFID Enabled RTE Management: Agent Based Modelling

An effective and efficient RTE network requires investigation of differing trade-offs between agile decision-making, leaner processes and the cost of increasing network visibility of the RTE and trucks. Therefore, in order to develop a model to represent a RTE network and analyse the RTE network behaviour, it is essential to capture the complex structure and dynamics of the RTE network through the components and interactions. Hence, the major aim of this thesis is to find a modelling and simulation technique to aid in RTE management improvement, whilst utilising the data advantages of RFID technology to increase visibility. After reviewing the literature and assessing the possible methodologies, a MAS approach is proposed to represent the RTE network and conduct simulations in order to improve the management. However, novelty of the research and complexity of RTE management and agent development leads to an exploratory research venture rather than a direct conversion from a real-world problem to a software solution. Consequently, in the simulations presented in chapter six, a standardised scenario is used in order to develop an agent-based model as a simulation tool for RTE network improvement.

This chapter discusses and describes the application of MAS to the RTE network problem. Section 4.1, reiterates the problem statement and analyses it to understand agent application and describes it through different unified modelling language (UML) diagrams such as: component, activity, package and class. Section 4.2 describes the components of the network through a component diagram presenting the RTE Batch and Truck agents. Section 4.3 introduces the Radio-Frequency Identification Enhanced Returnable Transport Equipment Agent Model (RFIDERTEAM) and explores the Java code through package and class diagrams. Section 4.4 describes the RTE delivery request flow with an activity diagram followed by a description of the RFID enabled data frames, which are required in conjunction with the model for higher levels of network visibility. Section 4.5 describes the Returnable Transport Equipment Agent Scheduling Algorithm (RTEASA) in detail with the assistance of an activity diagram. Section 4.6 describes the set-up of the simulation. The chapter is concluded with a discussion in section 4.7.

4.1 Project Aim

The major aim of the thesis is to improve RFID enabled RTE management. After reviewing and analysing the literature, it is observed that there is a requirement for both lean and agile approaches to be introduced in order to improve RTE management. A lean approach requires a reduction in the problems and costs. Currently, the major problems with RTE management include: low rotation rate, long cycle times and losses. These problems combined with poor visibility and management lead to an unresponsive and dormant network. Increasing agility in the RTE network will improve the network's ability to respond to unexpected changes and cope with dynamic elements. The introduction of effective decision-making units via agent-based RTE plays a significant role in increasing RTE network mobility. To introduce lean and agile components in the RTE networks and improve management there are varying levels of objectives.

4.1.1 **RFIDERTEAM Objectives**

Figure 4.1 illustrates the four levels of RFIDERTEAM objectives in this thesis in order to improve the RFID enabled RTE network. Cost advantage via the RFIDERTEAM for RTE network improvement is achieved by two streams: removal of RTE waste and removal of fleet waste, with the aim of creating a leaner network. The first stream, removal of RTE waste, requires the implementation of four methods: reduction in the quantity of days a RTE is empty, a reduction in the time RTE is idle, decrease of RTE cycle time and an increase of RTE rotation rate. Although the focus is on improving the RTE network through the achievement of objectives set in the first stream, the fleet has a significant impact on RTE network productivity .The second stream, removal of fleet waste, has three components for focus: reduction of empty miles, increase in the number and quantity of loadings and an evaluation of the RTE batch combination selected. The aim is to reduce the amount of time the truck is idle and utilise the driving hour to pick-up as many loads as possible in the reverse flow back to the distribution centre.

The RFIDERTEAM implements agility in the RTE network by increasing responsiveness, creating resilience and utilising the functionality of decentralised decision-making. At this point the decision-making goes beyond a RTE level to include the products and clients to gain strategic advantage. Firstly, the RTE network needs to be able to respond to events, particularly unexpected events in an efficient manner. Secondly, the model aims to foster resilience in the RTE by removing vulnerabilities, such as losses and break-downs. Finally, the decision-making is localised so that the agent representing the object makes efficient and effective decisions in a reasonable time-frame without having to include the whole network. An agile system responds to changes; however a network also requires flexibility which refers to how the network responds. A flexible RTE network needs to be able to reconfigure efficiently by returning the RTE as soon as possible back to the point of use whilst adhering to the product's requirement and client's objectives. By utilising the resources of the network, such as the fleet and information systems, flexibility is built into the system. In the RFIDERTEAM, the flexibility achievement is complex as it is multi-layered, with a variety of products, RTE and numerous clients. A client of a LSP may require a reduction in their inventory whereas another client may prefer a reduction in their product's lead-time. Each client may be utilising the same resources of the LSP but will require different services, therefore LSPs need to offer service differentiation. Therefore, the model offers flexibility by representing different types of supply chains and the ability to meet the product requirements and client's objectives.

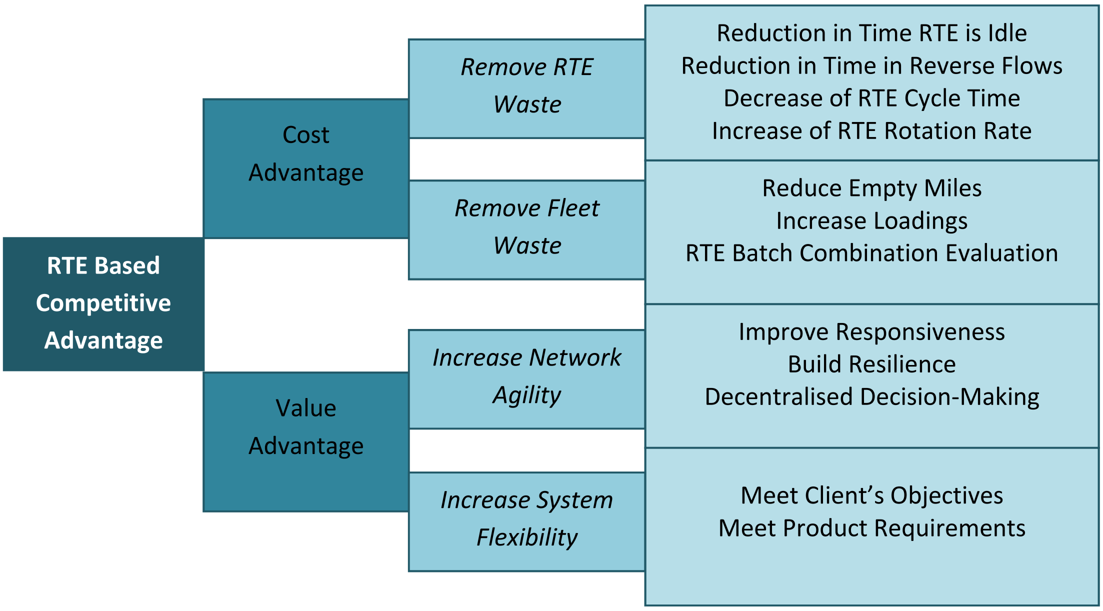


Figure 4.1 Illustration of RTE Network Competitive Advantage Objectives

The model developed aims to improve RFID enabled RTE management by simulating the features above, which will firstly create a leaner network as the current resources will be utilised further and remove waste. Higher levels of agility will enable the network to react to changes like fluctuations in demand and failures more effectively. The model will encourage a lean RTE network, whilst mobilising current resources to cope with unexpected events. Agent-based modelling is utilised to plan and schedule deliveries with the focus of reducing the amount of empty miles, whilst maintaining or reducing the number of trucks currently in the network. The model provides a platform, which demonstrates different logistics triggers for a particular scenario, improving efficiency of the RTE and truck movements, whilst dealing with agile or lean aspects such as a sudden surge of demand or lost network capacity.

As the RTE and truck agents negotiate loads to produce a schedule they require knowledge, which is based on the represented object and the environment. RFID technology increases network visibility through an object level. The essential use of RFID technology application in this problem is discussed further in section 4.4.2.

4.1.2 **Model Initialisation and Development**

In chapter four, the research describes the conceptual model that integrates and coordinates the components, activities and interactions of a RTE network, describing the MAS application and simulation. The model introduces new concepts, which will enable the coordination of RTE, warehouses, distribution facilities, customers and orders with the aim to improve the lean and agile features of the network, improving RTE management. As discussed in chapter three, the RTE network has a natural synergy with multi-agent systems (MAS), particularly when combined with RFID technology. The model will be able to assist in understanding RTE behaviour, aiding in identification of the measures required to improve the network. Furthermore, simulations are conducted in order to achieve better outcomes and provide an effective and interactive tool for future RTE network management. The analysis, design and implementation of MAS for RTE management are described and discussed further.

The controller class is developed in order to listen to any changes and real-time information, which may be relevant to the decision-making process for each agent; however the simulation is conducted in real-time, as the model acts as a representation of a real-world RTE network, and therefore the functions have been programmed in order to observe the behaviour.

4.1.3 **Unified Modelling Language (UML) 2.0**

The standard modelling language used to represent a problem or software methodology is Object Management Group's (OMG) Unified Modelling Language (UML) (Bauer and Odell, 2005, Martin, 2003). UML diagrams are used to model systems on a conceptual level, as a specification or in order to assist with implementation (Roff, 2003, Martin, 2003). Diagrams provide the software developer with a tool to communicate ideas and model visually the structure, behaviour and organisation of the model (Martin, 2003, Bauer and Odell, 2005). This provides accessibility of software components to users which in this instance are the LSPs and their clients; demonstrating the intended benefits of RFIDERTEAM. The thesis presents a new approach to RTE management through the use of software agents, which utilise RFID based data frames on a decentralised level. It is essential to describe and explain the architecture and detailed workings of the model, in particular conveying the algorithms and interactions. Therefore, this chapter will use standardised UML diagrams in order to illustrate, describe and analyse the code of the model.

There is trepidation in the ability of UML to represent agent-based modelling as agent features such as grouping and cloning cannot be represented by the standard (Bauer and Odell, 2005). The inability to present and showcase agent-based research is one of the main reasons for the increasing gap between industry and theoretical application of agents (Bauer and Odell, 2005). There was an attempt to form an Agent Unified Modelling Language (AUML), however the current UML 2.0 standard and its extensions offer a "smoother" transition from object orientated to agent based representation, therefore AUML was not developed further (Bauer et al., 2001). Thus generally UML 2.0 is used as a base for developing a modelling language for agent application, adding any features to the diagram as required. The four main UML diagrams used in this chapter provide a visual interface and a smooth transition from the RTE problem domain to the RFIDERTEAM's source code. The diagrams are: component diagram, package diagram, class diagram and activity diagram.

4.1.3.1 Component Diagram

The component diagram illustrates named physical parts of a system which are called the components (Scott, 2004, Roff, 2003). Interfaces required to and from components highlight the dependences, and therefore the relationships (Roff, 2003, Bauer and Odell, 2005). An interface is represented by a ball and socket notation, which signifies component requirements for the completion of its tasks (Scott, 2004, Roff, 2003). Figure 4.2 illustrates the ball and socket notation between a truck agent and a RTE agent. The RTE agent provides dates of delivery and the truck agent has a socket notation to accept this information, which is necessary in order for it complete its tasks.

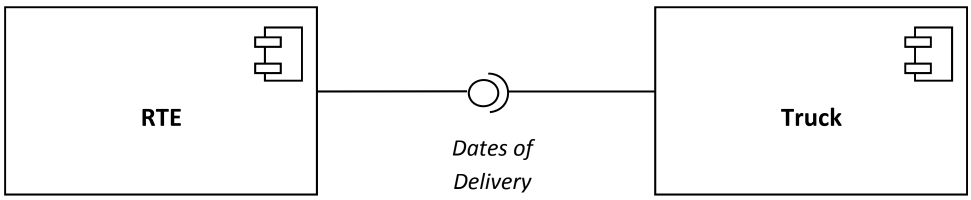


Figure 4.2 Illustration of Component Diagram Notation

4.1.3.2 Package Diagram

Packages are a useful tool for software developers to manage a model as they group classes with common features together, which is particularly appropriate for MAS with features of sub-teams (Scott, 2004, Roff, 2003). Each element in the package must have a unique name but it will not clash with other elements in other packages, which have the same name (Martin, 2003, Scott, 2004). A package is shaped as a file, which is named and classes within the package appear in the body of the file as illustrated in Figure 4.3 (Scott, 2004, Martin, 2003). The Control package has three classes: Controller, Routing and Model Variables. There can be dependencies between packages as they can import classes, and therefore arrows are used to illustrate the relationships (Martin, 2003). The closed arrow states that the Control package imports the Agents package. The open arrow represents an association between the Agents package and the Routes package.

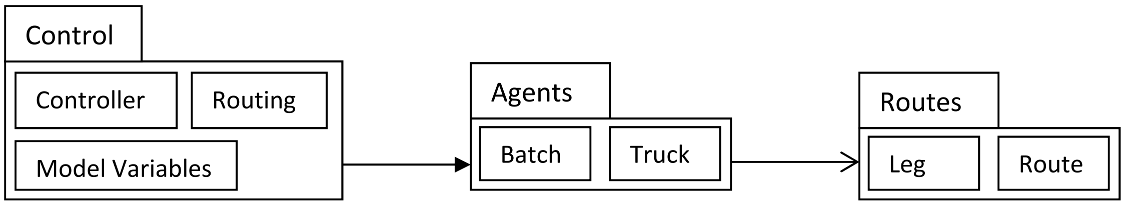


Figure 4.3 Illustration of Package Diagram Notation

4.1.3.3 Class Diagram

Classes are the foundation of software modelling as they represent the components and relationships of a system to be modelled (Scott, 2004). The class diagram will illustrate interactions, dependences and the structure of each class through its attributes and operations, enabling a visual understanding of the Java code (Martin, 2003, Roff, 2003). Figure 4.4 illustrates three compartments of a class, which contain the name, attributes and operations. Attributes are values, which give the object a "sense of state" (Scott, 2004, Roff, 2003). An operation is a method, which an object may evoke affecting the object's "behaviour", giving the object a "state of functionality" (Roff, 2003, Scott, 2004).

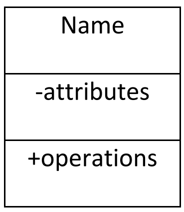


Figure 4.4 Illustration of a Class Diagram

Each attribute and operation may have a different level of visibility which is denoted by the following symbols at the beginning of each line (Scott, 2004):

- [~], package, allows other classes in its package to see and use the member
- (+), public, signifies that the attribute/operation is used by any object as it is fully observable by any class
- (#), protected, attribute/operation is only available for objects, which belong to the subclasses

- (-), private, attribute/operation is only be seen and used by objects of the class within which it belongs.

Another function, which is represented by a class diagram, is multiplicity. This specifies how many objects are affected by other objects of a different class (Roff, 2003). The level of multiplicity is indicated with numbers and placed at arrows out of and in to classes (Roff, 2003):

- 0..1, no instances, or one
- 1, exactly one instance or any other integer.
- *, many instances
- 1..*, one or more instances

4.1.3.4 Activity Diagram

An activity diagram is used to analyse complex behaviour and represent the flow of control between activities (Roff, 2003, Scott, 2004). An activity is a motion of actions, which cannot be interrupted during execution (Scott, 2004). It provides an overview of a system and allows specification of how control changes as the conditions change (Roff, 2003). It also specifies the conditions which need to be met in order for the goals to be accomplished (Roff, 2003). There is no goal of an activity diagram, however there is an end point, which is viewed as an activity to support a goal on a macro-level, enabling the ability to represent agent behaviour (Bauer and Odell, 2005).

A rectangle on the edge of an activity diagram is called an activity parameter (Scott, 2004). This is an activity node which represents a flow to or from an activity, to another one. There are several nodes, which are used to represent the flow of control as illustrated in Figure 4.5:

- Activity; an on-going set of actions
- Initial node; flow of control starts and an activity is evoked
- Decision node; holds a boolean expression, which must be completed before travelling along the activity edge

- Fork node; a flow which is represented by an arrow, leading into a rectangle, which then splits the flow into multiple parallel flows
- Join node; allows multiple flows to collect and wait until all flows leading into the join are complete before releasing one flow
- Flow final node; is a circle with a cross, which represents the end of a particular flow
- Activity final node; is a black circle with a white outline which represents the end of an activity

These diagrams will be used to illustrate and describe the components and activities developed for RFIDERTEAM with Java coding. Section 4.2 describes the components and functions of a RTE network.

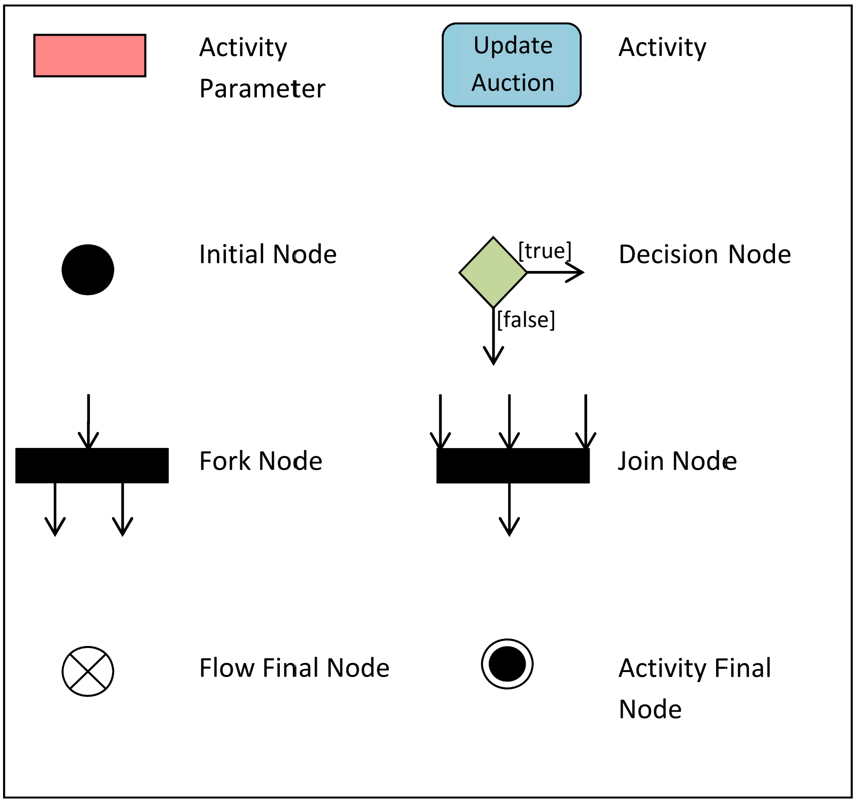


Figure 4.5 Elements of An Activity Diagram

4.2 RTE Network and Model Components

Logistics service providers (LSPs) use their expertise to manage their client's RTE, which is challenging in the highly dynamic outbound network environment. Multiple customers coupled with a variety of product types leads to a network of multitudinous RTE, where seeking competitive advantage through cost reduction and high service quality is demanding. There is a gap in the literature to present an agent-based model of an outbound network where the focus is on increasing RTE productivity and viability. Chapter 4 presents a RFID enhanced RTE Agent Model (RFIDERTEAM) and in this section the components: outbound network structure, RTE Batch agent and Truck agent, are described and discussed.

4.2.1 Outbound Network

An outbound network is used to model the environment in which the RTE are moved. Figure 4.6 illustrates a sample outbound network with three varieties of locations. The distribution facility (DF), from which all trucks depart from at the start of their shift, and return to at the end of their shift, is signified by the rectangle labelled *DF* x, where x is the distribution facility number. The warehouse, from which trucks conduct intermediate stops and exchange RTE, is signified by the rounded rectangle labelled *W* x, where x is the warehouse number. Customers, from which truck agents may load or unload RTE, are represented using circles labelled with *C*x, where x is the customer number. Lines between the locations indicate possible routes and which distribution facilities and warehouses serve each customer; each line is labelled with the number of hours it takes to travel between the two locations. Since it will always take a finite time to load and unload a truck, the facility is included in the model to have a single fixed load or unloading time within the model.

4.2.1.1 Distribution Facility (DF)

A DF is the biggest storage compound in the network. All deliveries start and end at the DF. There may be multiple DFs depending on the LSPs or client's network. Goods move from the DF to regional facilities, which are called warehouses.

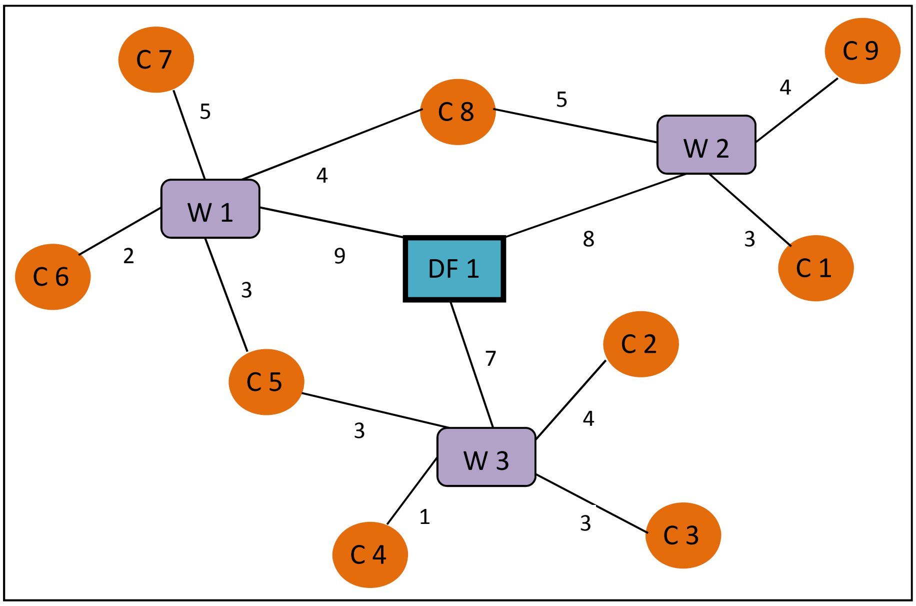


Figure 4.6 Example of an Outbound Network

4.2.1.2 Warehouse

Warehouses are regional storage facilities, which are located between DFs and customers. Goods are trunked; loads are revised and cross-docked for efficient delivery to customers. Each time a RTE moves through a warehouse, unloading and loading times will be added to the lead-time.

4.2.1.3 Customer

The customer is a client of the LSP and is awaiting delivery for the goods carried by the RTE. The customer's facility is served by at least one warehouse. The customer receives the RTE, unloads and prepares it for pick-up. The RTE also incurs an unloading and loading time.

Currently the deliveries between customers has not been considered as this makes the interaction and processing time longer and more complex whereas the model developed focuses on building an interaction to create more loads.

4.2.2 RTE Batch Agent

RTE has a dual purpose: to ensure goods reach their destination and then return back to the correct facility in order to start another cycle. The RTE has conflicting decisions such as getting to the customer on time versus increasing its own utilisation rate, making RTE an object appropriate for agent representation. There is a multitudinous number of RTE throughout a LSP's network at any one time. To have an agent represent each RTE would be very exhaustive computationally and time-intensive. Also, there are often multiple RTE, which need to go to the same destination, and therefore the RTE is batched together. Thus, a RTE batch agent is created to represent multiple different types of RTE.

Table 4.1 states the aim, motivations attributes, rules and tasks of a RTE Batch agent. The RTE Batch agent aims to increase its utilisation and is motivated with the following: on time deliveries of goods, minimisation of reverse delivery times (reduction of empty miles) and selection of trucks for delivery based on the shortest available cycle time. There are a variety of attributes a RTE Batch Agent is assigned: a quantity q, a weight w, a volume v, a destination, a forward delivery date and a return delivery date, which is equal to the forward delivery date plus the value of the batch processing day. Since it will always take a finite time process the batch, unload, remove the goods and prepare for pick-up then in the model, a modifiable single fixed processing time is added as the batch processing time.

The date for delivery is assigned in order to allow the batch to be unloaded, goods removed and then prepared for pick-up. RTE Batch agents aim to find trucks, which are able to deliver them to their destination on the forward delivery date, and return them to the depot on the return delivery date. Therefore, one of the agent's rules is to select an acceptable offer for delivery where the RTE Batch agent is required to negotiate through the RTE Agent Scheduling Algorithm (RTEASA). In case a RTE Batch agent does not receive an offer, it splits its batch in half and represents one half, whilst another agent represents the other half. A smaller size increases the number of offers for delivery.

The agent is also required to ensure that the reverse flow for the RTE batch is only scheduled once the forward delivery date is scheduled. The two main tasks a RTE Batch agent must conduct are: to form and dissolve as an RTE Batch for delivery enters the system and update the RTE batches' delivery time schedule to ensure that the correct information is assigned to each RTE batch.

RTE BATCH AGENT	
Aim	Increase utilisation
Motivations	Forward on time delivery Minimise reverse time delivery Pick shortest cycle time overall
Attributes	Quantity *q* Weight *w* Volume *v* Destination Forward delivery date Return delivery date
Rules	Select acceptable offer for delivery from truck to transport load Negotiate through RTE Agent Scheduling Algorithm (RTEASA) Split RTE Batch in half if no offers are received Ensure reverse flows are only scheduled once forward flow is scheduled
Tasks	Form and dissolve for each batch of RTE Update the RTE Batches' delivery time

Table 4.1 Description of the RTE Batch Agent's State and Functionality

4.2.3 **Truck Agent**

A truck is a vehicle, which is responsible for transporting items between locations. A truck has a multitude of factors, which need to be considered such as route length, load and driver's hours. Factors such as these are at odds with the RTE Batch agent's aims such as quick return delivery. Therefore, it is proposed that an agent represents each truck in the network forming a Truck agent. Table 4.2 describes the state and functionality of each Truck agent.

The Truck Agent aims to achieve a full truck load, whilst developing profitable routes when scheduling in RTE to be transported. The Truck agent is motivated to schedule full truck-loads on reverse and forward flows, whilst minimising the costs. Therefore, when considering making stops the agent has to account for loading and unloading costs, which both aid in achieving a full truck-load, utilising the capacity but could minimise the route and delivery options.

Each Truck agent represents a single truck within the system holding a set of attributes. Truck agents have a weight capacity *c*, which specifies the maximum weight of RTE they are able to carry at any given time, a volume capacity *v*, which specifies the maximum volume of RTE they are able to hold at any given time, and a current load *l* and *k*, which encodes the weight and volume respectively of the RTE they are currently carrying. Truck Agents also have a shift length, which limits the maximum number of hours each day they spend transporting RTE.

TRUCK AGENT	
Aim	Achieve full truck-load and profitable routes
Motivations	Schedule full truck-load on forward and reverse flows Minimise truck costs Utilise the capacity by maximising the load collection on scheduled routes within the allotted time
Attributes	Weight capacity *c* Volume capacity *v* Current load *l* and *k* Shift length
Rules	A RTE batch can only be scheduled once a contract has been agreed No initial route at the start of the simulation Offers of delivery are based on current commitments and availability An offer can only be initiated if it adheres to the truck's constraints An offer once accepted, is fixed Each stop incurs an unloading and loading cost
Tasks	Receive requests from RTE Batch Agents Once initial route is developed, bid for RTE to increase truck load Update transport route Update truck's current load and shift length

Table 4.2 Description of the Truck Agent's State and Functionality

Truck agents only transport RTE if they have agreed a contract, specifying a destination and delivery and return dates, with the corresponding RTE Batch agent. Initially, Truck agents start the program with an empty route. Once the simulation is running, Truck agents make an offer for a batch in advance, therefore on any given day a truck agent may already have stops assigned due to the acceptance of batches from previous days. A route is a series of stops to be made by the Truck agent on that day; each stop specifies a particular location at which a Truck agent may either pick up RTE for return to the distribution facility/warehouse, or unload RTE it is currently carrying.

A Truck agent is required to receive bids from RTE Batch agents and is tasked to accept bids until a full truck load, time-constraint completion or full route is achieved updating the truck's schedule and attribute balance. The rules and tasks highlight a balancing act, which the Truck agent is required to achieve between a full truck-load and maximising the hours available for truck's running time, which is based on the driver's hours as specified by regulations. Figure 4.7 illustrates the different solutions a truck may achieve and how this impacts the truck's viability and utilisation rates.

The graph illustrates the load capacity and route length of a truck. The acceptable solution space represents the maximum load capacity, which is a full truck-load and the maximum route length, which is nine hours and any solutions below these constraints. A solution which goes beyond these parameters would be infeasible in the real-world due to driver's hour's restrictions and the size of the truck, which are fixed. The green-lined solution in Figure 4.7 offers a full truck-load, however it only utilises three hours of the time available to make deliveries. Therefore, the truck would have to go back to the depot and wait for the next batch of bids, which is costly and inefficient. The red-lined solution fully utilises the running time of the truck, however the full-load capacity has not been reached, therefore leading to an underutilisation of truck space and a lost opportunity. The optimal RTE batch combination is defined by the orange line. This is where the batches selected by the Truck agent enable a full truck-load, whilst staying within the nine hour time frame. The Truck agent will continue to bid for deliveries until the load capacity or route length times have reached their maximum. It is the Truck agent's prerogative to maximise its loads or the available time for delivery for that day's auction, before it is closed.

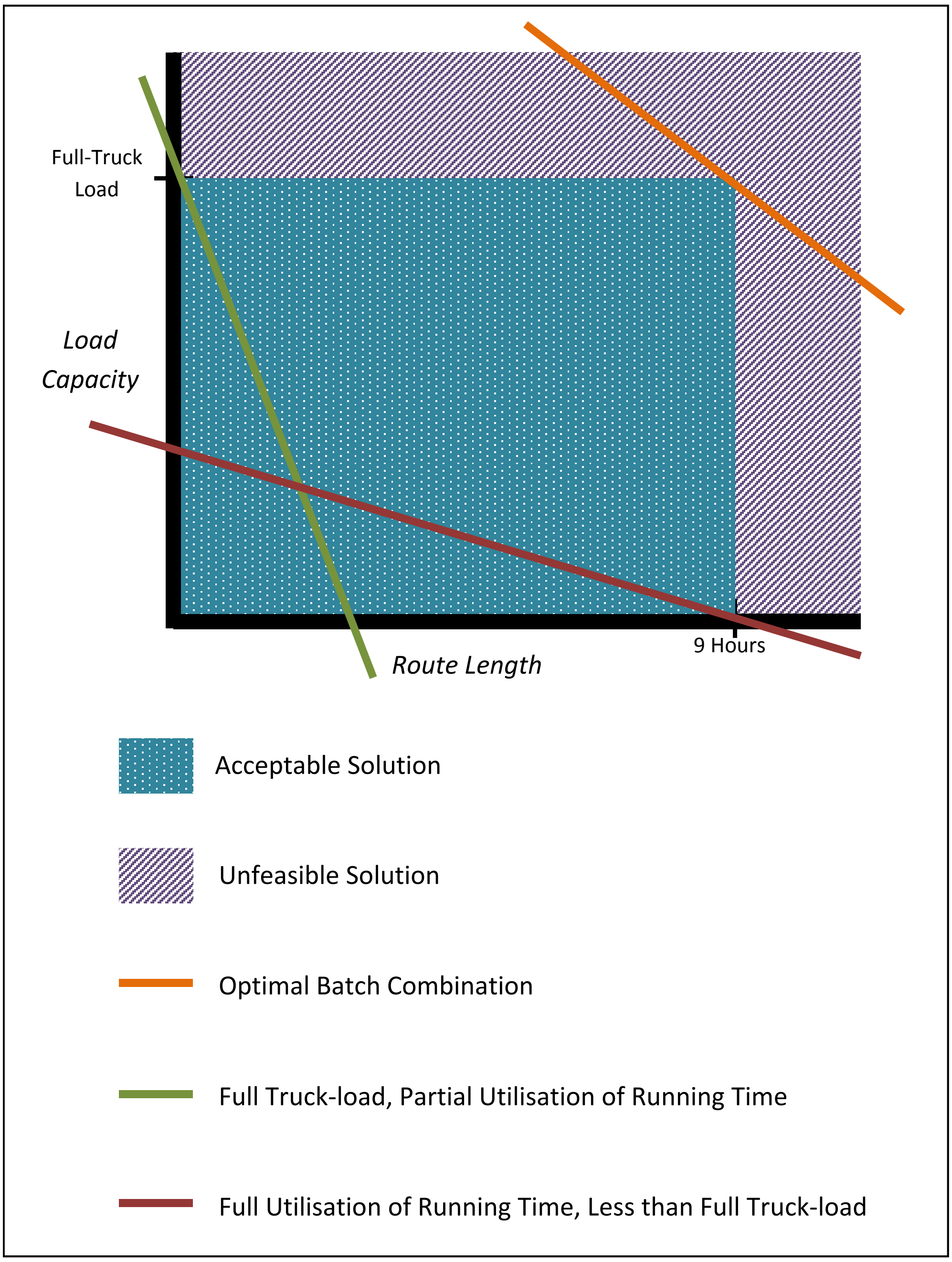


Figure 4.7 Trade-offs for a Truck Agent

4.2.4 Component Diagram

The RFIDERTEAM develops the mechanism for effective interaction between the RTE Batch agent, Truck agent and their environment, which is the outbound network. The main components of the system and the important interfaces with which they interact with are described in Figure 4.8. There are five main components described: RTE delivery request, RTE, Truck, Route schedule and Truck constraints. The RTE delivery request component is essential as it begins the simulation and sets the parameters with which the RTE Batch agent uses. It utilises various data gathered from the Truck and RTE Batch agents, defining the constraints it will have to negotiate in order to be scheduled. If an offer for delivery is accepted, it is of paramount importance that the information in relation to the RTE, trucks and network is updated as these schedules are utilised throughout the simulation by the main components. The Truck agent updates the route schedule and the truck's constraints whereas the RTE Batch agent updates the RTE delivery request and RTE's attributes. This forms a continuous cycle of interaction between the main components in the model.

4.3 RTE Agent Model

This section explores the Java code used to build the RFIDERTEAM. A package diagram provides an overview of the different classes and how they are sectioned in the software model. In section 4.3.2, classes are discussed further with class diagrams for each class. An explanation of the code for the activities and algorithms developed, particularly the interactions and environment created, is explored in further detail in sections 4.4 and 4.5. Section 4.6 describes the initialisation of the simulation. Developing a custom-built agent based system is a very arduous and time-consuming task for a non-programmer. Therefore, in order to adhere to time constraints of the project the expertise of a Java programmer, Michael Winsper, were acquired in order to program the agents in Eclipse. By using a professional programmer, there are less chances of incurring mistakes. In order to achieve a deeper exploration, an overview of the package of code is required.

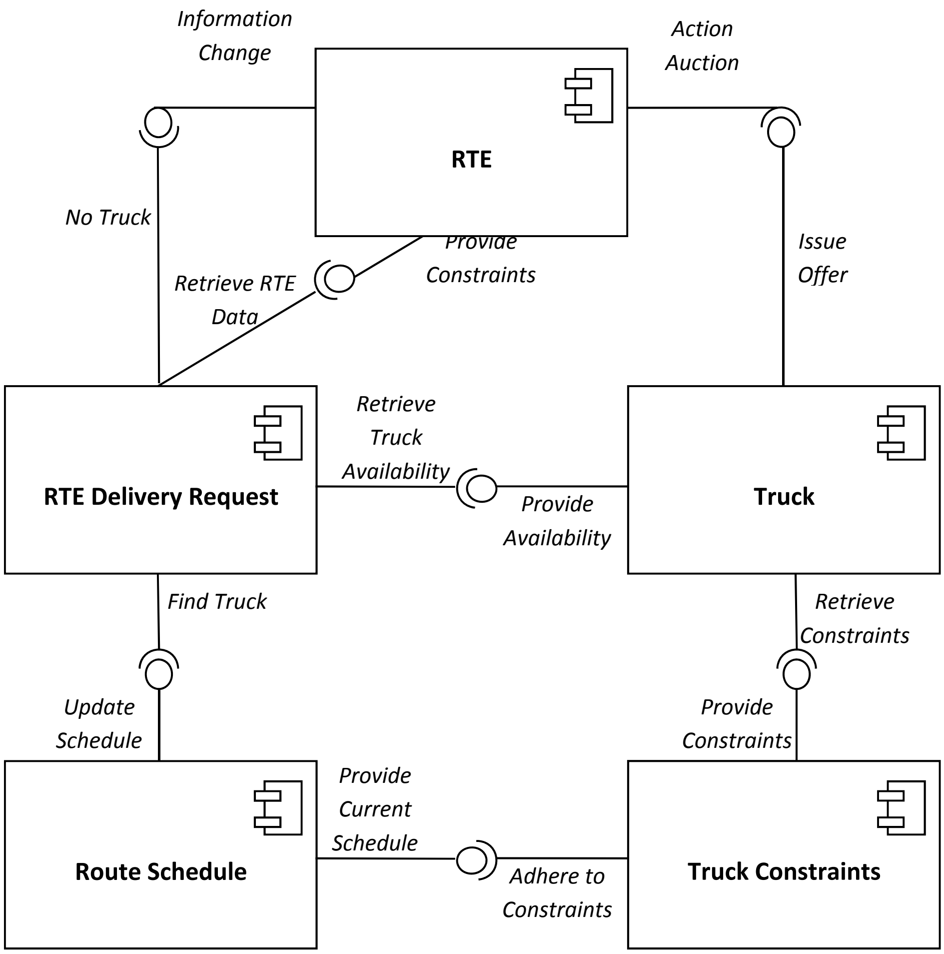


Figure 4.8 Component Diagram for the RTE Agent Model

4.3.1 Package Diagram

Figure 4.9 illustrates a package diagram for the RFIDERTEAM where there are eight packages holding twenty different classes. The figure illustrates each package and the class(es) which belong to the corresponding package. For example, the Agents package holds two classes: Batch and Truck. The Agents package holds the rules for each agent which moves through the outbound network. Packages are linked to each other via open arrows which represent the access rights and level of visibility between each package. The Control package is an important package as it starts and controls the simulation, and therefore interacts with all the other packages in the model. The classes in the packages are explored further through class diagram structures.

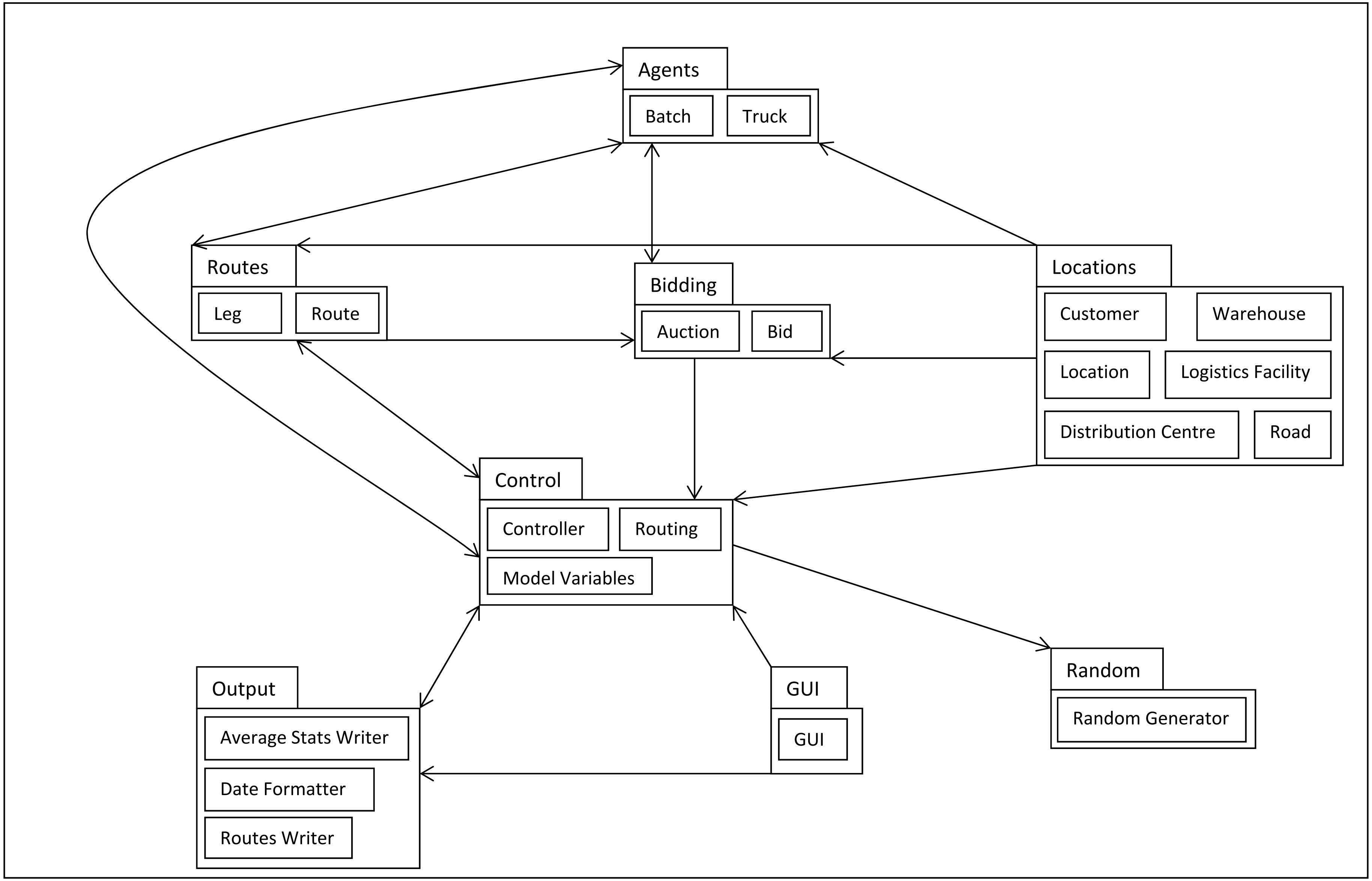


Figure 4.9 Component Diagram for the RTE Agent Model

4.3.2 **Class Diagram**

The package diagram provides an overview of the packages of code which have been developed, giving an overview of the interactions. This section will take each class and explore the attributes and methods in detail. This will specify all the different items that are considered in each class within the RTE Model and how the environment and interactions have been developed. As there are many classes some of which are very code intensive, the most prominent classes and their main features are discussed, which are: Batch, Truck, Leg, Route, Bid, Auction, Controller and Model Variables. The remainder of the classes are found in Appendix C. The annotated code for each class, which provides a guide to each class, is found in Appendix D.

4.3.2.1 Batch

The Batch class models one instance of a RTE Batch agent. A RTE Batch agent represents a quantity of RTE with an origin, a destination, an intermediate warehouse that the batch will accept indirect delivery to, a quantity, a volume, a weight and preferred delivery and return dates, further attributes are described in Figure 4.10. Other attributes, which are defined in this class are:

- *daysLateOnDelivery*; the difference between the batch’s actual delivery date and the batch’s original requested delivery date.
- *daysLateOnReturn*; the difference between the batch’s actual return date and the batch’s original requested return date.
- *daysPostponed*; the number of days that have passed since the auction's original delivery date without the batch having secured bids.
- *auctionPostponed*; whether any days have passed since the auction's original delivery date without the batch having secured bids.

Batch
-name: String -quantity: int -weight: double -volume: double -deliveryDate: int -returnDate: int -daysPostponed: int -daysLateOnDelivery: int -daysLateOnReturn: int -destination: Location -origin: Location -acceptableWarehouse: Location -currentLocation: Location -hasHalved: Boolean -auctionPostponed: Boolean -provisionalForwardBid: Bid -provisionalReverseBid: Bid
+runAuction(boolean) +selectBestClosedForwardOffer(bestBid:Bid) +selectBestClosedReverseOffer(bestBid:Bid) + selectBestOpenForwardOffer(bestBid:Bid) +selectBestOpenReverseOffer(bestBid:Bid) +createCopy(b.setHasHalved:Batch):boolean

Figure 4.10 Class Diagram of Batch Class

When the auction is opened and run, the RTE Batch agent will make a bid based on the current delivery requirements of the RTE batch it represents. The possible bids are; forward bid, from the DF to the warehouse, forward bid from the warehouse to the customer, direct reverse bid, reverse bid from the customer to the warehouse or reverse bid from the warehouse to the DF. The auctions are conducted according to a multi-stage algorithm, the RTEASA, which is described further in section 4.5. At the auction, either an acceptable set of bids is found, otherwise it will return false. There are various different return types, which may be selected if an acceptable set of bids is found; these are described in Figure 4.10:

- selectBestClosedCombinationOffer; selects the best bid that offers both forward and reverse deliveries with dates according to the batch's preferred delivery or return dates.

- *selectBestClosedForwardOffer*; selects the best bid that offers forward delivery only with a date according to the batch's preferred delivery date.
- *selectBestClosedReverseOffer*; selects the best bid that offers reverse delivery only with a date according to the batch's preferred reverse date.
- *selectBestOpenForwardOffer*; selects the best bid that offers forward delivery only with the closest date to the batch's preferred delivery date.
- *selectBestOpenReverseOffer*; selects the best bid that offers reverse delivery only with the closest date to the batch's preferred reverse date.

4.3.2.2 Truck

A class to represent a Truck agent where trucks will have: names, volume capacities, weight capacities, maximum shift lengths and origins as illustrated in Figure 4.11.

Truck
-name: String -volumeCapacity: double -weightCapacity: double -currentWeight: double -currentVolume: double -currentLoad: ArrayList <Batch> -currentShiftLength: double -maxShiftLength: double -routes: Route[] -origin: Location -currentLocation: Location
+setUpRoutes(tempRoutes:routes) +bidAccepted(routes:Route[]) +bid(boolean) +calculateBestRouteWithNewLeg(bestSoFar:boolean) +printRouteForToday(routes[0]) +printAllRoutes(routes[i])

Figure 4.11 Class Diagram of Truck Class

The Truck agent sets up its route in order to reflect the new routes proposed by the bids, which have been accepted by the RTE Batch agent. At day 0, an array is populated with empty routes. At days past 0, new routes are added to the end of the array at each day. A bid is placed and an auction process will be conducted. Each (else) if statement deals with specific parameters, including whether the bid is for forward or reverse delivery, whether the bid is for the assigned delivery date (closed) or the nearest available (open) date and whether the bid is for indirect or direct delivery. Provisional forward or reverse routes are checked to stop a truck offering indirect intermediate delivery on both legs on a single day; in this case a direct bid makes more sense. The return parameter *calculateBestRouteWithNewLeg*, finds the best place in a given route to add a new leg and returns the new route with the shortest distance. Then the truck's route for the day is recorded and printed followed by a print out of all the truck's planned routes.

4.3.2.3 Leg

The Leg class, models one leg of a particular route; the attributes and methods are described in Figure 4.12. A leg involves either travel between two locations with no loading and unloading (i.e. when returning "home" empty), travel between two locations with loading and unloading (normal behaviour) or loading and unloading at a single location (i.e. when loading multiple batches at a single location). Every time a batch is loaded or unloaded during a leg, the *calculateNewDetails* method updates the delivery details of each batch, which modifies the current weight and volume loads for each truck on each leg modified.

Leg
-location:Location -weightAtStart: double -weightAtEnd: double -volumeAtEnd: double -volumeAtStart: double -loadAtStart: LinkedList<Batch> -loadAtEnd: LinkedList<Batch> -batch:Batch -load:boolean
+calculateNewDetails(setVolumeAndWeightAtStart(), setVolumeAndWeightAtEnd())

Figure 4.12 Class Diagram of Leg Class

4.3.2.4 Route

Figure 4.13 illustrates a class that models a series of legs, which make a route. A truck adds a new leg to its route with the *addLegAtIndex* method, which will add a leg at the appropriate index, which is at the start or end of other legs already in the route. The *getLength* method displays the length of the route. Each leg where loading or unloading occurs, adds a finite amount to the route. This is added to the time taken to travel between locations, which are defined in the Model Variables class. For legs where no action is performed, only the travel time is added. The *allLegsValid* function ensures that, given the list of routes whether the weight and volume remain within the truck's loading capacities at all times.

Route
-day: int -legs: ArrayList <Leg> -owner: Truck
+addLegAtIndex(leg:Leg) +calculateNewDetailsForAllLeg(legs:i) +getLength(distanceCounter:double) +allLegsValid(boolean) +toString(route:String)

Figure 4.13 Class Diagram of Leg Class

4.3.2.5 Bid

A class to represent a bid by a given bidder encoding the delivery parameters as presented in the second compartment in Figure 4.14. The class sets whether the bid is for forward or reverse delivery, the dates, the owner (truck) of the bid, the batch the bid is for and the proposed dates of delivery.

Bid
-bidder: Truck -batch: Batch -forward: boolean -reverse:boolean -forwardDate: int -reverseDate: int -forwardRoute: Route -reverseRoute: Route
+toString(String)

Figure 4.14 Class Diagram of Leg Class

4.3.2.6 Auction

The auction class represents an auction for a particular batch. The class holds an *ArrayList*, which keeps track of the bids currently placed. The array list holds a randomly sorted list of all the bids received.

Auction
-Batch: Batch -Bids: ArrayList <Bid>
+bid(bidsToRemove:Bid) +combinationBids(acceptableBid:boolean) +forwardBids(acceptableBid:boolean) +reverseBids(acceptableBid:boolean)

Figure 4.15 Class Diagram of Auction Class

Once a bid has been accepted, if the bidder has already made a bid for the same combination of forward and reverse delivery, then the bidder's original bid is removed and the new one is entered. The auction class will attempt to seek a combination of both forward and reverse deliveries on time from the offers, which have been received from the trucks. If this is not possible then separate forward offers from all trucks are proposed according to the parameters supplied. There are two separate intermediate delivery requests on the forward flow: *intermediateToWarehouse* specifies that the truck is seeking delivery from the DF to the warehouse; *intermediateToDestination* specifies that it is seeking delivery from warehouse to customer. Once a separate forward bid has been accepted then it will seek reverse bids from all trucks according to the parameters supplied. There are two separate intermediate delivery requests on the reverse flow: *intermediateToWarehouse*, which specifies that the truck is seeking delivery from the customer to the warehouse; *intermediateToDestination*, which specifies that it is seeking delivery from warehouse to DF.

4.3.2.7 Controller

The controller class controls the simulation. It is responsible for setting up the network, agents, locations and running the simulation. The class also keeps track of the required statistics of delivery metrics for the output files:

- numberOfBatchesOnTimeForward; the quantity of batches, which arrive to the customer's facility on time.
- numberOfBatchesLateForward; the quantity of batches, which arrive to the customer's facility late.
- numberOfDaysLateOnForward; specifies the number of days each batch arrives late to the customer.
- numberOfBatchesOnTimeReverse; the quantity of batches, which arrive to the return destination on time.
- numberOfBatchesLateReverse; the quantity of batches, which arrive to the return destination late.
- numberOfDaysLateOnReverse; specifies the number of days each batch arrives late to the return destination.

There is only ever one instance of the controller so it is called and treated as an object. Once an instance of the controller is called, it sets up the trucks and batches and starts the simulation for the number of days specified in Model Variables class. A restart method, as described in Figure 4.16, allows multiple runs to be conducted for each type of simulation run. The simulation runs with the GUI or without.

In the first step of the simulation, all the trucks are set up and given a randomly chosen origin from the set of DF, and randomly chosen volume, weight and shift capacities between the values specified in the Model Variables class. A new number of batches are created with randomly chosen origins and destinations. Quantities, volumes and weights are randomly set between the values specified in Model Variables class. Delivery dates are randomly set between the current day and the maximum planning horizon minus the time it takes customers to process batches. Return dates are set equal to the delivery date plus the number of days it takes customers to process batches. The quantity of batches is based on the number specified for the initialisation of batches in the Model Variables class and the number to be introduced each day, which also varies.
The *createNetwork* method sets the network up with the number of DF, warehouses and customers as specified in Model Variables class. A number of roads are first added randomly between locations and a quantity of roads between locations, which refers to the road complexity factor of the network, which is dependent on the value of the road probability in the Model Variables class. Afterwards, a clean-up is performed to ensure that all locations have some valid route to every other location. The network is visible through the GUI, which is made possible through the *showNetwork* method. Two critical features that occur through the controller is that every time a batch splits in half during the RTEASA, the split batches are added to the array list through the *addSplitBatch* method. If a batch fails an auction then is added into the batch list for tomorrow, which is why an *addBatchForTomorrow* method is added.

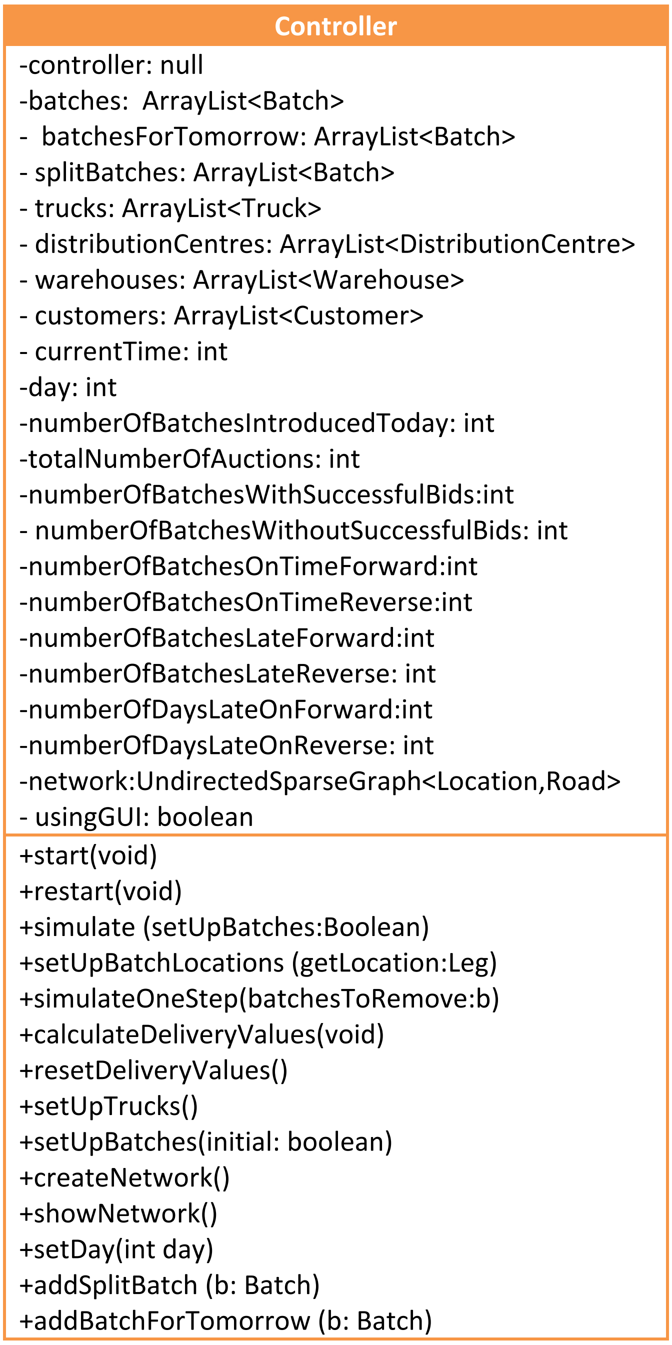


Figure 4.16 Class Diagram of Controller Class

4.3.2.8 Model Variables

The Model Variables Class holds the parameters used in the simulation, which are illustrated in Figure 4.17. This class allows the user to set the parameters of the model and set the parameters, which will be used throughout the simulation. It is also possible to create the network as required by defining the number of warehouses, customers and distribution facilities.

Model Variables
+days: int +seed: int +numDistCentres: int +numWarehouses: int +numCustomers: int +roadProbability: double +numberOfTrucks: int +minRoadTravelTime: double +maxRoadTravelTime: double +minTruckVolumeCapacity: double +maxTruckVolumeCapacity: double +minTruckWeightCapacity: double +maxTruckWeightCapacity: double +minTruckShiftLimit: double +maxTruckShiftLimit: double +daysAheadToPlan: int +numberOfBatchesAtInitialisation: int +numberOfBatchesPerDay: int +minBatchQuantity: int +maxBatchQuantity: int +minBatchWeight: double +maxBatchWeight:double +minBatchVolume: double +maxBatchVolume: double +batchProcessingDays: int

Figure 4.17 Class Diagram of Model Variables Class

4.3.3 **Summary**

Section 4.3 presents the foundations of the RFIDERTEAM. Descriptions of the packages and classes have been provided where further descriptions of the code are found in Appendix C and Appendix D. The packages illustrate how the code has been segmented and each class has been explored in detail. The attributes and methods, which have been explored describe some of the interactions between the agents and describe the environment within which they are embedded. Section 4.4, will explore the activity of the RTE delivery request and specify the importance of RFID technology to aid in RTE network improvement.

4.4 Activity Diagram: RTE Delivery Request

The activity diagram in Figure 4.18 illustrates the flow of a delivery request for a RTE and the main interaction points during the simulation. The RTE requires a delivery and is represented by a RTE Batch agent. A delivery request is initiated and an auction process is opened for the batch. If the RTE Batch agent receives no bids from the Truck agent then the batch is added to the next day's auction list. If a bid is received and then rejected, the batch is put into the next day's auction list. A bid which has been accepted would require the RTE batch's status and truck's schedule to be updated.

Once the schedules have been updated or a batch is placed onto the next day's auction list, the auction is closed and the delivery request for that batch for this simulated day terminated. For each activity, data is required in order for the activities to be completed in real-time and align with a real-world logistics network. RFID technology is a critical tool, which is utilised to provide the level of data automation required in order to allow the agents to conduct effective decisions.

4.4.1 Data Visibility

Currently, there is a lack of real-time data, which is available concerning RTE movement, making RTE management improvement challenging. It is difficult to simulate a system with poor data as it leads to losses, delivery mistakes, and poor decision-making throughout the network. In addition, there is often not a unique date for each RTE to return and ad hoc approaches are in place where you replace a full RTE with an empty RTE at each unloading, which often falls short of what is required. The RFIDERTEAM presented improves from current practices where there are often no forms of auto-identification, such as barcode to a system with RFID technology, which assigns each RTE batch with a unique ID and scheduled return date. Each RTE has its own tag and the individual RTE is aggregated into a batch and a batch is given a single unique identifier in order to make the simulation feasible.

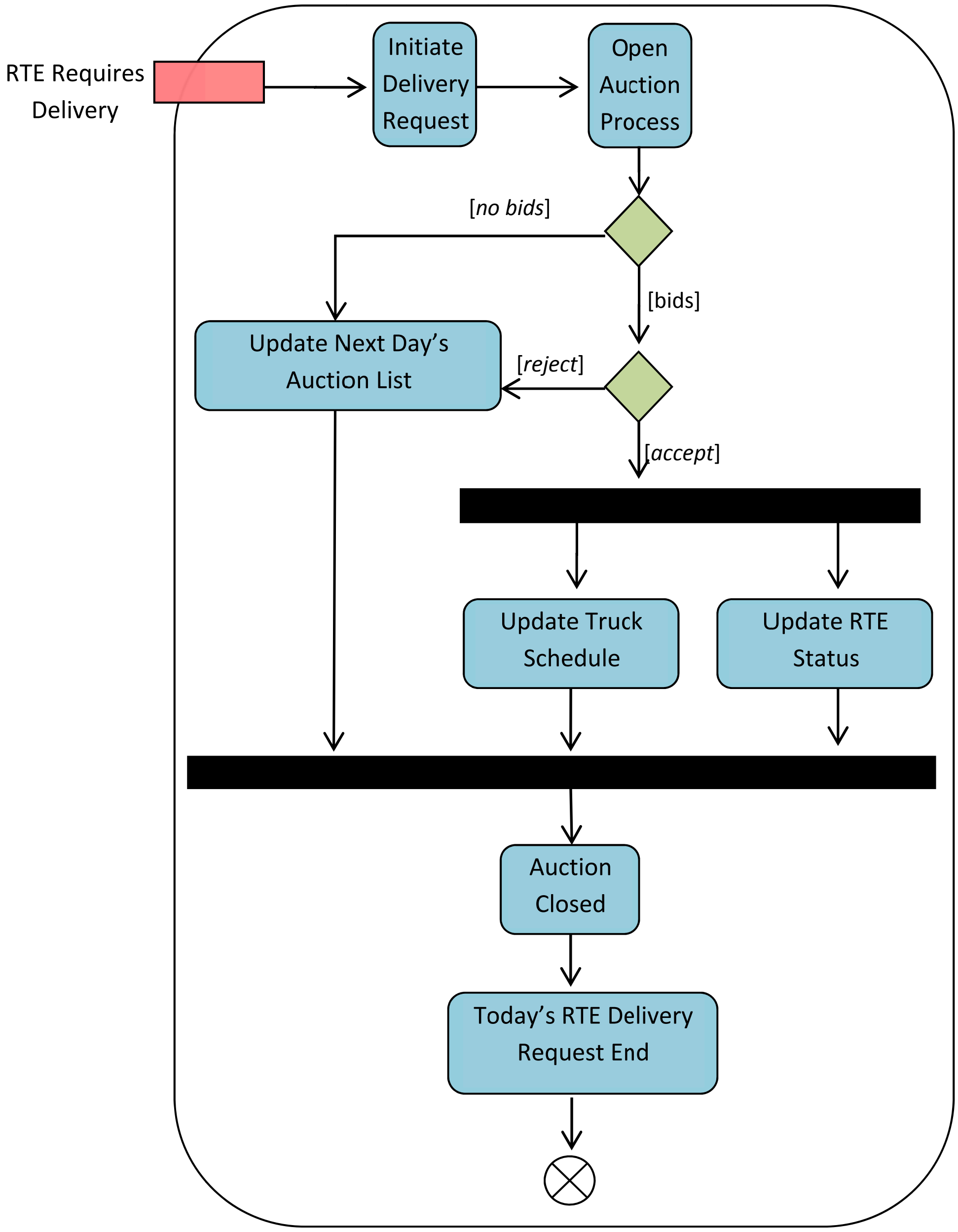


Figure 4.18 Activity Diagram of a RTE Request for Delivery

The data is required in order to increase visibility of the items and the network. This alone improves the RTE network as otherwise lost pieces are visible. However, this data is further utilised through increasing the data of the item with the agent, improving the agent's internal state, and the environment of the agent, which in this case are: the other units, truck, orders and facilities. Therefore, each component of the network requires a certain level of data in order to ensure the functionality of the agents, and therefore this thesis introduces each component's data frames.

4.4.2 **RFID enabled Data Frames**

The essential item, which will enable connectivity and interactions between the components and agents functionality is data from RFID tags. The data frames for the RTE network are described in Table 4.3 of which there are four: RTE; Trucks; Distribution Facility and Network.

4.4.2.1 RTE

Each batch of RTE is represented by an agent, and therefore requires real-time data for each parameter as specified in Table 4.3. A unique ID acts as an identifier for the batches of which the agent will be responsible to find a truck to deliver. All the parameters, which are in the Model Variables class would be located on the RFID tag so that as they change these will automatically be updated. The simulation may suggest that the RTE is moving to certain locations, however if this is not met, the RTE moves to another place, is stolen or lost, then the model will note the irregularity and send a trigger, highlighting to the user that there is a problem. Eventually the model will be developed in order to deal with any changes, which may occur in real-time and then compute, which schedule would be best.

Other parts of data such as the next maintenance check date or cleaning date are essential for further RTE network improvements and provide an extension to the current model, producing a preventive model for RTE efficiency. The data is put directly onto the RTE at the start of the cycle so that large quantities of data are not required to be stored in a centralised database, allowing the high levels of data to be more manageable. The agent model developed, fully utilises the decentralised data frames to make decisions and action automatic updates of the RTE as it moves throughout the network.

4.4.2.2 Trucks

All the parameters in the Model Variables class such as a truck’s volume and weight capacity will be represented in the RFID tag and continually updated but other data such as emissions, which is becoming more important can also be available. This adds another extension to the model, for example ensuring that the truck adheres to the emission policies set out by the government, which could possibly affect the routes developed, making further research in this area very interesting.

A GPS connection aids in referring data in regards to the status of the truck, for example if there is traffic or it has broken-down. This could have repercussions on scheduling RTE batches, and so the quicker the information is disseminated then the quicker the model can react and simulate the next best possible solution, whilst considering any negative effects such as late batches or high costs.

RTE	**Trucks**
ID Equipment Volume Capacity Equipment Weight Capacity RTE Weight Load Ability RTE Volume Load Ability Current Location Destination Forward Delivery Date Reverse Delivery Date Type of RTE Last Maintenance Check Date Next Maintenance Check Date Last Cleaning Date Next Cleaning Date Age of Unit Ownership Status ID of Goods being Delivered Special Requirement e.g. temperature, vibrations, GPS, etc.	ID Truck Volume Capacity Truck Weight Capacity Truck Volume Load Ability Truck Weight Load Ability Current Location Destination Forward Delivery Date Reverse Delivery Date Driver's Hours Route Length Mileage Average loading time Emissions Last Maintenance Check Date Next Maintenance Check Date Last Cleaning Date Next Cleaning Date GPS Connection e.g. monitor the location and miles per hour
Distribution Facility	**Network**
ID Location Maximum Throughput Current Throughput Maximum Cross-docking Rates Current Cross-docking Rates Maximum Stock Ability Current Stock Ability Average processing Time: a. Receiving b. Picking c. Shipping d. Tagging e. Cleaning	Maximum Capacity of Trucks Minimum Capacity of Trucks Maximum Capacity of RTE Minimum Capacity of RTE Nodes and Arcs of the Distribution Facilities Range and Quantity of Available RTE Range and Quantity of Available Trucks

Table 4.3 Data frames required for the RFIDERTEAM

4.4.2.3 Distribution Facilities

This data frame is essential in order to specify the current capacity at the DF, and therefore whether it is possible for a truck to stop at the location. Data from each DF is updated in order to aid with the evaluation of the cross-docking opportunities available. An extension to the model could be to have an agent represent each DF and look after its best interest, making sure that it is running efficiently and effectively. Therefore, the Truck agent would need to barter with the DF Agent in order to make a stop and unload or load at the location.

4.4.2.4 Network

The Model Variables class allows the user to define the network for example, the number of trucks, RTE and DFs. As these figures change, the parameters need updating. The data is collected from RFID enabled RTE, trucks and DFs, which is then imported into the Model Variables class so that the model automatically updates the simulation. An example of a modification is that if the level of RTE available in the network decreases from ten batches per day to five days per day, the *numberOfBatchesPerDay* parameter in the Model Variables class is modified in the simulation.

4.4.3 Benefits of RFID enabled Data Frames

There are a variety of benefits, which RFID related data frames bring to an agent-based RTE network model:

4.4.3.1 Improved Environment

The outbound RTE network is highly dynamic and variable. In order for agents to represent the environment effectively, real-time data needs to be available to the agents and activities conducted throughout the model. Real-time data of such large quantities may be hard to retrieve and process, however by decentralising the level of data and retrieving it from the various data frames, the environment is updated automatically. As the environment often feeds a lot of data into the agent, which is then utilised to conduct decisions, it is vital in producing a successful modelling and simulation tool.

4.4.3.2 Enhanced Internal State of Agent

The internal state of the agent carries out the procedures and rules and is like the "brain" of the agent determining its functionality and behaviour. The agent uses its knowledge, which could be in the form of algorithms, rules, frames, logical expressions and data. The data is an essential component, which feeds the knowledge area of the agent determining the level of functionality and behaviour of each agent.

4.4.3.3 Realistic and Real-time Data

The type of data gathered must be in real-time and accurate in order to allow the agent model to produce realistic results. In order to give the decisions, which are conducted meaning and real-world application, constant real-time data of each important component is necessary. RFID tags provide all the required data frames and each agent is fed the information, which is necessary for it to complete its tasks. It is not common to run simulations with real-time data, particularly when developing custom-built agents and algorithms; however RFID is essential to capture the network and enable real-world application.

4.4.3.4 Alignment with Real World Events

Throughout the RTE network there are various events such as changes in demand and available network capacity. A static model would not have the capability to capture such events, however the dynamism of an agent-based model allows these types of events, which are often unexpected (i.e. truck failure) to be represented. In order for these events to be captured and the advantageous capability of agent-based modelling in comparison to other modelling techniques to be displayed, data of such events must be captured in real-time and integrated into the agent model. The RFID data frames provide the appropriate data, whilst the middleware enables a transfer of information through databases, which act as an interface between the middleware and agent-based model.

4.4.3.5 Faster and Higher Quality of Decision-making

RFID data frames enable a decentralised and localised form of decision-making, which produces faster and higher quality solutions. A localisation of decision-making is useful in the RTE network, i.e., if a truck breaks-down, it would be beneficial to just reschedule the batches which are affected and leave all other deliveries and batches as scheduled.

4.4.3.6 Automation of Data Retrieval

The data specified on all the data frames is of a large quantity and is difficult to retrieve and maintain on a central database. A RFID tag enables data to be automated and put onto the tag, which is attached directly on the item. A RTE with a barcode would require scanning of each individual piece of RTE, however a RFID uses radio-frequency, and therefore batches of RTE are scanned simultaneously, which fits perfectly with any decisions that are conducted with the RTE Batch agent.

4.4.3.7 Removal of Waste

There are various forms of waste, many of which are from the processes, which are conducted throughout the RTE network. In conjunction with data automation, each process is automated or at least improved with the advantageous attributes of RFID based data frames. Each process is investigated in terms of whether it is possible to be compressed in terms of time or eliminated completely.

A RTE cycle has dual flows: reverse and forward. The agent model developed needs to consider both flows, and therefore a novel scheduling algorithm, the RTEASA, is developed and described in section 4.5. The algorithm develops the level of functionality by introducing: a bid formation, and offer acceptance and communication to enhance decision-making, which will rely on the accuracy of the RFID data.

4.5 RTE Agent Scheduling Algorithm

A novel algorithm to enable the scheduling of RTE where both the forward and reverse flows are considered has been developed. Figure 4.19 illustrates the sequence of bidding between the Truck and RTE Batch agents via an activity diagram. The activity begins at the opening of an auction for the RTE in the system. The RTE agent bids for forward and reverse delivery dates, if bids are received and accepted then the RTE is scheduled and the agent dissolves. However, a RTE Batch agent cannot have a reverse delivery until the forward delivery date has been scheduled. If the agent secures a direct forward delivery but not direct reverse delivery, then it tries to get an indirect reverse delivery. If the agent is not able to secure a direct forward delivery, then it will get an indirect forward delivery and direct reverse delivery. If none of these options are acquired, then the agent tries to get indirect deliveries for both forward and reverse. The following steps occur for the scheduling algorithm:

RTE Batch agent seeks bid for a direct forward delivery:

- If true, then the RTE agent moves to step c.
- If false, no acceptable bid found, then the RTE Batch agent carries out the following stages in order to find a truck able to carry out the delivery in the current auction process:

a. Seek forward delivery bid from distribution centre to warehouse.
   - If true, agent moves to step b.
   - If false, the agent splits the batch in half and continues the auction with half a batch seeking a bid for delivery from distribution centre to warehouse.
     - If true, move to step b.
     - If false, place the RTE batch in next day's auction.

b. Seek forward bids from warehouse to customer leg.
   - If true, agent moves to step c.

- If false, the agent splits the batch in half and continues the auction with half a batch seeking a bid for delivery from warehouse to customer.
  - If true, move to step c.
  - If false, the agent places the RTE batch in next day's auction.

c. Seek direct reverse bids.
   - If true, accept the bid and close the auction.
   - If false, no acceptable bid found, the RTE agent moves to step d.

d. Seek reverse bids from customer to warehouse.
   - If true, agent moves to step e.
   - If false, the agent splits in half and continues with half a batch seeking a bid for delivery from customer to warehouse.
     - If true, move to step e.
     - If false, the agent places the RTE batch in next day's auction.

e. Seek reverse bids from warehouse to distribution centre leg
   - If true, accept the bid and close the auction.
   - If false, the agent splits in half and continues with half a batch seeking a bid for delivery from warehouse to distribution centre.
     - If true, accept bid and close the auction.
     - If false, the agent places the RTE batch in next auction.

The legs are split and the RTE agent attempts to get the RTE delivered to as close as possible to the destination required. Therefore, the RTE agent and Truck agent will select the most "acceptable" facility which will be the one closest to the destination. When an agent splits a batch into half, the agent currently in the auction keeps half of the batch, whilst the other half of the batch is placed at the end of the current auction process where it will be formed with another agent when it is time for it to go through the auction process. The agent splits the batch in half in regards to its volume and weight. The purpose of the split is in order to receive more bids from trucks, which would come due to the smaller quantity and space. The batches will remain at the quantity and volume once split for the remainder of the simulation and it does not reconstitute itself.

An agent which is unable to secure direct deliveries or legs during the auction, will place the batch or what is remaining into the auction for the next one. The bid will be placed at the head of the queue for the next auction, which is currently being carried out daily but this could be increased if required.

When seeking a direct reverse bid step c, the delivery date will be the forward delivery date plus batch processing days. If after the batches have been split, a reverse delivery date, on time, is not received then the reverse delivery date is opened as the reverse date is more flexible than the forward delivery date. The RTE Batch agent will then accept the bid, which receives the closest delivery date to the return date assigned. If a batch with an open return date is split, then for each leg the date will revert back it its original reverse date for delivery. If no offers are received after halving the weight and volume and opening the return delivery date, then the original reverse date is assigned to the RTE and placed in the auction for next time.

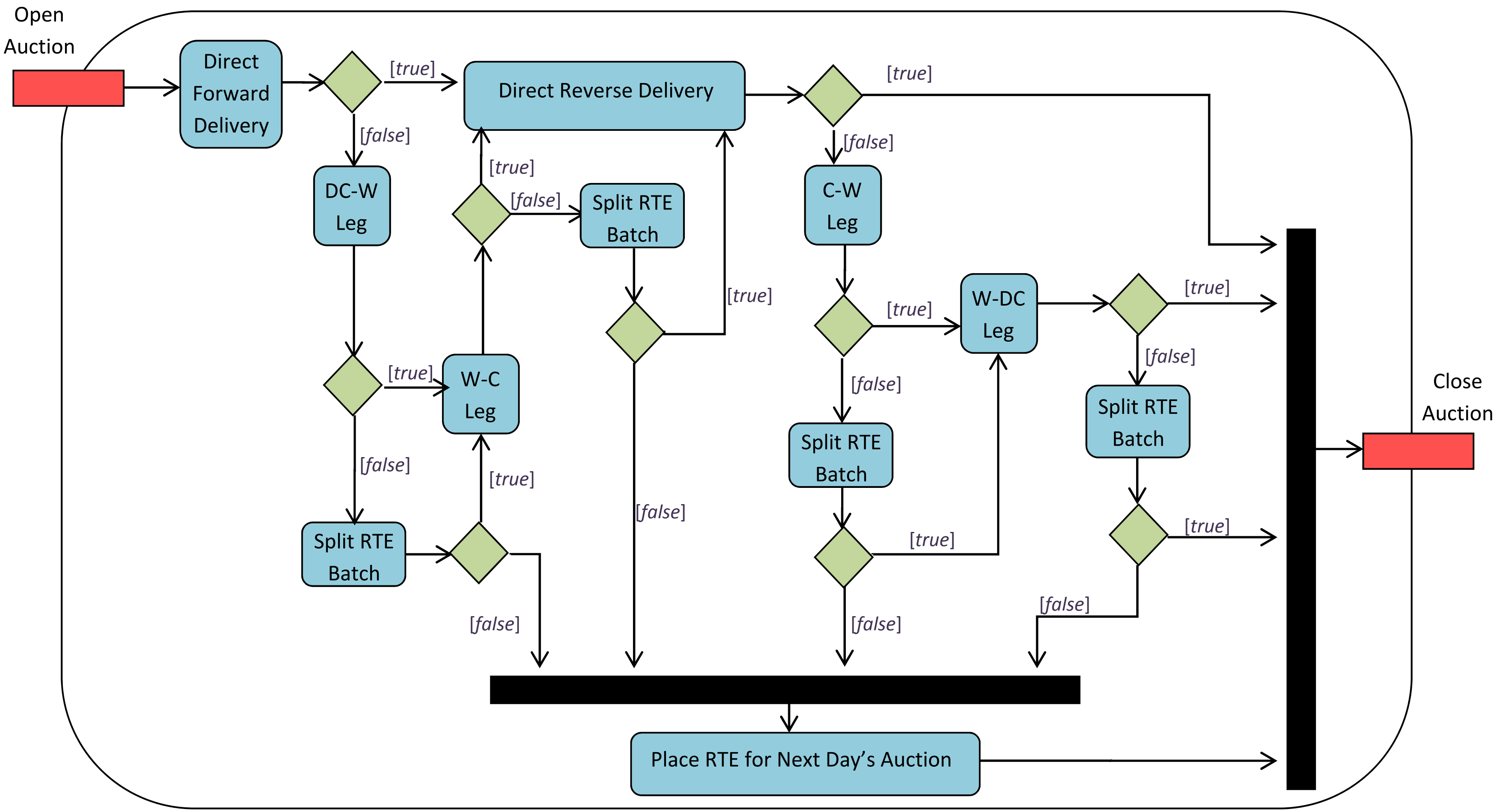


Figure 4.19 Activity Diagram of the RTE Agent Scheduling Algorithm

4.6 **Simulation Set-up**

The Controller class starts the simulation by setting up the network according to the parameters listed in the Model Variables class. Upon initialisation, it is assumed that all trucks are at the distribution facilities, have a current load of zero, and have empty routes. All trucks are assigned a randomly chosen capacity. At this point, a random number of initial Batch agents are chosen to introduce, each of which has a randomly chosen quantity, weight, destination, delivery date and return date. In order to simplify the decisions Truck agents face when bidding, Batch agents are introduced one by one.

Truck agents bid for any Batch agent, which is compatible with their planned routes and capacities: since trucks have clear or very sparse schedules at this point, it is likely that Batch agents will receive multiple identical offers. In this case, the Batch agent chooses randomly from the set of offers they have received. Once the set of initial Batch agents has been exhausted, trucks with commitments on that day leave the distribution facility and begin their routes.

At randomly chosen hours during each day, a small, randomly chosen quantity of additional Batch agents are introduced. Bidding is, as before, conducted on a one by one basis. Truck agents only bid for Batch agents whose requirements they are able to satisfy.

Once the bidding between the agents is complete, any batches required for delivery at the next day is updated and the all the truck's schedule is updated. Then the simulation runs for the specified number of days and finishes.

4.7 **Summary**

Chapter four introduces a novel approach to RTE management improvement. A RFIDERTEAM is presented where software agents are developed in order to represent the complexity in the RTE network. Each agent has autonomy as it has to make decisions based on its current decisions and accept or reject offers as it deems fit through the implementation of the RTEASA and rules that each agent possesses. Chapter 5 describes how the model meets the objectives of the RTE network. This is followed by a presentation of simulation results in chapter 6, which are performed to test whether the lean and agile aspects of a RTE network are improved with the model developed.

Chapter 5

Achieving Competitive Advantage: Autonomous RFID Enabled RTE Approach

In chapter 4, the RFID enhanced RTE Agent Model (RFIDERTEAM) was introduced with a description of the Java implementation. In order to foster competitive advantage through RTE networks, both cost and value advantage components are proposed for development, which is discussed further in this chapter. RFID tag features, such as no line of sight and readable through materials asserts the superiority of RFID technology in comparison to other auto-id technologies. This creates a stream of high quality data, which increases the visibility of RTE and trucks in the network, leading to more accurate decision-making by the agents representing the objects. The application of RFID technology to RTE and trucks combined with the formation of agents, which negotiate with each other plays a critical role in attempting to achieve the objectives, which are presented in Figure 5.1. For each objective, the data gathered from RFID tags can assist the decision-making for the agents and encourage RTE based cost and value advantage. Encouraging cost advantage through a RTE network requires both the reduction of assets but more importantly an increase in utilisation of assets in the network, as illustrated in the competitive advantage model presented in chapter 2 (see Figure 2.12), which is specifically designed for RFID enabled RTE network improvement.

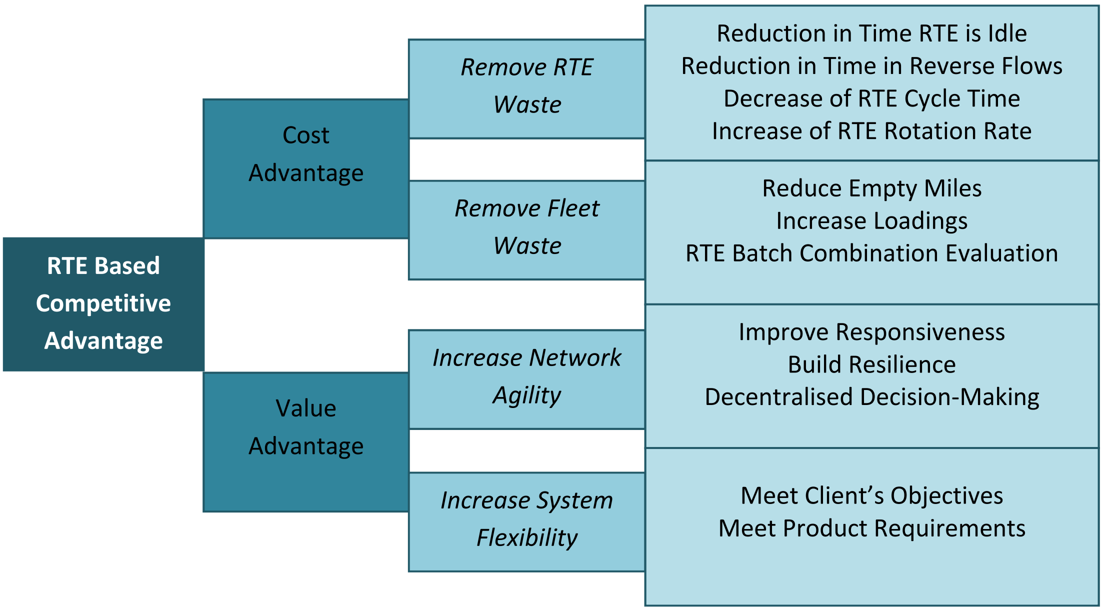


Figure 5.1 Illustration of RTE Network Competitive Advantage Objectives

Cost advantage via the RFID enabled RTE Agent Model (RFIDERTEAM) for RTE network improvement is achieved by two streams: the removal of RTE waste and the removal of fleet waste. The first stream, removal of RTE waste (which is explored in section 5.1) requires the implementation of four methods: reduction in RTE idle time; reduction of time that RTE is in reverse delivery; decrease of RTE cycle time and increase of RTE rotation rate. The second stream, removal of fleet waste (which is discussed in section 5.2) has three components for focus: reduction of empty miles, increase in the number and quantity of loadings and an evaluation of the RTE batch combination selected. Although the focus is on improving the RTE network through the achievement of objectives set in the first stream, the fleet has a significant impact on RTE network productivity, and therefore needs to be incorporated.

Value advantage is an important component of competitive advantage in the RTE network as discussed in chapter 2. In the RFIDERTEAM, achieving value advantage is approached by increasing the level of RTE network agility and increasing flexibility in the system as illustrated in Figure 5.1. Section 5.3 discusses the model's approach to increasing RTE network agility through: improving responsiveness, building resilience and offering a decentralised decision-making methodology. The methods used to enhance the RTE network's propensity to react to unexpected changes are highly dependent on the access to real-time data. To capture real-time and accurate data, RFID tags are placed on each RTE and as the equipment passes through the gates of a facility lined with RFID readers or loaded onto Smart trucks, the data is gathered and passed onto the corresponding RTE Batch agent (DHL, 2011, Neely, 2009) . Each Smart truck is equipped with a RFID technology and GPS equipment so that the loads and location data can be captured. Each truck has RFID scanning functionality as the antennae are used to monitor the RTE in the truck. During a route schedule, the system automatically updates itself to capture any changes and the data is directly transmitted to the agent based system with each Smart truck's data frame linking to a Truck agent. Depending on the updates and changes, the agent will alter its negotiation position. DHL currently utilises the Smart truck feature (Neely, 2009). The data gathered, which is real-time, and therefore accurate enables the agent to be responsive to unexpected events and fosters the creation of a more resilient RTE network.

Section 5.4 discusses the next stream in Figure 5.1 where value advantage is created by increasing system flexibility. System flexibility refers to the network's ability to reconfigure effectively to changes in supply and demand, whilst utilising the available resources efficiently. The RFIDERTEAM instils flexibility into the RTE network by adhering to product requirements and meeting the client's objectives. Information in relation to the products in the RTE and the client's requirements is available via the RFID tags attached to each object in the network, increasing the objects data frame of information, which is modified as changes occur. The chapter concludes with a summary in section 5.5.

There are three main components in a RTE network are: RTE, truck and storage facilities. The storage facilities are out of the scope of the current project and treated as black box objects in the network where flows go in and out and have no size restrictions. Presently, there is no agent representation of the storage facilities, but the model is developed with the possibility of a storage agent extension. If a Warehouse agent is introduced as an extension in the model, the trucks can be placed with active tags so that as they reach the entry and exit point of facilities, the systems controlling the entry and exit points are informed and the resources can be implemented appropriately, which aids in the reduction of queuing time, reduce congestion and facilitate better use of the staff time. The RFIDERTEAM focuses on two main components of RTE and trucks, as this requires the development of complex interactions and the implementation of multiple algorithms.

5.1 Cost Advantage: RTE Component

Cost advantage via the RFID enabled RTE Agent Model (RFIDERTEAM) for RTE network improvement is achieved by two streams: removal of RTE waste and removal of fleet waste. The first stream, removal of RTE waste, requires the implementation of four methods: reduction in RTE idle time; reduction of time in reverse delivery of RTE; decrease of RTE cycle time and increase of RTE rotation rate. RFID data is gathered at each stage of the RTE's movement and the increase in visibility and status of the RTE is crucial in enabling the model developed to initiate the ability to achieve the components. Each method is discussed in further detail.

5.1.1 Reduction in RTE Idle Time

A RTE unit which is idle, creates underutilisation in the network, which is costly and the cost is further amplified as there is a higher chance of the RTE being stolen or misplaced as it lies unused. This creates a lost opportunity of utilisation which the model developed in chapter 4 seeks to prevent. The RFIDERTEAM reduces idle time by assigning a return date to each RTE batch and scheduling the return deliveries alongside the forward deliveries, which is currently lacking in RTE network management as asserted by the case studies, which were observed at the RTE department at DHL Exel Supply Chain as described in Chapter 2. The return date is assigned via the variable, *returnDate*, which is in the Controller class and the code is presented in Appendix D-13. The return date for each batch of RTE is calculated by using the values, which have been entered via the Model Variables class. One class member, the *batchProcessingDays*, which can be modified by the user represents the amount of time it takes for a RTE batch to be unloaded, goods processed, RTE prepared for pick-up and loading. The value inserted into this variable is added to the scheduled forward delivery date, which is then assigned as the return date for delivery for each RTE batch.

The batch processing day is a value, which can be modified by the user. For example, if a batch of goods has to be delivered by the 17^{th} of June, the batch processing day could be set as 2 days resulting in the return date of 19^{th} June. This allows flexibility as conditions change and representation for RTE with longer and shorter processing times as some RTE may be available for immediate pick-up. Any changes in the scheduled times can be monitored by the RFID tags and the relevant data transferred to the RTE Batch agent in order for it to use the information to conduct accurate bidding procedures for delivery. A RFID tag can hold data reflecting to the date set for return delivery; if there are any discrepancies, the corresponding agent can make another bid in order to receive a more advantageous offer from the Truck agent.

5.1.2 **Reduction of Time RTE is in Reverse Delivery**

Each time an empty RTE makes a return journey there is a lost opportunity for utilisation. However, unless the RTE is moved to a position in the network where it can be utilised, then it will not add any further value. It is necessary to incur costs in order to move it to a position where it will be useful. A reduction in the number of days a RTE is empty can reduce waste caused by non-value adding activities. Therefore, there is an incentive to reduce the number of days a piece of RTE is empty so that the RTE is able to add value to the network by transporting goods, hence enabling the RTE to fulfil its purpose. The reverse flow should be reduced to the lowest possible time, whilst maintaining optimal logistics conditions for example, profitable routes. The quicker an empty piece of RTE can reach its destination, the more value that can be added to the network with that particular unit.

The model initiates a RTE agent scheduling algorithm (RTEASA), which contains a method to enable indirect deliveries if direct deliveries are not possible. Figure 5.2 illustrates an extract of the algorithm whereby an acceptable forward bid is found but a direct reverse delivery is unavailable in the current auction iteration.

The RTEASA initiates interaction between the agents to obtain an indirect reverse flow where the RTE batch is moved to an intermediate warehouse in the network. This is useful in the reverse flow as the RTE and fleet capacity will be utilised quicker and more efficiently within in the time constraints. This interaction is beneficial in the reverse flow due to the higher level of flexibility in the delivery dates in comparison to the forward delivery schedule. The forward deliveries must arrive to the customer on time, however in the reverse flow there is the option to move RTE around different locations and schedule deliveries, which maximise the available truck capacities, utilising the return date flexibility.

RTE Batch Class

```
….// Acceptable forward bid, however no direct reverse bid therefore, seek reverse bids to warehouse

else if (forwardBid != null && reverseBid == null) {
       Bid reverseBidToWarehouse = auction.reverseBids(trucks, false, true, false);

// Further contingencies of the algorithm ….
```

Figure 5.2 Extract of Agent Scheduling Algorithm

Once a schedule is formulated, a systems update occurs so that an assessment can be conducted as to whether the RTE should remain at its current location or to move it to its destination as originally scheduled. To facilitate a systems update, the reverse date and volume and weight dimensions need to be monitored where the implementation of RFID tags, which have a re-writable feature, enables the relevant information to be stored. The tagged RTE and trucks can be regularly updated with changes to the status, location, date requirements, weight, volume, dimensions and any other relevant information, offering real-time data frames to the agent to use in order to facilitate accurate results for the auctions conducted on behalf of the object being represented. For example, if a RTE is to move to an intermediate warehouse in a bid to reach its final destination, the tag is updated with the current location and a check is conducted as to whether there is a demand for that particular RTE. Furthermore, if there are any trucks, which are leaving to move to a nearby warehouse, an evaluation as to the benefit to move the RTE to the other facilities is conducted.

Further utilisation of a flexible reverse date of delivery can be implemented due to the transfer of the updated return date on the RFID tag. Also, the RTE have differing weights and volumes as they move through the network, particularly from a full forward load to an empty reverse load, therefore a constant update of load dimensions is required. Once the RTE agent has up-to-date information in regards to the new delivery requirements, it can negotiate accordingly. Finally, analysis of the data gathered from RFID tags can be used to monitor the length of time a RTE is in the reverse flow, and this information can be used by the agent in order to bargain a more urgent delivery action if required. This enables the RTE agent to negotiate with the truck agent using accurate values from the real-world, which are collected via the RFID enabled RTE.

5.1.3 Decrease of RTE Cycle Time

The cycle time of a RTE unit is the combination of the forward delivery to the customer and the reverse delivery back to the storage facility for re-use. The length of the cycle time is determined by the solution presented after negotiations between the Truck agent and RTE Batch agent. The RTE

Agent Scheduling Algorithm (RTEASA) algorithm begins with a request for direct forward and reverse deliveries. However, if the assigned dates cannot be scheduled, the RTE Batch agent will initialise a sequence of negotiation to secure indirect deliveries. The first leg will consist of seeking a delivery to a warehouse nearest to the destination. The RTE Batch agent picks a forward and reverse delivery schedule which will result in the shortest overall cycle time, whilst maintaining the forward delivery date as close as possible to the requested date.

RTE Batch Class

```
….// Acceptable forward bid, however direct/indirect reverse bid is unavailable
therefore, initialise contingency one

// Contingency 1: halve weight and volume
       if (reverseBidToWarehouse == null && !this.hasHalved) {
       Batch newBatch = this.createCopy(true);
       Controller.getController().addSplitBatch(newBatch);
              reverseBidToWarehouse = auction.reverseBids(trucks, false, true, false);

// Further contingencies of the RTE Agent Scheduling Algorithm ….
```

Figure 5.3 An Extract of Java Code Demonstrating Contingency One of the RTEASA

The contingency enables the RTE batch to schedule itself in an attempt to be moved instead of lying idle or being empty. The contingency provides flexibility for the RTE Batch agent to get the lowest possible overall cycle time. During the auction, the original dimensions of the RTE batches may change, so to ensure that all parties in the network are aware of the modifications, agents are updated by the attached RFID tag. RFID tags are essential in order to record and acknowledge the receipt and shipment of RTE at each stage and record the dimensions. Therefore, as RTE moves it is clear to operators that the RTE has been modified, avoiding confusion and mistakes. The ability to assign each RTE a unique ID, with a tag which has re-writable properties, is a critical feature which is required for this purpose.

5.1.4 **Increase of RTE Rotation Rate**

The rotation rate refers to the quantity of cycles each RTE batch achieves in a particular time. The aim is to increase the rotation rate in order to increase RTE utilisation rates, and therefore return on investment. An increase in the rotation rate is exploited through the flexibility in the return flows. There are three levels of contingency: indirect and intermediate deliveries, halving of the load dimensions and delivery date changes. The contingencies are used in order to encourage the RTE to flow through the network towards its destination, and therefore increase the rotation rate. If the contingencies in the previous sections fail to achieve a delivery time, the final contingency of "opening" the delivery date is implemented, which is only initialised in the reverse flow. The date is "opened" during the algorithm for the reverse flow. This action is only applied to the reverse date as the forward date is inflexible due to the requirements of on-time delivery to the customer. Once a RTE batch return date is "opened", the RTE Batch agent will attempt to get the RTE scheduled at the nearest date to the requested return delivery date. If a delivery is still not scheduled, then the batch is placed in the auction for the next day. However, the contingencies have been put into place in order to prevent this situation from arising.

At any point in the algorithm the auction may be closed and the state of deliveries could be in various stages. For example, RTE Batch agent may be split in half with two agents representing the two separate batches. At each point, it is essential that the RFID tag attached to the RTE is updated. As RTE moves through the unloading and loading processes, the tracking system is updated, and the agent representing the RTE becomes knowledgeable of its current position. When the auction restarts the next day, the agent will then have updated information in regards to the dimensions, locations and dates for delivery. This is crucial for efficient negotiations to be conducted, enabling the agent to make an informed and accurate decision.

RTE Batch Class

```
….// Acceptable forward bid, however no direct/indirect reverse bid therefore
initialise contingency three…

// Contingency 3 : open delivery date
       if (reverseBidToWarehouse == null) {
              reverseBidToWarehouse = auction.reverseBids(trucks,true,true,false);
                                                           }

// If bidding failed then the batch in placed in tomorrow’s auction….
```

Figure 5.4 An Extract of Java Code Demonstrating the Different Levels of Contingencies Utilised During Auctions

5.2 Cost Advantage: Truck Components

Although the focus is on improving the RTE network through the achievement of objectives set in the first stream, the fleet has a significant impact on RTE network productivity and must be integrated into the model, therefore the RTE Batch agents interact with Truck agents in order to achieve their delivery requests. Each truck in the system is represented by a Truck agent, which negotiates with RTE agents to develop a schedule, which considers the truck’s requirements. Cost advantage from the perspective of the Truck agent is attained through the following three methods:

1) Reduction of truck empty miles;
2) Increase of truck loads for the entire route;
3) An evaluation of a batch combination which adheres to all cost, dimension and time related constraints.

5.2.1 Reduce Truck Empty Miles

To ensure the utilisation of each truck, it is advantageous to reduce the number of empty miles per schedule. Empty miles are incurred, whilst the truck moves without carrying any RTE, which incurs a cost but adds minimum value. Therefore, the logic of the truck agent is developed in order to

Truck Class

```
….// bidding algorithm…

/**
	 * Finds the best place in a given route to add a new leg. Returns the 
new route with the shortest distance.
	 * @param route
	 * @param origin
	 * @param destination
	 * @param b
	 * @param forward
	 * @return
	 */
	public Route calculateBestRouteWithNewLeg(Route route, Location origin, 
Location destination, Batch b, boolean forward) {
		double shortestLength = getMaxShiftLength()+1;
		Route bestSoFar = null;
		for (int i = 0; i <= route.getLegs().size(); i++) {
			for (int i2 = i+1; i2 <= route.getLegs().size()+1; i2++) {
			Route candidate = new Route(route.getDay(), this);
			for (Leg l : route.getLegs()) {
			candidate.addLeg(l);
			}
			Leg pickup = new Leg(origin, b, true);
			candidate.addLegAtIndex(i, pickup);

			Leg dropoff = new Leg(destination, b, false);
			candidate.addLegAtIndex(i2, dropoff);

			if (candidate.getLegs().get(candidate.getLegs().size()-
1).getLocation() != getOrigin()) {
				Leg toOrigin = new Leg(getOrigin(), null, false);
				candidate.addLeg(toOrigin);

			}

			if (candidateLength < shortestLength && 
			candidate.allLegsValid() && candidate.getLength() < 
			getMaxShiftLength()) {
				candidate.allLegsValid();
				shortestLength = candidateLength;
				bestSoFar = candidate;
			}

			}

		}
		return bestSoFar;
		}

// getters and setters
```

Figure 5.5 Java Code Extract for the Method to Add a New Leg

encourage the addition of new legs throughout the route, carrying further loads in order to reduce the amount of empty miles. This is achieved through the initialisation of a method called calculateBestRouteWithNewLeg in the Truck class, which is illustrated in Figure 5.5. A new leg is added to the route as long as the truck values remain within certain parameters. The parameters of concern are illustrated in Figure 5.6 whereby the loadings, which are added must be within the volume and weight dimensions of the truck.

Route Class

```
// Value of legs, length of route ….

/**
	 * @return Whether, given the list of routes, whether the weight and
volume remain within
	 * the truck's weight and volume capacities at all times.
	 */

	public boolean allLegsValid() {
		calculateNewDetailsForAllLegs();

		for (Leg l : legs) {
			double r = getLength ();
			Batch b = l.getBatch();

			l.setVolumeAndWeightAtStart();
			l.setVolumeAndWeightAtEnd();
			if (l.getWeightAtStart() > getOwner().getWeightCapacity())
{
				return false;
			}
			if (l.getVolumeAtStart() > getOwner().getVolumeCapacity())
{
				return false;
			}
			if (l.getWeightAtEnd() > getOwner().getWeightCapacity()) {
				return false;
			}
			if (l.getVolumeAtEnd() > getOwner().getVolumeCapacity()) {
				return false;
			}

			if ((b != null) && (b.getDaysPostponed()== 0)){
				return false;
			}
		}
			return true;

// print route ….
```

Figure 5.6 An Extract of Java Code which Ensures All Added Legs Remain Within the Truck's Dimensions

In addition to truck dimensions, there are driver's hours regulations, which are issued by the EU and UK and are currently set at 9 hours per day (VOSA, 2011). The 9 hour restriction refers to driving time and does not include the required 45 minute break per 4.5 hours, plus the time spent waiting during loading and unloading. The maximum driver's hours is represented in the Model Variables class as the *TruckShiftLimit*, which can be set by the user. Currently, the maximum number of hours a truck's route length can be is nine hours. As scheduling the truck to loading bays at facilities is beyond the scope of this thesis, the simplest solution of 9 hours of driving time is selected for simulation. Figure 5.7 illustrates an example of how a new leg is added in order to encourage a reduction in empty truck miles.

If a truck has a shift limit of 9 hours, and the current solution is from the distribution centre to Customer A, as illustrated in Figure 5.7, then the algorithm developed will add a new leg to form better solutions as only 6 hours of the 9 hours are utilised. The new route will then be from the distribution centre to Customer A to Warehouse B and to its final destination of returning back to the distribution centre. This will result in a truck shift total of 9 hours and the search for a new leg will stop as the truck will have filled deliveries for that days' schedule. If the 9 hour truck shift limit is not reached then a search for a new leg will continue until the truck shift limit of 9 hours is reached or the truck can no longer be re-scheduled as it is required to conduct the deliveries or the truck dimensions have been reached.

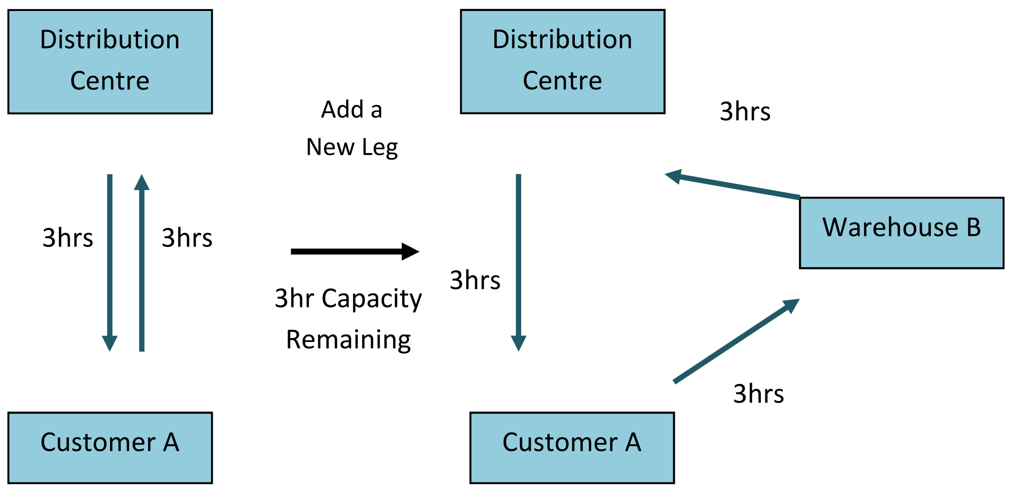


Figure 5.7 Illustration of Adding a New Leg to a Route

As the Smart truck monitors the load, the available capacity on the truck can be calculated, and if possible a new leg can be added, which will avoid the formation of schedules where trucks run empty. RFID tags can be used to monitor and hold the information in relation to the truck shift limit which can be used to determine the amount of hours there have been used and whether there is any time capacity available. In addition, tags can be used to ensure the fleet is maintained, as any breakdowns lead to inefficiencies, one of the most costly being the under-utilisation of driver's hours.

5.2.2 **Increase Loads for the Entire Route Length**

To increase the quantity and number of loadings, the schedule developed needs to be the shortest path so that further legs can be added in order to encourage a fulfilment of the truck's limits. To enable more load collections in a bid to fulfil truck capacity during the length of its route, a shortest path algorithm is implemented, which is presented in Figure 5.8. Dijkstra's Shortest Path algorithm is implemented in order to calculate and present the shortest, and therefore the cheapest path between a pair of locations. Firstly, the shortest distance between two locations is selected, followed by a calculation of the shortest route between two locations.

Another variable a Truck agent needs to consider before accepting a new load onto an existing route is the loading and unloading times at each stop. This is implemented via the code illustrated in Figure 5.9. The variable *distanceCounter* adds a value to the route. The value of the distance counter is modifiable by the user in reference to different batches or locations as required. In the example in Figure 5.9, the distance counter is set as 1.0, which represents one hour. Therefore, the Truck agent considers the loading and unloading times when deciding whether to add another batch of RTE onto the route. RFID tags can be used to monitor how long the loading and unloading action is taking, and if the actions are quicker than expected, the Truck agent can use this information to reschedule the route and possibly conduct a stop to pick up returns. If the loadings are too long, then an alert can be placed in the system in order to re-arrange deliveries.

ROUTING CLASS

```
….// set up routing class

/**
 * A class to calculate the shortest path between a pair of locations
 *
 */
public class Routing {

	public static Transformer<Road,Double> getTransformer() {
		Transformer<Road, Double> wtTransformer = new
Transformer<Road,Double>() {
			public Double transform(Road road) {
				return road.getLength();
			}
		};
		return wtTransformer;
	}

	/**
	 * Calculate the shortest distance between two locations
	 * @param start
	 * @param finish
	 * @return
	 */
	public static double getShortestPath(Location start, Location finish) {
	DecimalFormat df = new DecimalFormat("0.00");
	*DijkstraShortestPath<Location,Road> alg = new
DijkstraShortestPath<Location,Road>(Controller.getController().getNetwork(),
getTransformer());
	double distance = Double.valueOf(df.format(alg.getDistance(start,
finish)));
	return distance;
	}

	/**
	 * Calculate the shortest route between two locations
	 * @param start
	 * @param finish
	 * @return
	 */
	public static List<Road> getPath(Location start, Location finish) {
		*DijkstraShortestPath<Location,Road> alg = new
DijkstraShortestPath<Location,Road>(Controller.getController().getNetwork(),
getTransformer());

		return alg.getPath(start, finish);
		}

	}

//end of class

//Dijkstra's Shortest Path Algorithm imported
from:http://jung.sourceforge.net/doc/api/edu/uci/ics/jung/algorithms/shortestpath/Dijk
straShortestPath.html
```

Figure 5.8 Extract of Java Code which Implements Dijkstra's Shortest Path Algorithm

Route Class

```
// Model Series of Legs…

//Update all details of legs…

//set the route….

/**
	 * @return The length of this route. Each leg where loading/unloading
occurs adds 1 hour to the route, plus
	 * the time taken to travel between locations. For legs where no action
is performed, only the travel time is
	 * added.
	 */
	public double getLength() {
		double distanceCounter = 0.0;
		for (int i = 0; i < getLegs().size(); i++) {
			if (getLegs().get(i).getBatch() != null) {
			distanceCounter+=1.0;
			}
			if (i == 0 && getLegs().get(i).getLocation() !=
getOwner().getOrigin()) {
				distanceCounter+=
Routing.getShortestPath(getOwner().getOrigin(),
getLegs().get(i).getLocation());

			}
			if (i > 0){
				distanceCounter+=
Routing.getShortestPath(getLegs().get(i-1).getLocation(),
getLegs().get(i).getLocation());
			}
		}
		DecimalFormat df = new DecimalFormat("0.00");


		return Double.valueOf(df.format(distanceCounter));
		}

// valid all legs (figure  5.6) ….

//print route…
```

Figure 5.9 A Java Code Extract which Calculates the Length of the Route and Adds a Loading Time

5.2.3 Present a Batch Combination Adhering to Cost and Time Limitations

The Truck agent must select a RTE batch combination which adheres to cost, time and dimension limits. Therefore, these values need to be calculated to ensure that the truck maintains these within routes as they are being developed. In order to calculate these, a base value is used, which stems from the *RoadTravelTime*, which is set in the Model Variables class. The value of the road length between locations is set as 0.1 and 1.0. The values are modifiable to allow flexibility. These values are assigned to each leg, which when added together, form the route. The value represents the length of time and is used to calculate the cost rate of the vehicle per hour. Therefore, the truck must pick a combination of batches to deliver, whilst maintaining the maximum truck shift length and truck dimension boundary. The value is then used and converted into a cost by multiplying the value of the length which represents hours, by the truck operating cost per hour, which is discussed further in chapter 6. The agent requires numerous forms of data each time a change occurs in relation to the state of the RTE, therefore RFID tags can be used to hold the data and each change can be transmitted to the corresponding agent enabling bids and decisions to be conducted with real-time status updates.

5.3 Value Advantage: Increasing Network Agility

The RFIDERTEAM implements agility in the RTE network by increasing responsiveness, creating resilience and utilising the functionality of decentralised decision-making. Firstly, the RTE network needs to be able to respond to events, particularly unexpected events, in an efficient manner. Secondly, the model aims to foster resilience in the RTE by removing vulnerabilities such as losses and break-downs. Finally, the decision-making is localised so that the agent representing the object can make efficient and effective decisions in a reasonable time-frame without having to include the whole network. In this section, each method is explored further. The network agility is represented by monitoring the reduction of the overall operating cost, which will be demonstrated in chapter 6.

5.3.1 Improve RTE Network Responsiveness

Currently RTE networks as asserted by the case study with DHL Exel Supply Chain in chapter 2, struggle to cope with expected (standard) events in the network. The management of RTE networks become even more challenging as they become more dynamic and unexpected events are introduced. There are a variety of factors, which can affect the supply and demand of RTE such as missing items, pilferage, higher customer demand, fleet disturbances, broken RTE and dirty RTE. The RFIDERTEAM addresses these problems by utilising RFID technology to acknowledge any disturbances or fluctuations in the real-world. This is followed by acknowledgement by the agents of the change, which enhances the decision-making.

Once a change has been acknowledged then the second feature, decentralised decision-making is used by the agent in order to create a reaction through an effective decision making process to enable agility. For the model to conduct decentralised decisions, a responsive network is necessary through the development of an information-centric environment, which is enabled by RFID technology as it transmits information from the objects and the information is then used by agents in the virtual world.

The responsiveness can be assessed by the ability to record changes and the time taken to acknowledge an abnormal occurrence in the network, for example a delayed delivery and the time taken to resolve the issue. In addition, it includes the quality of the response, which can be measured by the rate of completion of the action or the extra cost or time taken to complete the action. The responsiveness is implemented in the model by the RTE Batch agent and the Truck agent where a decentralised decision-making approach has been adopted.

5.3.2 Adopt Decentralised Decision-making

A decentralised decision-making approach is implemented in order to encourage better decision-making as discussed in chapters 2 and 3. A decentralised representation of RTE networks is essential as the network is multi-layered and multi-dimensional; a central mode of decision-making requires many assumptions, which do not allow sound representation. By representing the RTE network as it is, instead of via a central unit, a more realistic model can be developed, which offers better representation and functionality of the distributed RTE network.

The RTE Batch agent and Truck agent is representing objects in the network and is given control to make decisions in the best interest of its corresponding object. Therefore, no matter how dynamic the environment is or whether the object's goals are conflicting, the agents can negotiate with each other in order to make the most effective decision.

The purpose of disseminating control throughout the network through objects and creating decentralised decision-making is to improve the quality and quantity of decision-making. By localising the problem, other parts of the network are not involved in the decision, which has no impact on them, saving time and resources. Therefore, the agents are representing the objects in the real-world and are only interacting with the parts of the network which are required, hence localising the problem when formulating a solution. This offers a better representation of the real-world rather than conducting every decision through a central unit and across large networks.

Conducting the decision-making on a localised level, the data is assigned to each object in the network and the agent can access this information and use it to conduct a solution on a local level. The agents need to acquire real-time data in regards to the status and requirements of the objects (RTE and Trucks), which are being represented, and therefore the RFID technology's ability to offer high quality data on an object-level is pivotal. The information gathered by the data is imperative for the agents in the model in order to provide knowledge and enable the agents to make informed decisions. Therefore, as product lead-times, inventory and fleet parameters change, this can be read by the tags and the information transferred to the agents.

5.3.3 Creating Resilience

Once a responsive network with the ability to make decisions on a local level has been created, it is important to remove as much risk as possible and create resilience in the RTE network. There are a number of vulnerabilities in the RTE network, both internal and external. The biggest risk is the under-utilisation of RTE because it then has a higher chance of being lost, stolen, dirty or broken.

RFID ensures that a tracking feature is available for each RTE and the model utilises this further by initialising preventive measures such as maintenance. RFID tags can be used to hold the information of each RTE and the model can use this information when initialising the scheduling algorithm. If preventative measures such as maintenance of trucks and cleaning of RTE are implemented and if there are unexpected events, then the network is in a stronger position to respond as it has decision-making units and real-time data for each object. Various measures can be implemented in order to create the resilience: scheduling in maintenance checks for the RTE; RTE checking; vehicle maintenance checks and analysis of KPIs.

5.4 Value Advantage: Increasing System Flexibility

An agile system can respond to changes; however a network also requires flexibility, which refers to how the network responds. By utilising the resources of the network such as the fleet and the information systems, the flexibility of the system can be incorporated. In the RFIDERTEAM, the flexibility achievement is complex as it is multi-layered, with a variety of products and RTE and numerous clients. Therefore, a flexible RTE network reconfigures efficiently by returning the RTE as soon as possible back to the point of use, whilst adhering to the product’s requirement and client’s objectives.

5.4.1 Adhering to Product Requirements

In addition to having many different types of RTE in one network, there can be multiple different levels of products. Each product may require a different type of RTE to transport it and certain conditions unique to that product, for example frozen, chilled, and fragile. Therefore, the model developed enables this level of flexibility as each agent can have a certain set of attributes and requirements for its RTE and the product it is carrying. The data can be transferred via RFID tags, which will give the agent a unique set of data to use as information when negotiating a schedule.

5.4.2 Meet Multiple Clients' Objectives

Meeting client's objectives, a logistics service provider (LSP) would have multiple clients where each one may have different objectives for their supply chain. If a client has a certain objective, then this can be quantified. For example lead-time reduction will result in changes to the date and this is transferred to each RTE, therefore each RTE can be moved with the appropriate treatment. The objectives can either be put in manually, for example fleet reduction, which will result in fewer trucks in the model, or via the RFID tags for objectives such as modifications to delivery dates.

5.5 Summary

This chapter seeks to highlight the functionality of the model presented in chapter four in order to achieve competitive advantage for RTE management. The chapter describes the objectives of the model and focuses on cost and value advantage achievement through the RFID Enhanced RTE Agent Model (RFIDERTEAM). The critical nature of RFID technology application is specified in terms of creating the vital level of real-time visibility to facilitate agent interaction enabling accurate bidding and decision-making. RFID tags are needed in order to provide the agents and the environment of the agent based simulation with a constant stream of updated information of any changes throughout the scheduling of RTE. RFID tags can hold various data frames of information such as dimensions, available capacities, dates changes, and requirements. The combination of agility and flexibility in the RTE network creates a model, which is more realistic to the dynamic environment of distribution, and therefore more applicable to the real-world. Further evaluation of the significance of this model is conducted in chapter 6 through the simulation results of a number of scenarios, which is followed by an explorative and extended discussion in chapter 7.

Chapter 6

Simulation Results

Chapter six presents the simulation results of various scenarios, which illustrate the different aspects of the RFIDERTEAM developed. Unfortunately due to unforeseen circumstances, real-world data from project partners DHL was unavailable; therefore the source of input data is based on standard issue sizes and regulations as presented in Figure 6.3. For each scenario, during the simulation, data is generated based on pre-programmed parameters or through random numbers to offer an array of different scenario types to be tested. A simulation requires a source of random numbers in order to generate outcomes to analyse. Random numbers are used to generate the same set of loads so that a comparison can be made when different scenarios are conducted.

Section 6.1 describes the initial set-up of the scenarios where the measures used during the simulation are described in Table 6.1 and the variables in Table 6.2. This is followed by a presentation of cost-based calculations for the late batch cost and truck operating cost in section 6.2. A comparison of the late batch costs from the model to real world figures from (Ilic et al., 2009) is presented in section 6.3. Section 6.4 investigates the impact of increasing the amount of time a batch is being processed, which means the amount of time a RTE spends at a customer or idle during the forward flow against the fleet cost. Section 6.5 investigates the impact of increasing the network complexity whereas section 6.6 explores the effect of implementing a larger network. Section 6.7 presents simulation results of smaller RTE batch sizes, whilst section 6.8 demonstrates the impact of a bigger range of RTE. Section 6.9 tests a scenario with a high level of complexity and the chapter concludes with a summary in section 6.10.

Measures
Late Batch Costs
Amount of days batches are late, which are multiplied by a cost per day of not using it.
Truck Operating Costs
The amount of hours a truck is in use, which is multiplied by a cost of operating the truck per hour.
Total Cost
The total cost is a summation of the late batch cost and the truck operating cost.
Volume Utilisation of a Truck
The volume utilisation rate represents the loadings of a truck for a route, each day.
Weight Utilisation of a Truck
The weight utilisation rate of the truck refers to the average weight of the loads in the truck through a route, each day.
Time Utilisation of a Truck
The time utilisation rate refers to the percentage of time the truck spends carrying loads during each route.
Number of Auctions
The number of auctions conducted between the RTE Batch agent and Truck agent in order to schedule the RTE.

Table 6.1 List of the Measures Used During Simulations

Variables
Batch Processing Days
The amount of days a RTE is being processed and awaiting pick-up, which also determines the return delivery date.
Number of Trucks
The number of trucks running through the network is modifiable.
Size of Network (Number of Facilities)
The network is built of distribution facilities, which the trucks leave and return to each day and warehouses and client's facilities.
Network Complexity Factor
The network complexity can be modified to include more or less roads between facilities.
Number of RTE in Each Batch
The amount of RTE in each batch can be modified and set between a lower and upper limit.
Dimensions of Each RTE Batch
The dimensions of each RTE Batch can be modified, which refers to the weight and volume.
Quantity of RTE Batches
The quantity of RTE batches which are placed in the system at the start of the simulation and the amount of RTE batches introduced into the system, each day.

Table 6.2 List of Variables Used During Simulations

6.1 Model Validation: Scenario Set-up

Scenarios are used to test the impact of the algorithms developed and the interactions of the agents. The rules devised in the model will be tested through the simulation of various scenarios. This chapter will present results of an array of simulations, which are set-up through the graphical user interface (GUI) as illustrated in Figure 6.1 and implemented via eclipse.

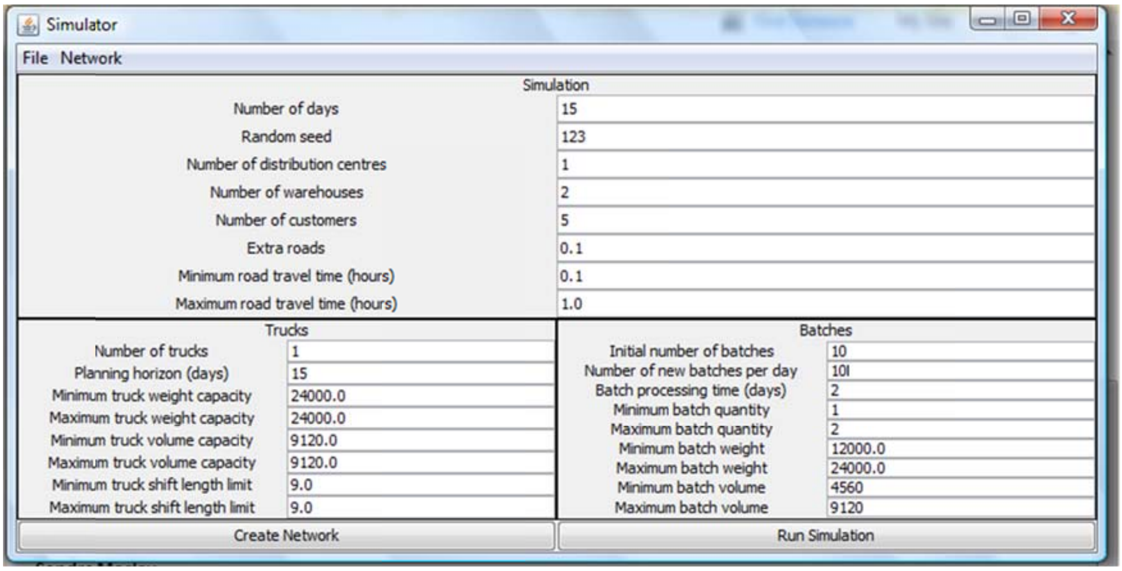


Figure 6.1 GUI to Set-up the Simulation for a Scenario

The GUI accepts parameters of a scenario via three compartments: simulation, trucks and batches as demonstrated in Figure 6.1. A simulation is set up with the following properties:

- Each simulation will be run for 15 simulated days, which is chosen to enable a fully working network for evaluation. The length of the simulation allows multiple simulations to be conducted within the time-frame and enable the data to be manageable for analysis.
- A fixed random seed function is integrated into the model in order to allow multiple scenarios to be conducted with the same set of random numbers to allow for comparison and testing.
- Figure 6.2 illustrates 8 locations, which are set-up in the network: 1 distribution centre, 2 warehouses and 5 customer facilities. The initial network is small in order to enable many simulations to be run quickly as the larger the network, the longer each simulation run will take. Throughout this chapter different scenarios of various network sizes will be tested.
- The number of roads between the facilities is determined by the complexity factor, which is between 0 or 1. The current value is 0.1 as the higher the complexity the longer the

simulation. The impact of the road complexity factor is investigated in section 6.5. Each road is assigned a random value between 0.1 and 1.0, which represents its length.

The network created is viewed via a view network tool on the GUI, which is illustrated in Figure 6.2.

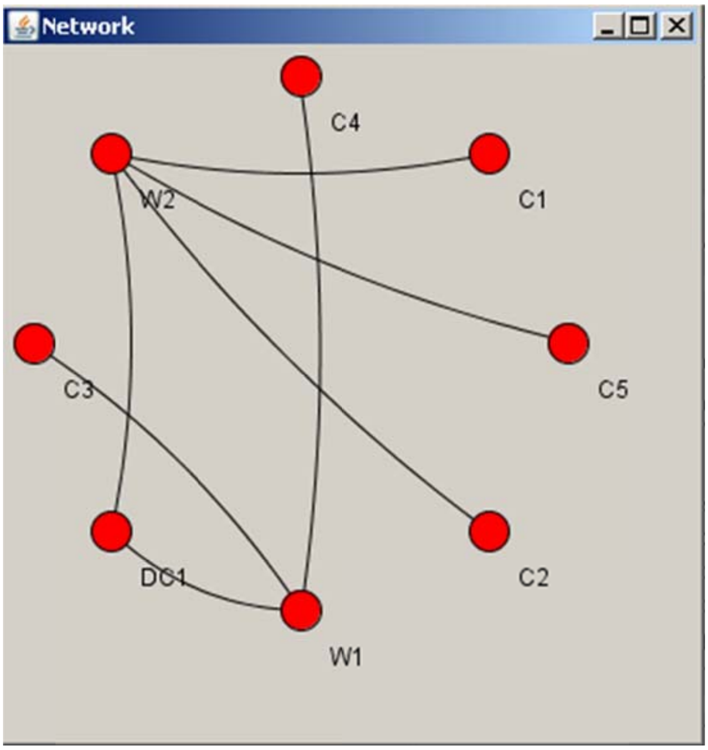


Figure 6.2 GUI based View of Network Set-up

In the second compartment of the GUI, Trucks, the quantity and size of the trucks is modified.

- The number of trucks in the scenario is modified as and when required for each simulation run depending on testing requirements.
- The planning horizon for the truck, which represents the number of days in advance an agent plans its schedule is set at 15 days so that the truck begins to set-up loads for future deliveries.
- For the current scenario, each truck is sized at the dimensions according to DHL as illustrated in Figure 6.3. The weight is set at 24,000 kilograms and as each truck is the same size, the minimum and maximum values are the same.

- The available volume capacity of the truck is 91.20 cubic metres again for both the maximum and minimum values.
- The maximum and minimum truck shift length refers to the amount of hours a truck is operating, which is dependent on the driver’s hours. In accordance with UK and EU regulations, the maximum daily driving limit is 9 hours, therefore the truck shift limit is set at 9.0 (VOSA, 2011).

The third compartment of the GUI represents the batches:

- An initial number of 10 RTE batches are set-up at the start of the simulation.
- Each day more batches are added, which is set at 10 batches per day. An increase in batches is explored in sections 6.4 to 6.9.
- The batch processing time determines the return date and is currently set at the simplest solution of 2 days, therefore this means that as soon as the goods have been delivered the RTE should be returned in 2 days. Any days after this return date and the batch is delivered late. An increase in batch processing days is increased in sections 6.4 to 6.8.

Each batch is made up of a variety of RTE. Based on the dimensions in Figure 6.3 a full truck load would consist of 34 pallets, which would total 24,000 kg and 91.20 cubic metres (cbm). The lowest boundary is set at half truck loads, which are 12,000 kg and 45.60 cubic metres as illustrated in Figure 6.4.

Standard vehicle dimensions and weight

Standard transport equipment

Trailer (standard)	Outside dimension in metres Length	Width	Height	Inside dimension in metres Length	Width	Height	HeightPortal	Maximum weight capacity per unit in kg	Nos of Pallet spaces per unit	Maximum cubage per unit in cbm
Semitrailer Tarpaulin	13,67	2,55	2,70	13,62	2,48	2,70	2,70	24.000	34	91,20
Tautliner	13,67	2,55	2,70	13,62	2,48	2,70	2,63	24.000	34	91,20

Measurement / Weight ratios

Unit	Minimum chargeable weight
1 cbm	333 kg
1 ldm	1.650 kg

Individual requirements which exceed standard equipment can be fulfilled upon request.

For further information on DHL Freight service availability, visit your local DHL website. All services are subject to terms and conditions.
freightsales.europe@dhl.com

Figure 6.3 DHL Trailer Dimensions

Figure 6.4 illustrates how the RTE are batched and the sizes. The RTE batches start the simulation in the sizes illustrated, however as the RTE batches are split multiple times during the auction process, and therefore the size of the RTE batches will change each time a batch splits. A new agent is formed for the additional RTE batch.

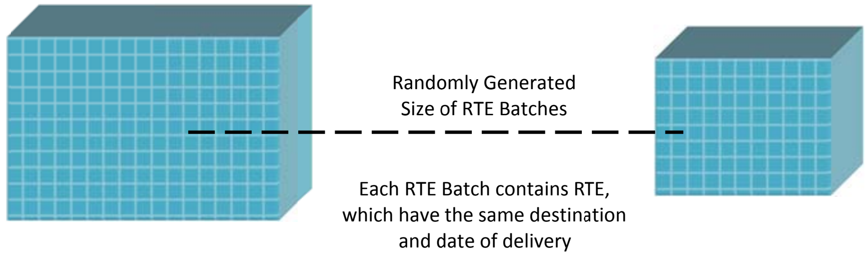


Figure 6.4 RTE Batch Dimensions

6.2 Cost Based Calculations

Interaction between the RTE Batch agent and Truck agent is based on the aim to return RTE as soon as possible for re-use, increasing RTE utilisation, whilst accounting for fleet operating costs and utilisation rates. Two types of costs are calculated after each simulation: late RTE batch costs and truck operating costs. This allows a comparison between the amount of days a RTE batch is not being utilised versus the cost of running a truck to move the RTE. The calculation of each cost is described further in this section.

6.2.1 Cost of Late RTE Batches

A RTE unit, which is late in either the forward or return delivery flow, incurs a cost of under-utilisation for each day it is not in use. The model presented reduces this by assigning a return date and encouraging on-time delivery of forward units through an algorithm, which enables agent interaction. Any batches, which are received late on the forward flow are captured by the parameter *DaysLateOnDelivery*, and any batches which are received late on the reverse flow is captured by the parameter *DaysLateOnReturn*. The results for these parameters are presented in an Excel file and an example of an extract is illustrated in Table 6.3.

Day of Simulation	Total Number of Auctions	Number of Batches Delivered On Time (Forward)	Number of Batches Delivered On Time (Reverse)	Number of Batches Delivered Late (Forward)	Number of Batches Delivered Late (Reverse)	Total Days Late on Delivered Items	Total Days Late on Returned Items
1	118	4	0	0	0	0	0
2	135	4	1	1	0	1	0
3	152	1	4	0	0	0	0
4	166	1	5	0	0	0	0
5	180	3	1	0	0	0	0
6	195	2	1	1	0	1	0
7	210	2	3	0	0	0	0
8	227	2	3	0	0	0	0
9	243	3	2	0	0	0	0
10	260	2	2	0	0	0	0
11	275	1	3	1	0	1	0
12	289	2	2	0	0	0	0
13	305	3	2	0	0	0	0
14	323	2	2	0	0	0	0
15	339	2	3	0	0	0	0

Table 6.3 Extract for KPI Excel File of Model Output

Once the number of days that each batch is late has been calculated, then it is multiplied by a cost per batch per day to assess the impact of under-utilisation. This is the cost of not having the batch in the system and is modifiable by the user. This allows flexibility as certain RTE may more valuable than others. For the scenarios in this chapter, in order to allow comparison, the cost is calculated at £80 per RTE batch per day. This is to account for the under-utilisation of up to 34 pallets for each RTE batch. The cost calculation represents the cost of having to hire or buy up to 34 pallets and this is given the same weighting as an hour's operating cost for trucks, which is set at £80 per hour and described further below.

6.2.2 **Truck Operating Costs**

The truck route is formulated during the simulation runs and is presented in an Excel file as illustrated by Figure 6.5.

Day	Truck Name	Truck Route
1	T1	DC1 Loading B0(6) // C2 Unloading B0(6) // DC1 No Action // Length: 3.8
1	T2	None
2	T1	DC1 Loading B1(1) // C4 Unloading B1(1) // C1 Loading B0(4) // DC1 Unloading B0(4) // Length: 7.08
2	T2	DC1 Loading B1(5) halved original // DC1 Loading B1(3) // W2 Unloading B1(5) halved original // C4 Unloading B1(3) // C4 Loading B0(0) // DC1 Unloading B0(0) // Length: 8.54

Figure 6.5 Extract of Truck Schedule from the RFIDERTEAM Output

The truck schedule lists the day of the simulation, the name of the truck and the route for each truck. In addition, the length of each route is specified, which remains within the truck shift limit as set by the user, representing the driver's hours as per regulations. Each integer refers to 1 hour of truck usage. This is then used as a basis to calculate the hourly rate.

Truck Name	Route Length	Operating Cost/Hour	Route Cost
T1	3.8+7.08 = 10.88	£80	£870.40
T2	8.54	£100	£854

Table 6.4 Route Cost Calculation

The cost calculations for the extract in Figure 6.5 are presented in Table 6.4. Trucks T1 and T2 have different operating costs per hour, which is due to size or truck functionality, for instance, refrigerated vehicles. The total operating truck costs for the days of the simulation, in this case, 2 days, are £870.40 for T1 and £854 for T2.

The truck operating costs in the simulations conducted in this chapter are set at £80 an hour in order to allow comparison of the working of the model and comparison to the under-utilisation of RTE batches. The components of the £80 per hour are made up of:

- Driver
- Fuel
- Cost of Tractor
- Cost of Trailer
- Insurance
- Tyres
- Planning Costs
- Overhead Costs

When an LSP offers rates to clients, a profit margin needs to be added but for the simulation conducted the base costs of operating each truck is required as this is what the cost would be to the LSP.

6.3 Model Validation: Late Batch Cost Comparison

The parameters of a simple network are described in Figure 6.6 and the network is illustrated in Figure 6.7. The network is formed in order to test the impact of the algorithm on late batch costs. The quantity of batches is increased to test the impact of higher numbers of RTE with and without the model.

The number of batches is increased by 10 per day up until 60 batches as illustrated in Figure 6.6. This is in order to obtain results in regards to the number of late batches as the quantity of batches in the system is increased and compared to a RTE system with slow returns and losses.

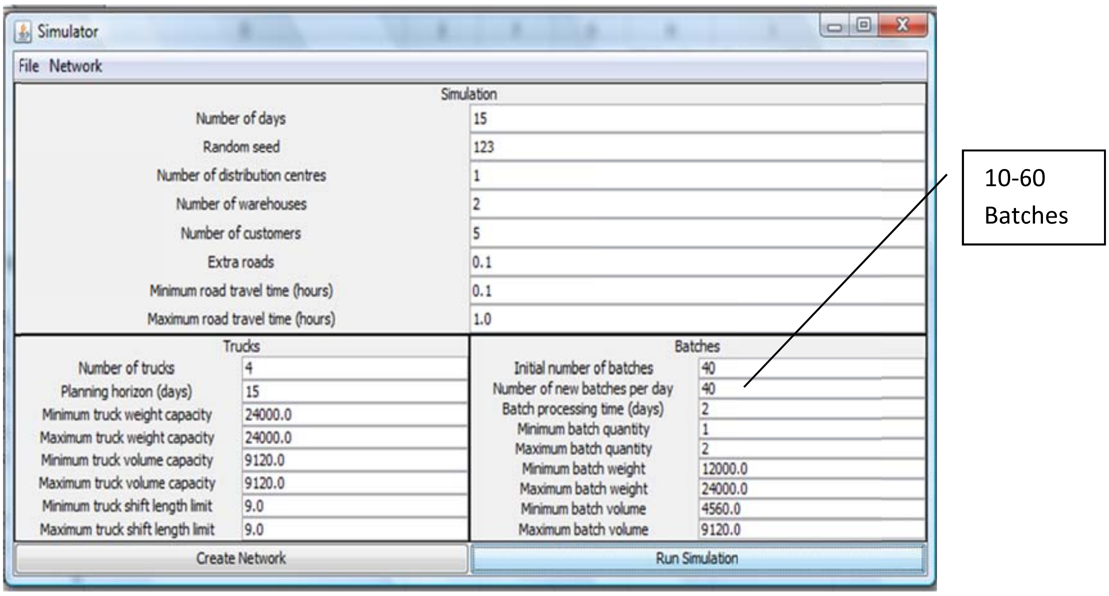


Figure 6.6 Model and Simulation Set-up for Scenario of Late Batch Cost Comparison

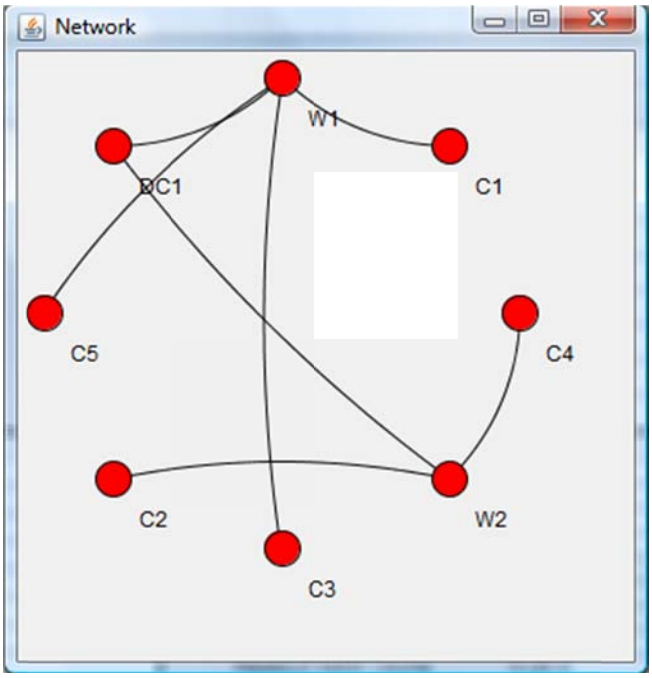


Figure 6.7 Network View of Scenario

6.3.1 **Slow RTE Returns**

To represent different variations of returns within the network, Table 6.5 demonstrates the calculations of 2 different systems of late RTE batches without the RTEASA. For 50% RTE Return within 2 days, half of the RTE Batches are returned within 2 days, whilst the others arrive at staggered times during the rest of the days of the simulation (15 days). For no return within 2 days, the RTE batches arrive at various days within the 15 days. These calculations are used for comparison to the results obtained via the simulations.

	50% RTE Return within 2 days	No RTE Return Within 2 Days
50%	Returned within 2 days	4 days
25%	RTE take a further 4 days	7 Days
25%	RTE take a further 7 days	10 Days

Table 6.5 Network View of Scenario

Table 6.6 contains the calculations for 50% of RTE, which are returned within 2 days. As the batches quantities are increased the numbers rise. Calculations are also conducted for batches, which are not returned within 2 days in Table 6.7. The number of late days are summarised in

Table *6.8* with the values gained from the agent based model and simulation. Each day is multiplied by £80 and the costings are presented in Table 6.9. The line graph in Figure 6.8 illustrates that the model returns batches on time (within 2 days), which is important in reducing the cost of under-utilisation of RTE batches. Without the model, the costs of slow returns rise in a linear fashion as the quantity of batches increase, and therefore the more batches in the network the more costly the impact is in not implementing a RTE return strategy.

6.3.2 **RTE Losses**

(Ilic et al., 2009) reported that the average loss rate for the pallets is around 9–10% annually. Besides losses, the RTI industry faces the problem that on average 9% of the assets are subject to breakage every year. A case study observed by (Ilic et al., 2009) identified that a company was losing 50% of RTE, whilst also having to cope with 30% of RTE being damaged. Each of the observed loss rates are used for comparison as illustrated in Table 6.10, which demonstrates the number of batches lost in the studies presented in the literature. Table 6.11 presents the number of batches which are lost at a rate of 10%. Table 6.12 presents the calculations at a rate of 20% and Table 6.13 at 80% rate of losses. The number of days batches are late are summarised in Table 6.14 with the result from the agent based model and simulation. Table 6.15 presents the costs for each batch rate, which is illustrated by a line graph in Figure 6.9. The graph illustrates that the implementation of the algorithm has a high impact on late RTE batch cost reduction in comparison to RTE network with losses but this is further amplified when including RTE, which is damaged in addition to losses. Further exploration into how the model reduces late RTE batch costs is explored further by testing different scenarios in sections 6.5 to 6.9. However, first simulations comparing the impact of the RTE batch processing day to the fleet costs is conducted.

50% RTE Return within 2 Days					
		Batches	Days Late	Days Late/Batch	Total Days Late
10 Batches	50%	5	0	0	
	25%	2.5	4	150	
	25%	2.5	7	262.5	**412.5**
20 Batches	50%	10	0	0	
	25%	5	4	300	
	25%	5	7	525	**825**
30 Batches	50%	15	0	0	
	25%	7.5	4	450	
	25%	7.5	7	787.5	**1237.5**
40 Batches	50%	20	0	0	
	25%	10	4	600	
	25%	10	7	1050	**1650**
50 Batches	50%	25	0	0	
	25%	12.5	4	750	
	25%	12.5	7	1312.5	**2062.5**
60 Batches	50%	30	0	0	
	25%	15	4	900	
	25%	15	7	1575	**2475**

Table 6.6 Calculation for 50% RTE Return within 2 Days

No RTE Return within 2 Days					
		Batches	Days Late	Days Late	Total Days Late
10 Batches	50%	5	4	300	
	25%	2.5	7	262.5	
	25%	2.5	10	375	**937.5**
20 Batches	50%	10	4	600	
	25%	5	7	525	
	25%	5	10	750	**1875**
30 Batches	50%	15	4	900	
	25%	7.5	7	787.5	
	25%	7.5	10	1125	**2812.5**
40 Batches	50%	20	4	1200	
	25%	10	7	1050	
	25%	10	10	1500	**3750**
50 Batches	50%	25	4	1500	
	25%	12.5	7	1312.5	
	25%	12.5	10	1875	**4687.5**
60 Batches	50%	30	4	1800	
	25%	15	7	1575	
	25%	15	10	2250	**5625**

Table 6.7 Calculation for No Return within 2 Days

	Agent Simulation: Number of Days Late	50% RTE Return Within 2 Days	No RTE Return Within 2 Days
10 Batches	1	412.5	937.5
20 Batches	27	825	1875
30 Batches	16	1237.5	2812.5
40 Batches	8	1650	3750
50 Batches	17	2062.5	4687.5
60 Batches	11	2475	5625

Table 6.8 Number of Days a RTE Batch is Late

	Agent Simulation: Number of Days Late	50% RTE Return Within 2 Days	No RTE Return Within 2 Days
10 Batches	£80	£33000	£75000
20Batches	£2160	£66000	£150000
30 Batches	£1280	£99000	£225000
40 Batches	£640	£132000	£300000
50 Batches	£1360	£165000	£375000
60 Batches	£880	£198000	£450000

Table 6.9 Costing for Late RTE Batches

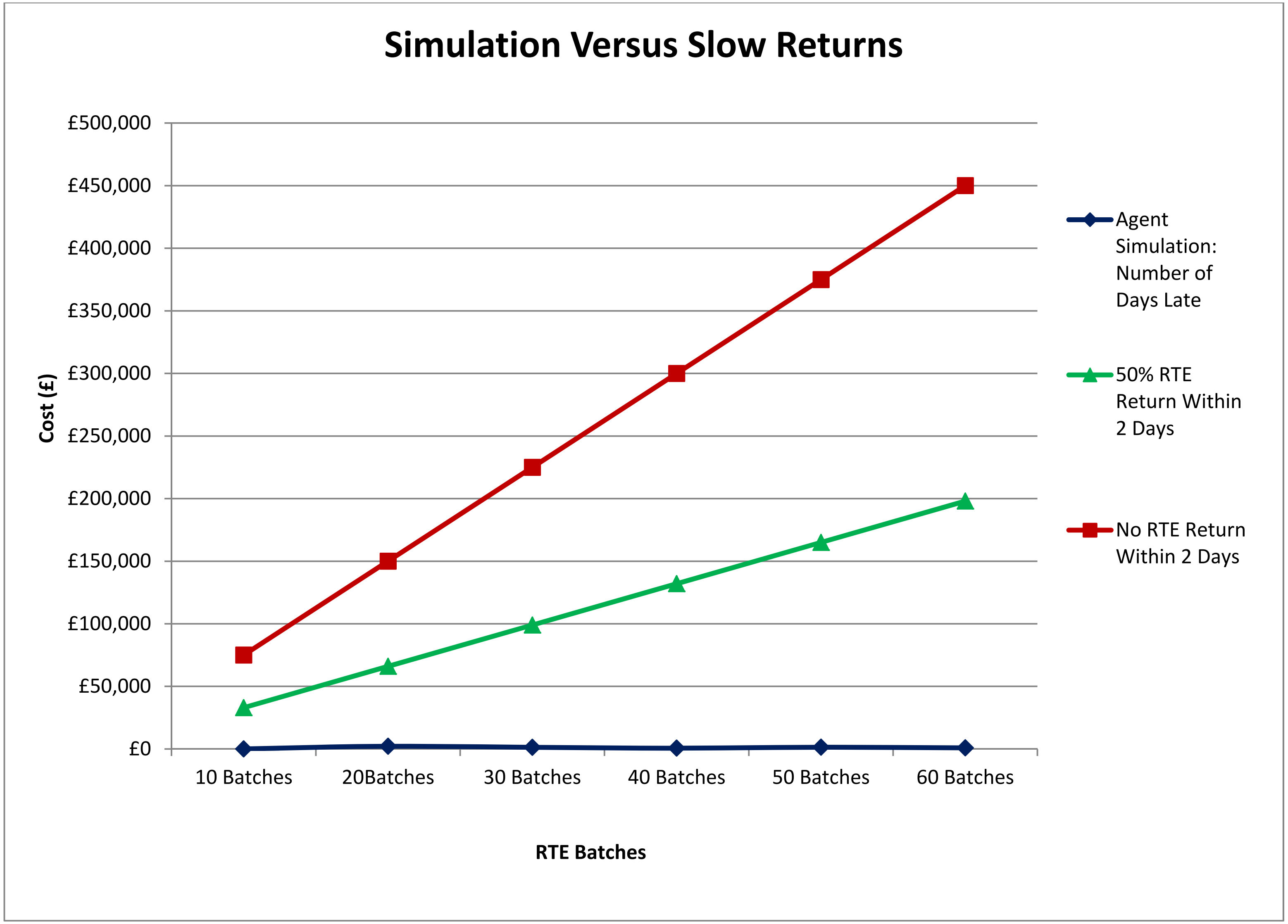


Figure 6.8 Graph Illustrating the Simulation Results versus Slow Returns

Losses for 15 Simulated Days			
Batch Quantity	10%	20%	80%
10 Batches	1	2	8
20 batches	2	4	16
30 Batches	3	6	24
40 Batches	4	8	32
50 Batches	5	10	40
60 Batches	6	12	48

Table 6.10 Calculation for Losses

	10% losses					
Day of simulation	**10 Batches**	**20 batches**	**30 Batches**	**40 Batches**	**50 Batches**	**60 Batches**
1	14	28	42	56	70	84
2	13	26	39	52	65	78
3	12	24	36	48	60	72
4	11	22	33	44	55	66
5	10	20	30	40	50	60
6	9	18	27	36	45	54
7	8	16	24	32	40	48
8	7	14	21	28	35	42
9	6	12	18	24	30	36
10	5	10	15	20	25	30
11	4	8	12	16	20	24
12	3	6	9	12	15	18
13	2	4	6	8	10	12
14	1	2	3	4	5	6
15	0	0	0	0	0	0
	105	**210**	**315**	**420**	**525**	**630**

Table 6.11 Number of Days Late for 10% Losses

	20% losses					
Day of simulation	**10 Batches**	**20 batches**	**30 Batches**	**40 Batches**	**50 Batches**	**60 Batches**
1	28	56	84	112	140	168
2	26	52	78	104	130	156
3	24	48	72	96	120	144
4	22	44	66	88	110	132
5	20	40	60	80	100	120
6	18	36	54	72	90	108
7	16	32	48	64	80	96
8	14	28	42	56	70	84
9	12	24	36	48	60	72
10	10	20	30	40	50	60
11	8	16	24	32	40	48
12	6	12	18	24	30	36
13	4	8	12	16	20	24
14	2	4	6	8	10	12
15	0	0	0	0	0	0
	210	**420**	**630**	**840**	**1050**	**1260**

Table 6.12 Number of Days Late for 20% Losses

	80% losses					
Day of simulation	**10 Batches**	**20 batches**	**30 Batches**	**40 Batches**	**50 Batches**	**60 Batches**
1	112	224	336	448	560	672
2	104	208	312	416	520	624
3	96	192	288	384	480	576
4	88	176	264	352	440	528
5	80	160	240	320	400	480
6	72	144	216	288	360	432
7	64	128	192	256	320	384
8	56	112	168	224	280	336
9	48	96	144	192	240	288
10	40	80	120	160	200	240
11	32	64	96	128	160	192
12	24	48	72	96	120	144
13	16	32	48	64	80	96
14	8	16	24	32	40	48
15	0	0	0	0	0	0
	840	**1680**	**2520**	**3360**	**4200**	**5040**

Table 6.13 Number of Days Late for 80% Losses

	Agent Simulation	20% Losses	10% Losses	80% Losses
10 Batches	1	210	105	840
20 batches	27	420	210	1680
30 Batches	16	630	315	2520
40 Batches	8	840	420	3360
50 Batches	17	1050	525	4200
60 Batches	11	1260	630	5040

Table 6.14 Number of Days Late for Batches

	Agent Simulation	20% Losses	10% Losses	80% Losses
10 Batches	£80	£16,800	£ 8,400	£ 67,200
20 batches	£2,160	£33,600	£16,800	£ 134,400
30 Batches	£ 1,280	£50,400	£25,200	£ 201,600
40 Batches	£ 640	£67,200	£33,600	£268,800
50 Batches	£1,360	£ 84,000	£42,000	£ 336,000
60 Batches	£ 880	£ 100,800	£50,400	£ 403,200

Table 6.15 Cost of Losses versus Simulation Results

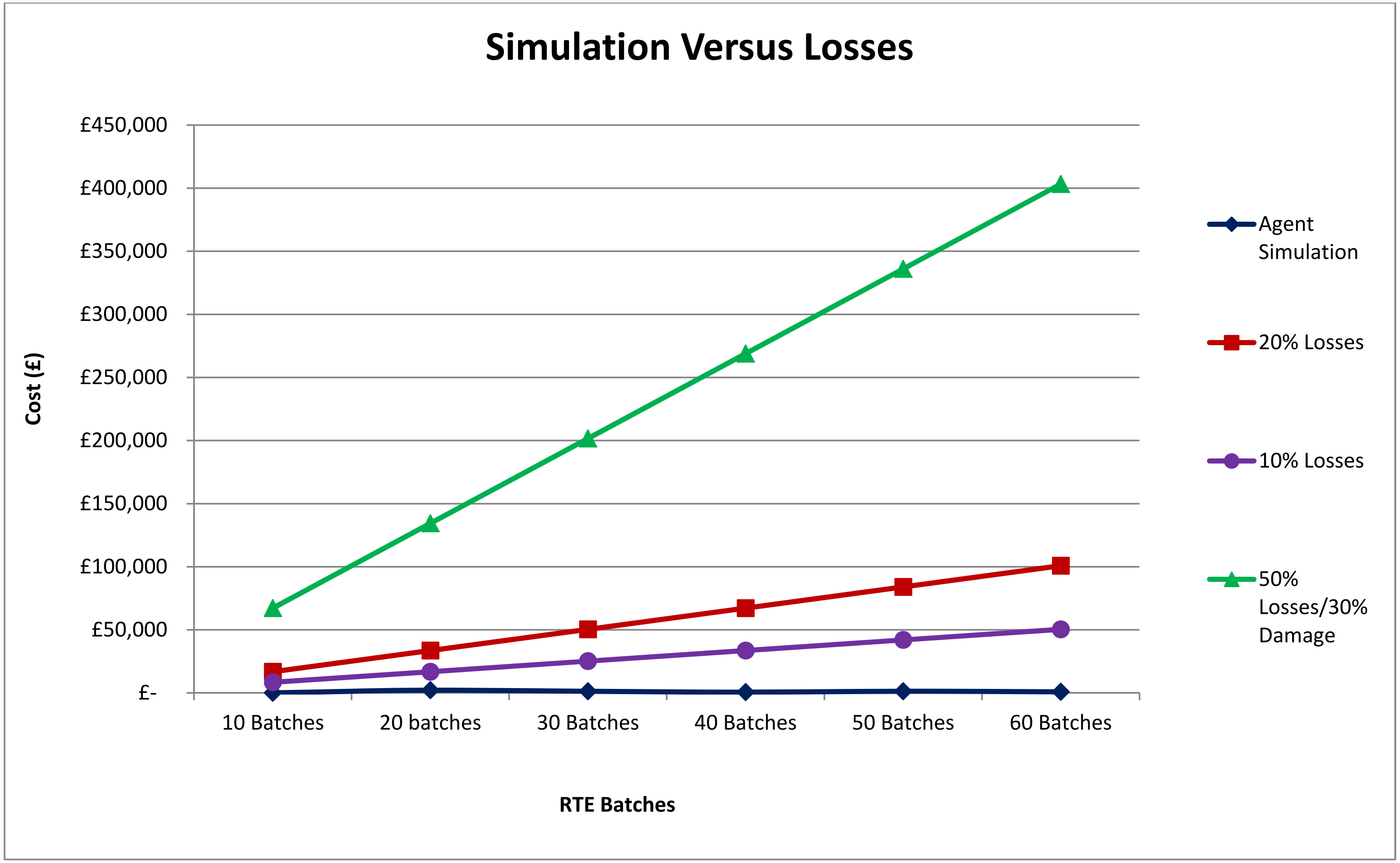


Figure 6.9 Comparison of Simulation Results Versus Losses

6.4 Batch Processing Versus Fleet Costs

The scenario conducted in section 6.3, demonstrated the impact of the model on returning the RTE. This section explores the effect of increasing the batch processing time and the impact on the fleet. The parameters for the scenario are illustrated in Figure 6.10 and the batch processing day is modified by 2 to 9 days. As the batch processing day is increased, the number of late batch days and fleet operating costs are presented in Table 6.16, which is illustrated in Figure 6.11.

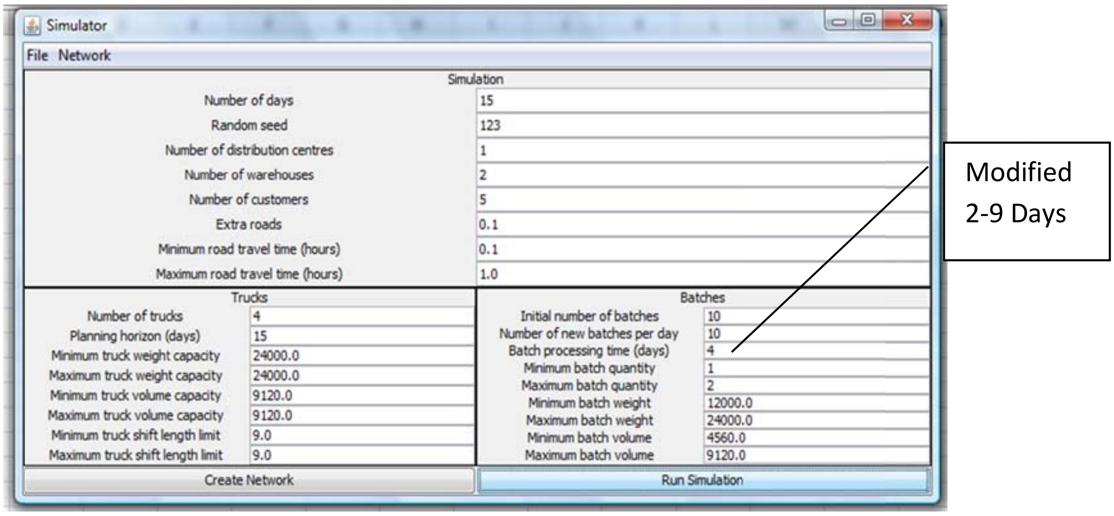


Figure 6.10 Scenario Set-up

Figure 6.11 illustrates that there are certain batch processing days, which return RTE batches without incurring a high number of late days for this particular scenario. In addition, the fleet indicated that there are certain number of batch processing days, which are required for the fleet to reduce its operating cost. For the scenario conducted, as the return date is increased, the late batch cost stay relatively even, apart from at 6 days where there is a slight rise.

The graph illustrates that the cost of fleet will rise as the batch processing days increase because the longer the batches are waiting to be moved then there are less opportunities available to be collected for delivery earlier. When the batches are finally ready to be picked-up then there is a strain on the fleet resources. As the dates for delivery are randomly generated, there will be variable fleet costs but it is evident in Figure 6.11 that the rise in batch processing days increases the truck operating costs, and therefore it is beneficial for the truck agent to get batches back quicker, improving the RTE utilisation rate.

Batch Processing Days	Number of Days Batches Late	Fleet Operating Values	Late Batch Cost	Fleet Operating Cost	Total Cost
2 Days	14	404.64	£ 1,120.00	£ 32,371.20	£ 33,491.20
3 Days	8	393.66	£ 640.00	£ 31,492.80	£ 32,132.80
4 Days	6	373.2	£ 480.00	£ 29,856.00	£ 30,336.00
5 Days	2	386.04	£ 160.00	£ 30,883.20	£ 31,043.20
6 Days	24	391.84	£ 1,920.00	£ 31,347.20	£ 33,267.20
7 Days	8	412.76	£ 640.00	£ 33,020.80	£ 33,660.80
8 Days	12	424.58	£ 960.00	£ 33,966.40	£ 34,926.40
9 Days	3	439.9	£ 240.00	£ 35,192.00	£ 35,432.00
10 Days	17	480.1	£ 1,360.00	£ 38,408.00	£ 39,768.00

Table 6.16 Comparison of the Return Date Increase Versus Truck Operating Costs

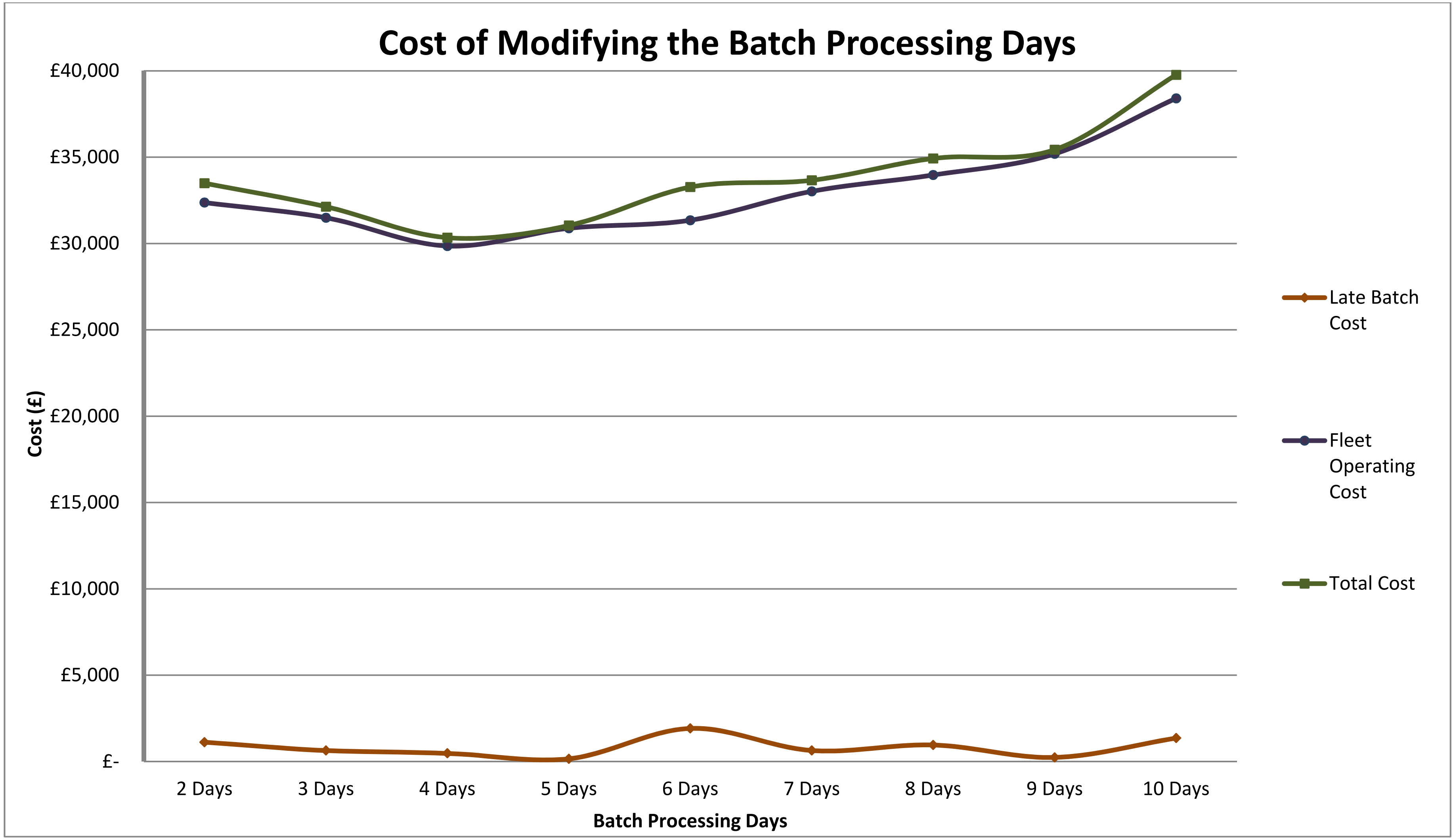


Figure 6.11 Comparison of Fleet Operating Cost Versus the Late RTE Batch Cost

6.5 Increasing Network Complexity

In the real-world there are many different route options for a truck to select when compiling a schedule. On each potential route, there may be an opportunity to collect RTE batches for return, improving utilisation rates. Therefore, scenarios with network complexity are conducted to investigate the impact on the fleet and RTE. In the test scenario, the network complexity is set at 0.9, which increases the number of links between the facilities in comparison to the control scenario as illustrated in Figure 6.12.

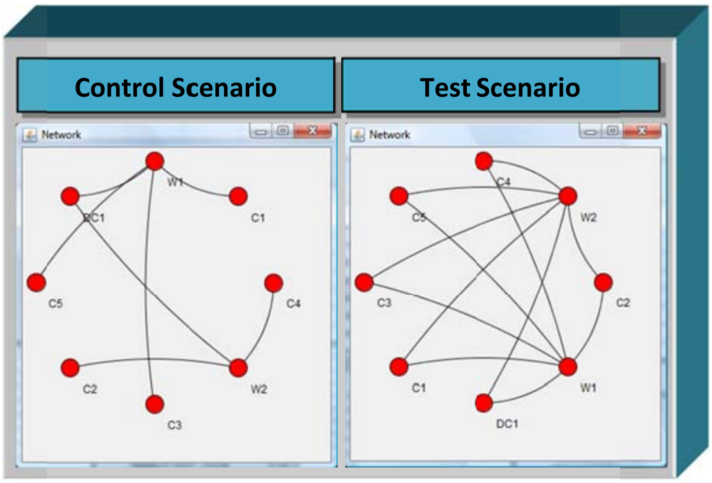


Figure 6.12 Control Scenario Versus Test Scenario

Figure 6.13 illustrates the impact of increasing the RTE batch quantity from 10 batches to 60 batches at an interval of 10 batches and its effect on truck operating costs and late RTE batch costs. It is observed that for lower quantities of batches to be scheduled, between 20-50 batches the late RTE batch costs and truck operating costs increase. For batches 20-30, the truck cost increases indicating that the trucks are picking up a certain number of load, after which it is not possible for the Truck agent to negotiate further load pick-ups, which adhere to the destination and delivery requirements of its current route. This impact leads to a rise in late batch costs. It is noted that by offering trucks more variety, the cost of late batches rise because there is a higher probability of not being selected for scheduling due to the increase in competition. Therefore, to reduce late batch costs, there is a case to be made to restrict the number of links between facilities in a network when devising the operating values of the functions. Although this is the case for lower batch quantities, as the quantity is increased to 60 batches, network complexity has less impact on the number of late batches. This is because the number of facilities remain the same, whilst the quantity of batches at each facility increases, therefore enhancing the probability of a batch being picked up as a truck has more of an incentive to take the route (larger load collection). The impact of increasing the number of facilities is tested in section 6.6.

Figure 6.14 illustrates the truck utilisation rates of time, weight and volume across a range of batch quantities. It is noted that in many of the simulations, the volume and weight utilisation rates of the truck are higher as the network is more complex. In addition, there is less variability in the weight utilisation rate, which is between 40% to 60% for the control scenario, and more stable for the complex scenario at 52% to 58%. This is because by adding more links between facilities, the options for the truck to pick-up RTE batches increases. This correlates with the time utilisation rate, which increases at batches 20 to 30, as the addition of a *newLeg* method is implemented in the model, which enables a truck's driving time to be fully utilised. However, as the quantity of batches in the system increases to 60 batches, the time utilisation decreases. This indicates that as the loads at each facility increases, the truck's ability to add more legs to the route and fully utilise its driving time becomes more challenging. Therefore, there is a case to made to allow the RTE batches to wait at the facilities in order to become of sufficient size, encouraging trucks to pick-up the loads, however this would have a negative effect on the RTE utilisation rate in comparison to a quick return strategy.

Figure 6.15 confirms that as the quantity of the RTE batches increases, the number of auctions conducted also rises. However, with a complex network the number of auctions rises at 20 batches per day in comparison to the control scenario. This concurs with the time utilisation value presented in Figure 6.14, where at 20 batches the highest level of utilisation was reached, and therefore a high number of auctions were conducted to ensure that the amount of fleet utilisation was pushed to the upper limit.

Figure 6.16 illustrates the effect of increasing the batch processing day, which is the amount of time each batch spends at the customer's facility. The graph indicates that for a complex road structure, in comparison to the control scenario, the late batch cost is more stable across all the batch processing days which have been tested. This is because there are more opportunities along each scheduled route to make stops and pick-up batches in comparison to the control scenario, which has fewer roads. The increase in opportunities has also impacted the truck costs as the complex network is able to provide an environment to maintain lower truck operating costs, and in addition maintain more on time batch deliveries in comparison to a scenario of fewer roads. Figure 6.17 demonstrates the large variability of time utilisation, which is approximately 56% to 87% for a scenario with high network complexity in comparison to the control scenario, which has a variability of 78 to 91%. Returning RTE after 4 days of batch processing reduces the fleet utilisation, and therefore leaves more truck capacity for further batches or the option to reduce the fleet. Figure 6.17 also illustrates that for certain batch processing days, for example 5 days or 7 days, the volume and weight utilisation rates are higher, whilst the time utilisation is lower. This suggests that by allowing the batch to be at the customer's facility for a certain number of days, it will enable the truck to negotiate a schedule with a higher volume and weight utilisation in less driving time. Figure 6.18 illustrates the auction rate and for both the control and road complexity scenarios, the number of auctions range from 1500-2000. At most stages, each of the scenarios has different quantities of auctions but generally the network complexity scenarios have a lower number of auctions, which is due to the higher level of opportunities for a batch pick-up, allowing the formation of a schedule quicker. At the batch processing day of 3, Figure 6.17 illustrates that the RTE batches have a similar rate of time utilisation at 90% and 87%, however to achieve the 3% extra for the control scenario, approximately 400 extra auctions need to be conducted, and therefore highlighting the significance of the auction process.

Therefore, setting a batch processing day and increasing the network complexity is advantageous for the trucks' schedule in regards to the operating time and utilisation rates. However, if the batch processing day is too long, then it will not be advantageous for the RTE batch utilisation rate as this requires a quicker return. Section 6.6 explores the impact of increasing the number of facilities.

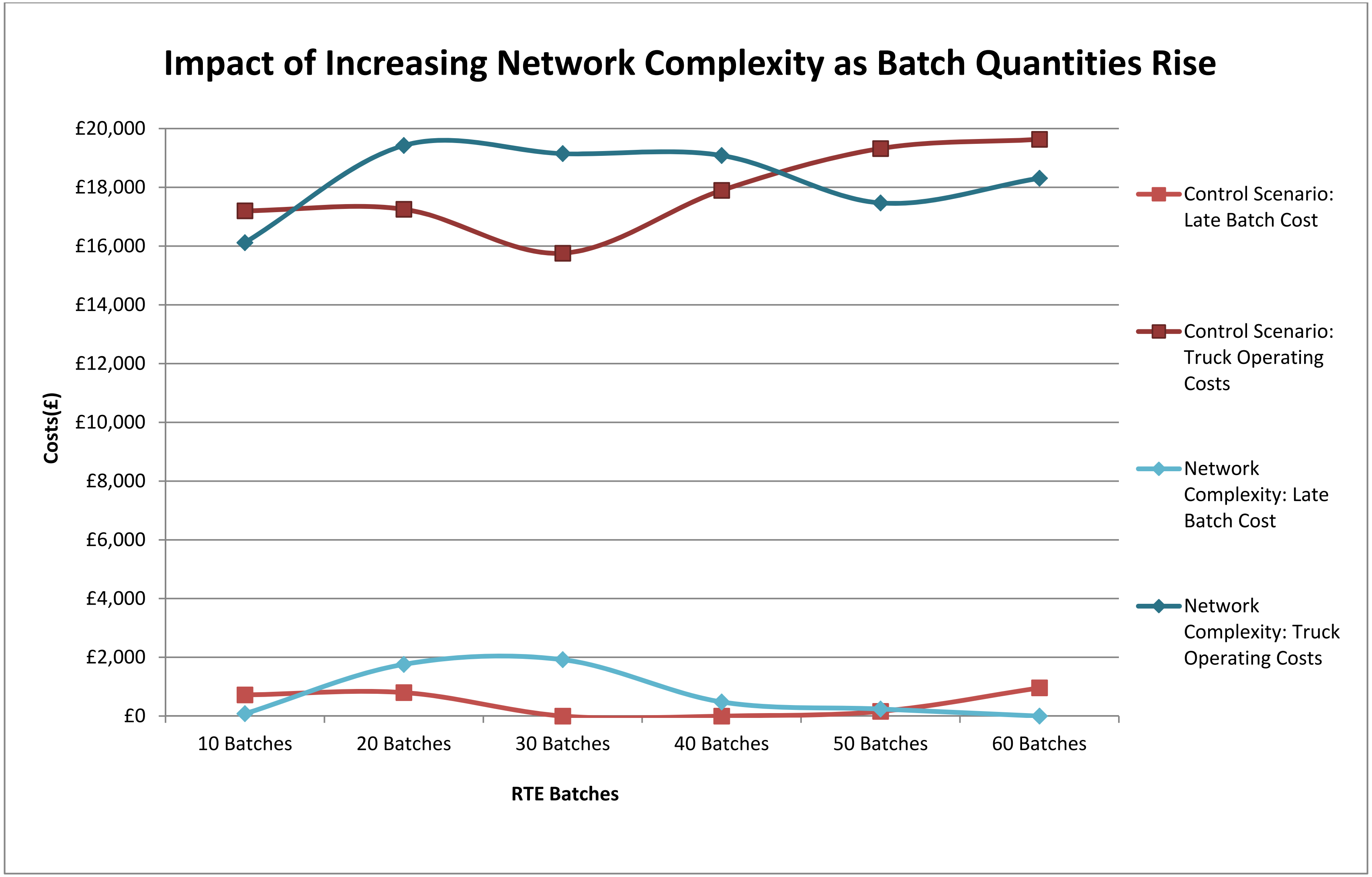


Figure 6.13 Impact of Increasing Network Complexity as Batch Quantities Rise

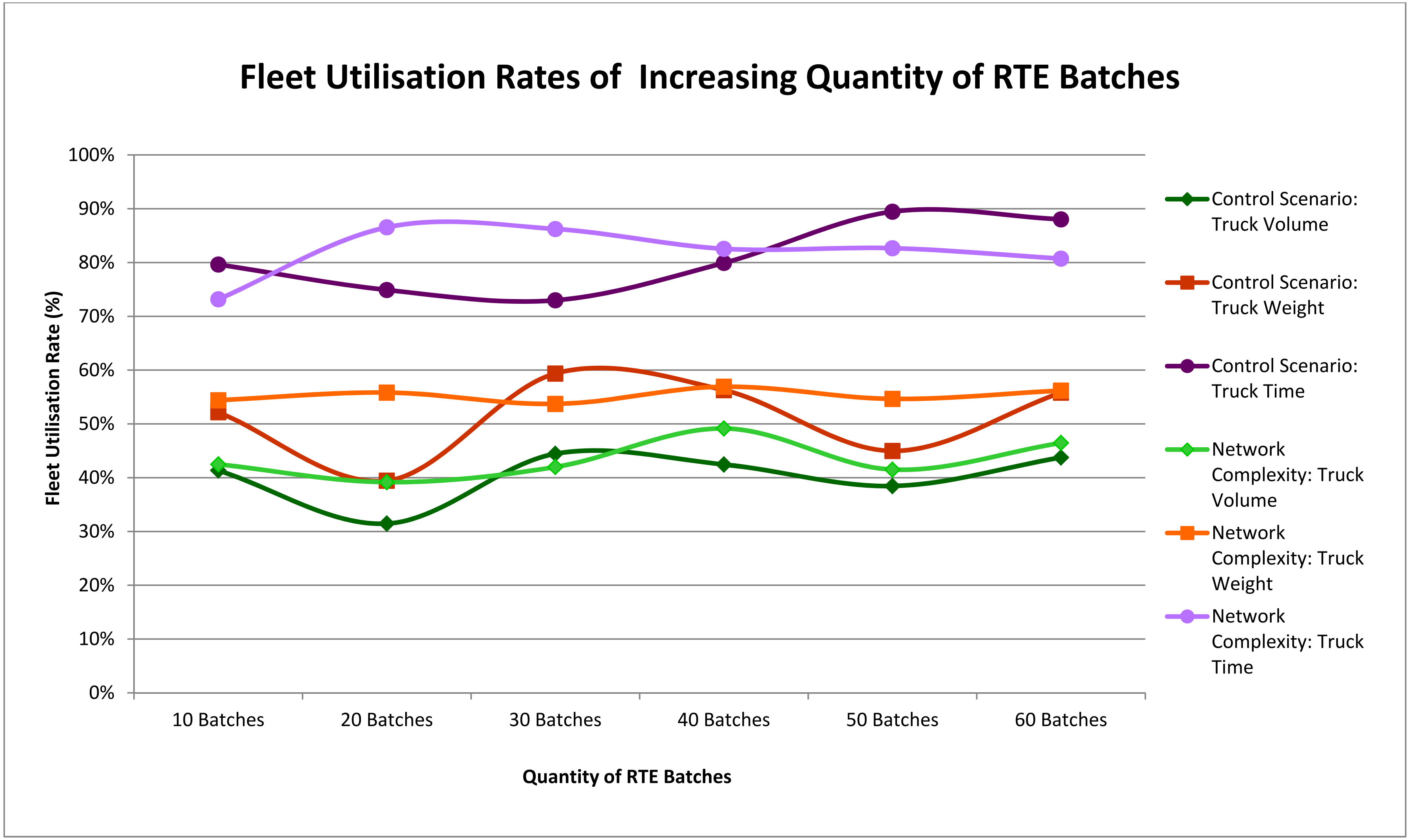


Figure 6.14 Fleet Utilisation Rates of Increasing Quantity of RTE Batches for a Complex Network

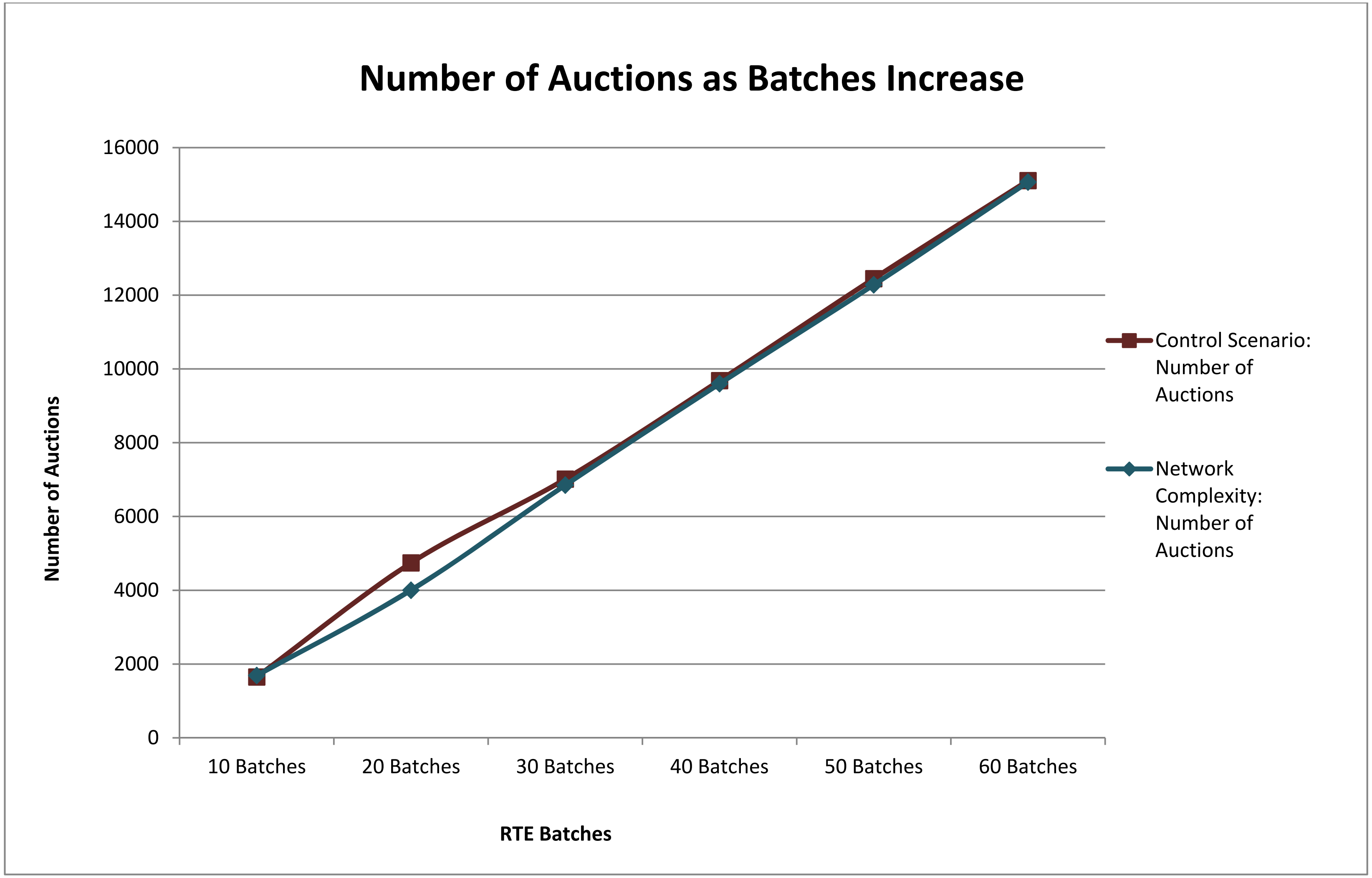


Figure 6.15 Number of Auctions as Batches Increase for a Complex Network

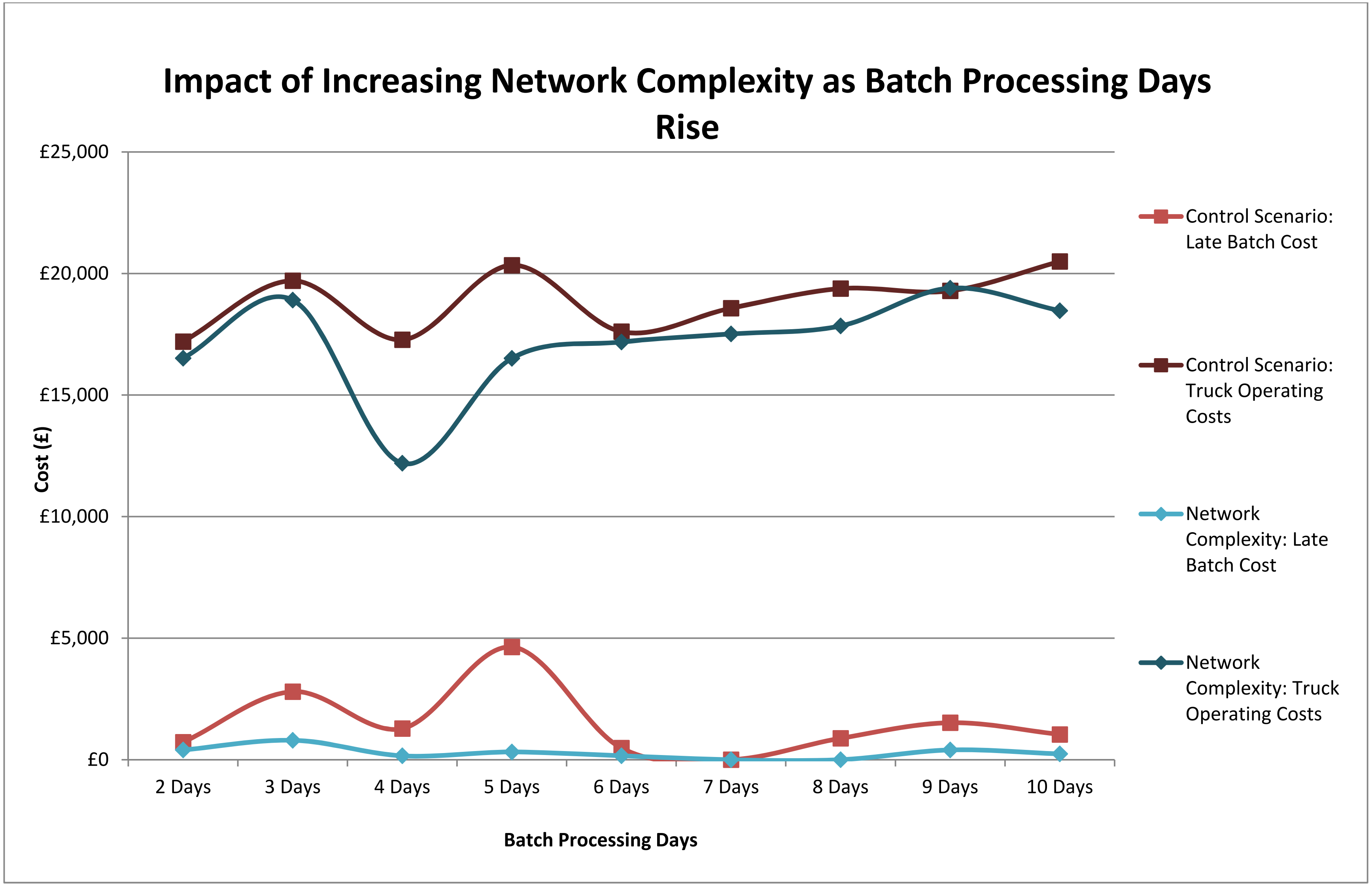


Figure 6.16 Impact of Increasing Network Complexity as Batch Processing Days Rise

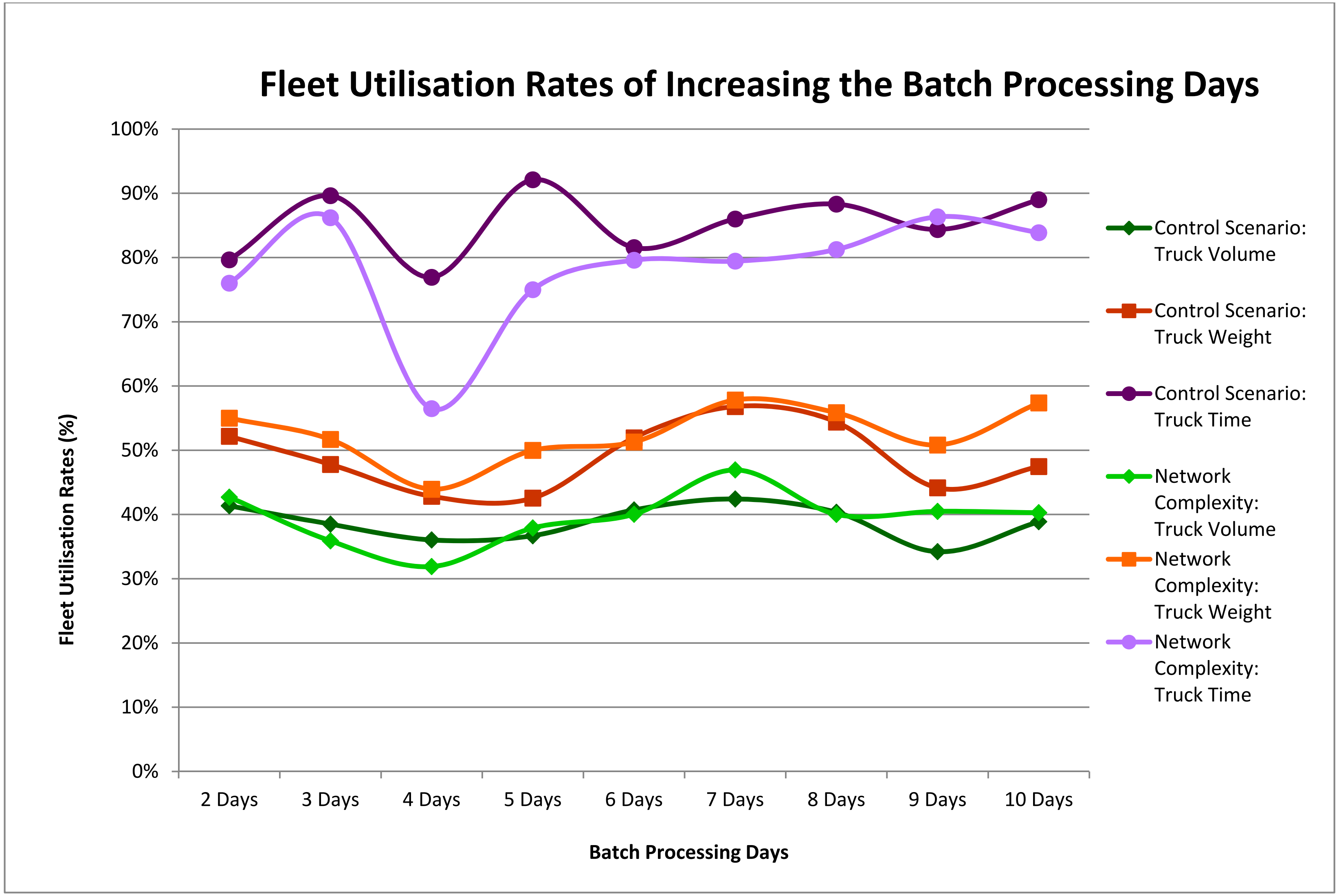


Figure 6.17 Fleet Utilisation Rates of Increasing the Batch Processing Days for a Complex Network

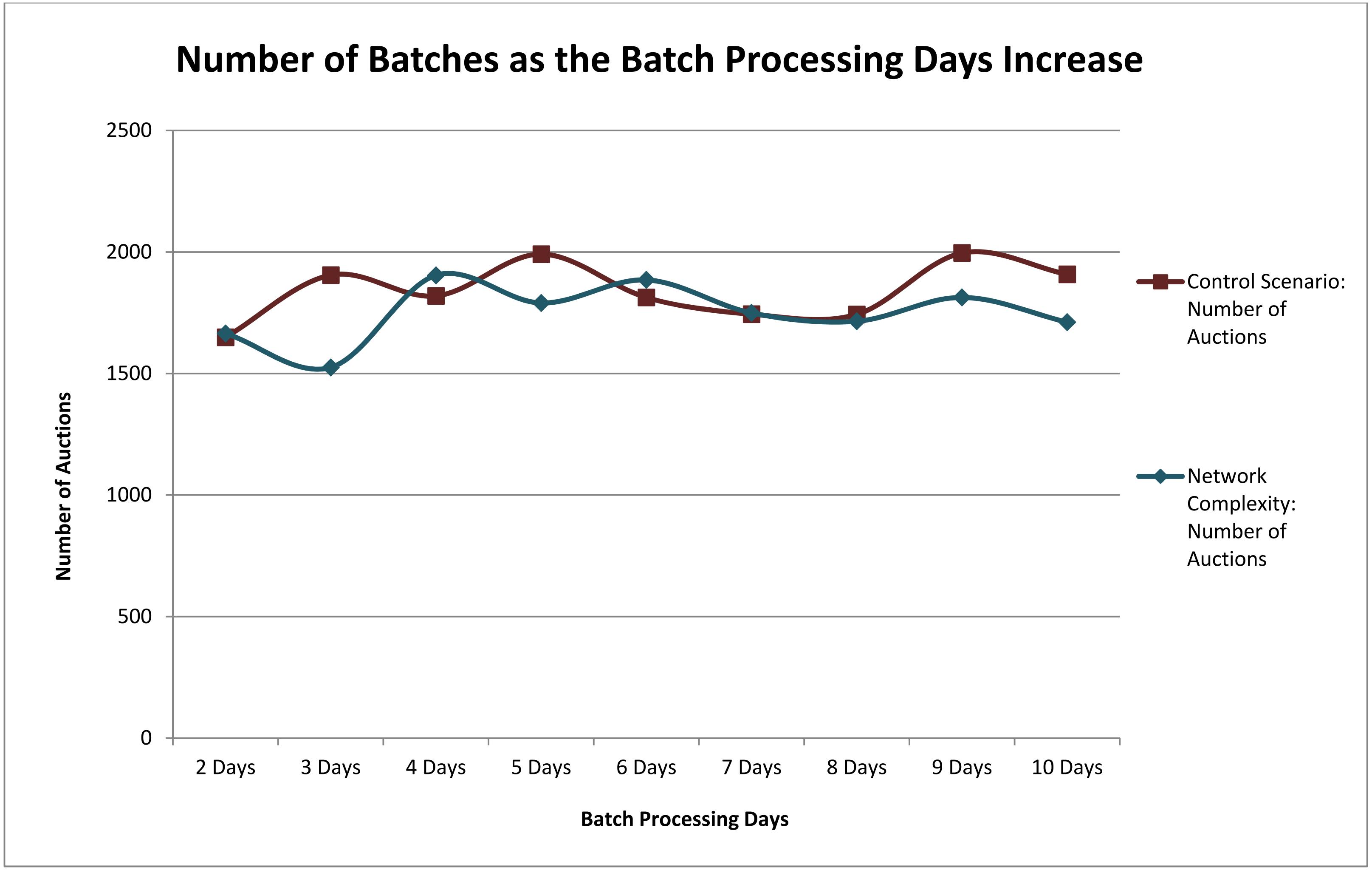


Figure 6.18 Number of Auctions as the Batch Processing Days Increase for a Complex Network

6.6 Larger Network

Logistics service providers have to contend with a variety of networks, each of which is a different size, therefore the control scenario, which consists of a network of 1 distribution centre, 2 warehouses and 5 customers is compared to a larger network consisting of 4 distribution centres, 8 warehouses and 20 customers as illustrated in Figure 6.19.

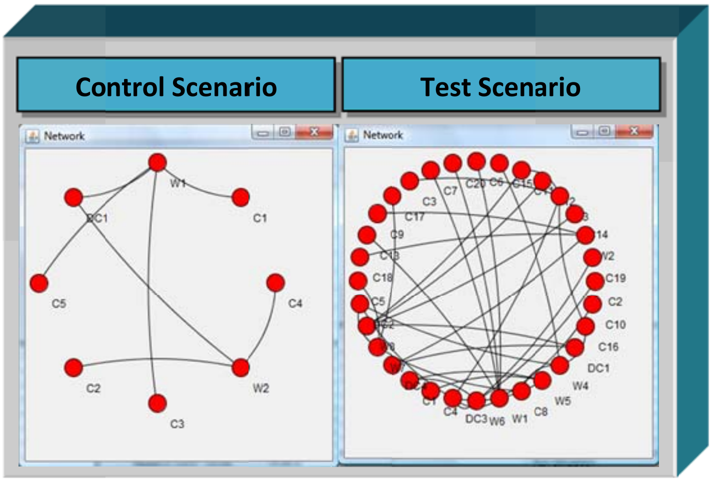


Figure 6.19 Control Scenario Versus Test Scenario

Figure 6.20 illustrates the impact of a higher quantity of facilities on truck operating and late RTE batch costs. The results show that as the number of facilities is increased, the late RTE batch costs rise. This is because the quantity of RTE batches is the same as in the control scenarios, and therefore is spread across more facilities. This leads to the trucks having to pick up more RTE batches in order to meet time, volume and weight requirements. However, the driving hour limit is reached before more loads are collected as illustrated in Figure 6.21, where the weight and volume utilisation rates are lower and the time utilisation higher than the control scenario. Therefore, more batches will be waiting in the network for delivery, increasing the RTE late batch cost as demonstrated in Figure 6.20. Both Figure 6.20 and Figure 6.21 illustrate that as the batch quantities are increased to 60, the costs and rates are more aligned with the control scenario. Therefore, as the batch quantity in a network with larger facilities increases, this increases the quantity of RTE batches at each facility, making a stop at a facility more advantageous for the truck as a larger load can be collected within the time frame. It is noted that a network with more facilities and a small quantity of RTE batches, overstretches the fleet, therefore a case may be made to assign each truck a region to facilitate deliveries, so that the truck can pick up as many loads within a specified area. Possibly considering a strategy where RTE are trunked at designated facilities, may increase the quantity of loads at the facilities for each batch. This impacts the rules devised in terms of the truck routing limitations. Despite the lower weight and volume utilisation, the number of auctions conducted is similar to the control scenario as illustrated in Figure 6.22, apart from at 20 batches where less auctions are needed for the fleet to complete the scheduling.

Figure 6.23 illustrates the impact of the batch processing day on the control scenario in comparison to the larger network scenario. It is noted that generally the late batch costs and truck operating costs are lower for the larger network scenario despite the volume, weight and utilisation rates being lower than the control scenario as illustrated in Figure 6.24. With a larger number of facilities, as the links between the facilities are increased, once a batch is processed and made available for collection, it is more likely to be eligible for scheduling by truck than in a smaller network. Therefore, for a network with fewer facilities there is a case to be made to add incentives to make batches more desirable to encourage pick-ups by possibly increasing the number of links available between facilities. The auction rates vary for each batch processing day as illustrated in Figure 6.25. For the larger network there is a larger range in the amount of auctions, which are between 1500 and 1900 whereas for a smaller network, the range is between 1650 and 2000 indicating that the negotiation intensity is higher for a larger network.

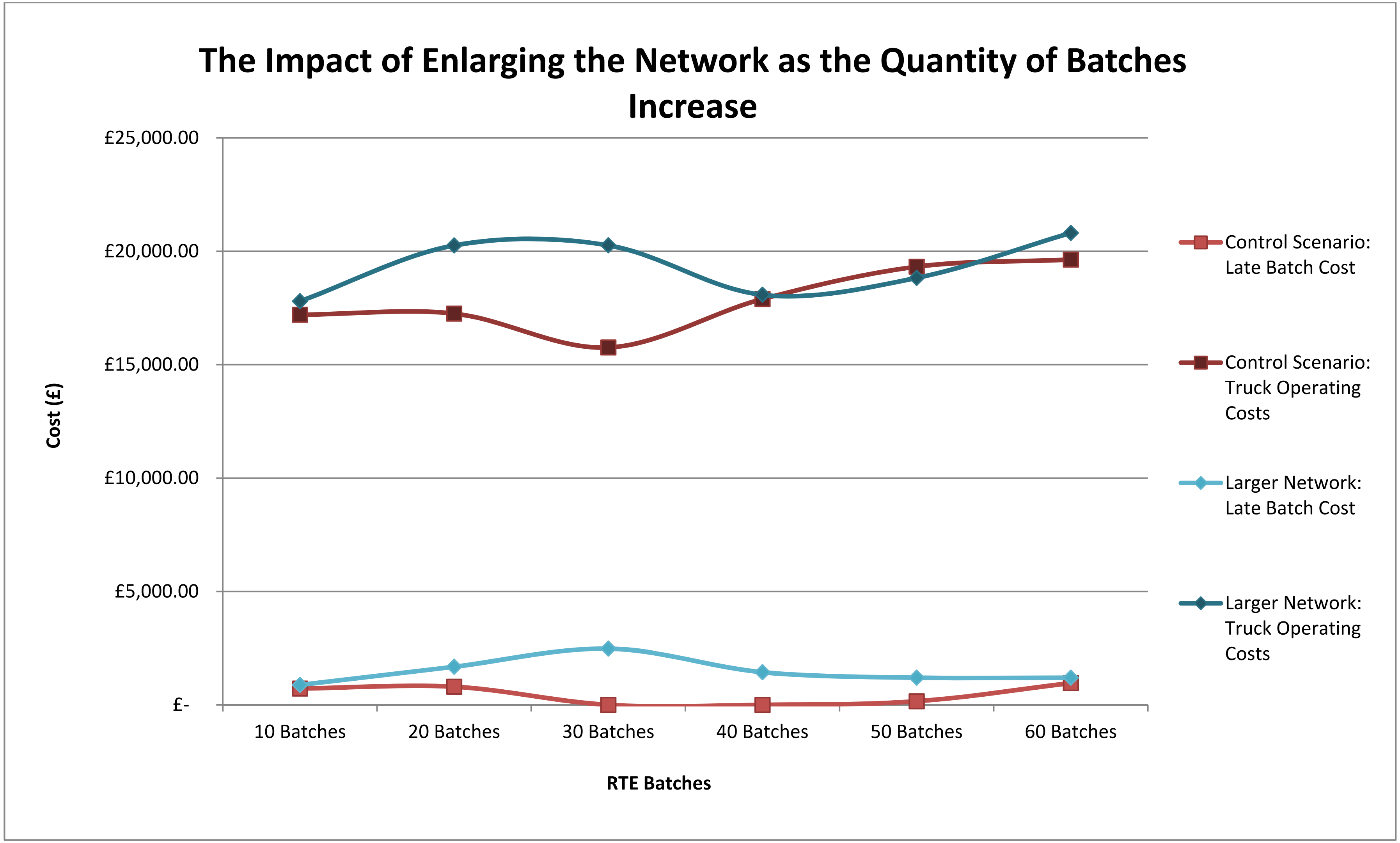


Figure 6.20 Impact of Enlarging the Network as Batch Quantities Rise

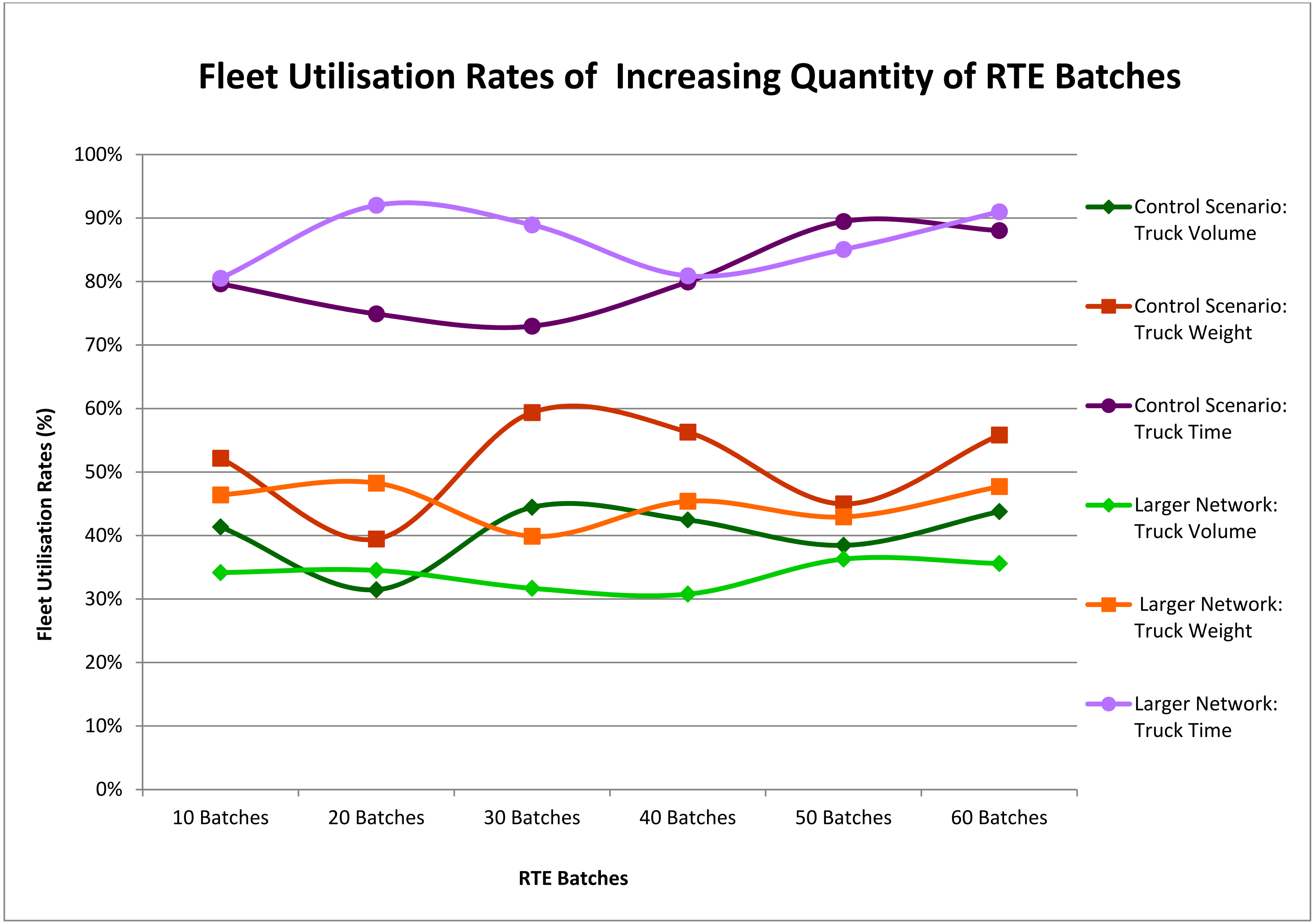


Figure 6.21 Fleet Utilisation Rates of Increasing Quantity of RTE Batches as the Network is Enlarged

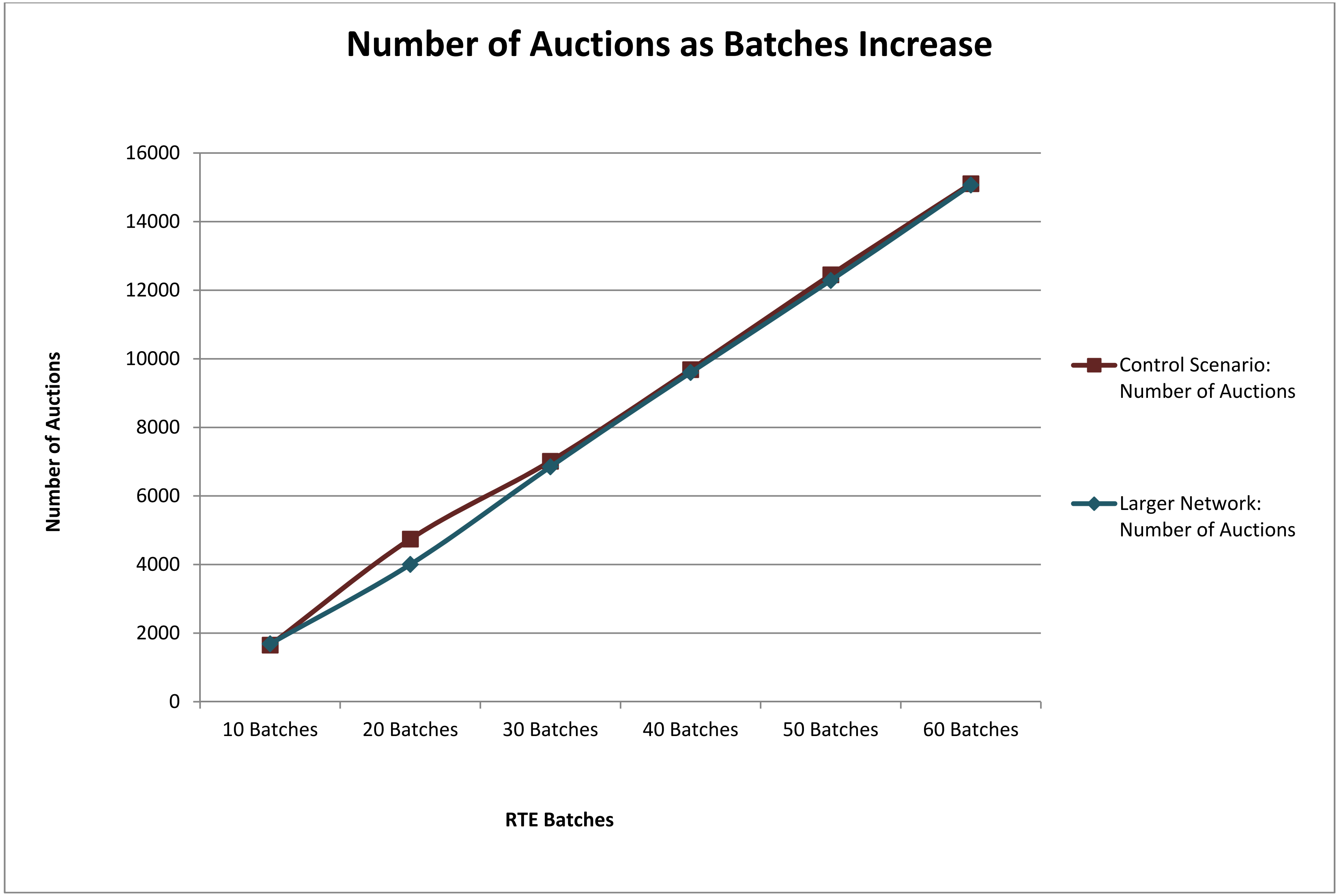


Figure 6.22 Number of Auctions as the Network is Enlarged

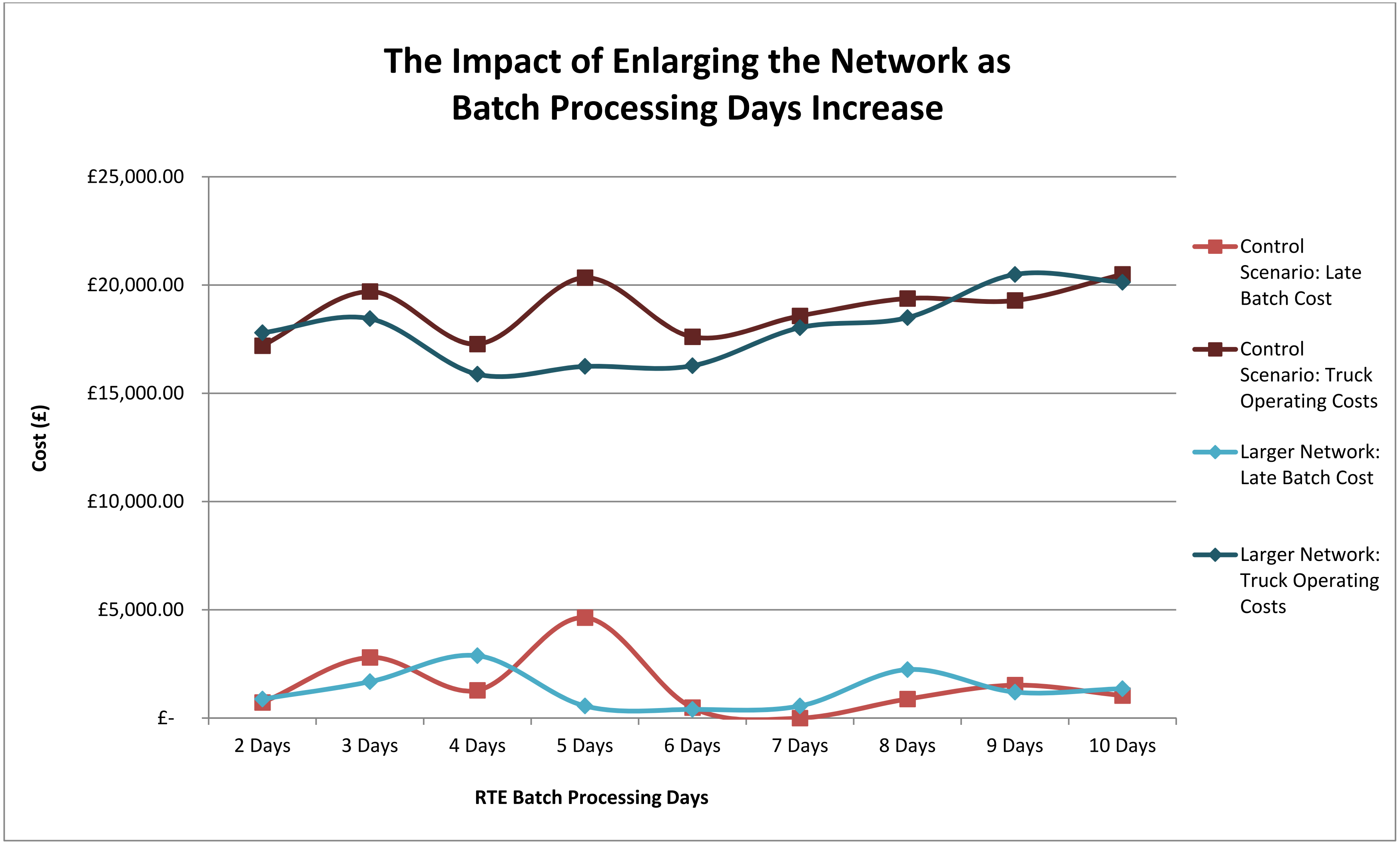


Figure 6.23 Impact of Enlarging the Network as Batch Processing Days Rise

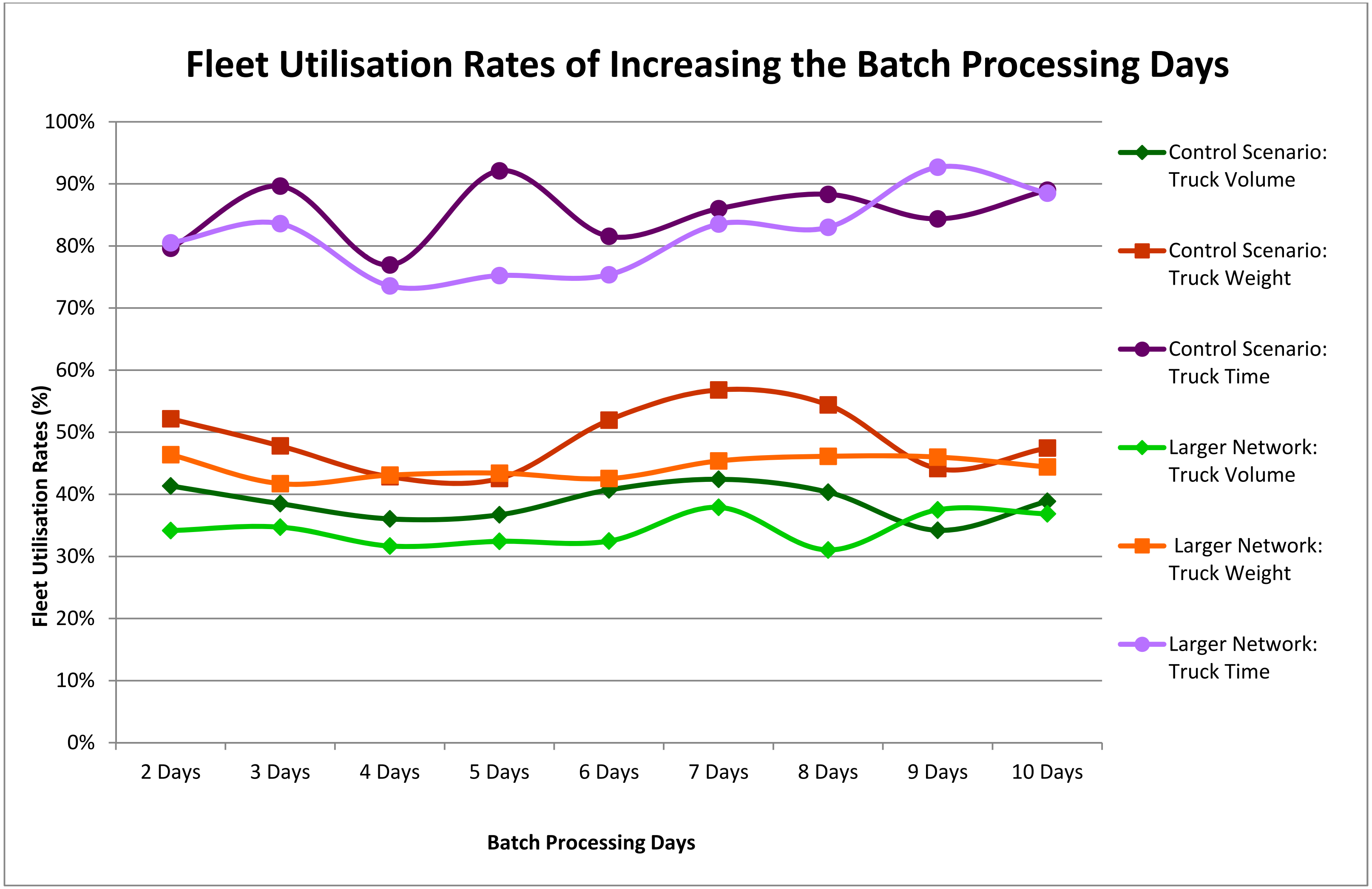


Figure 6.24 Fleet Utilisation Rates of Increasing the Batch Processing Days as the Network is Enlarged

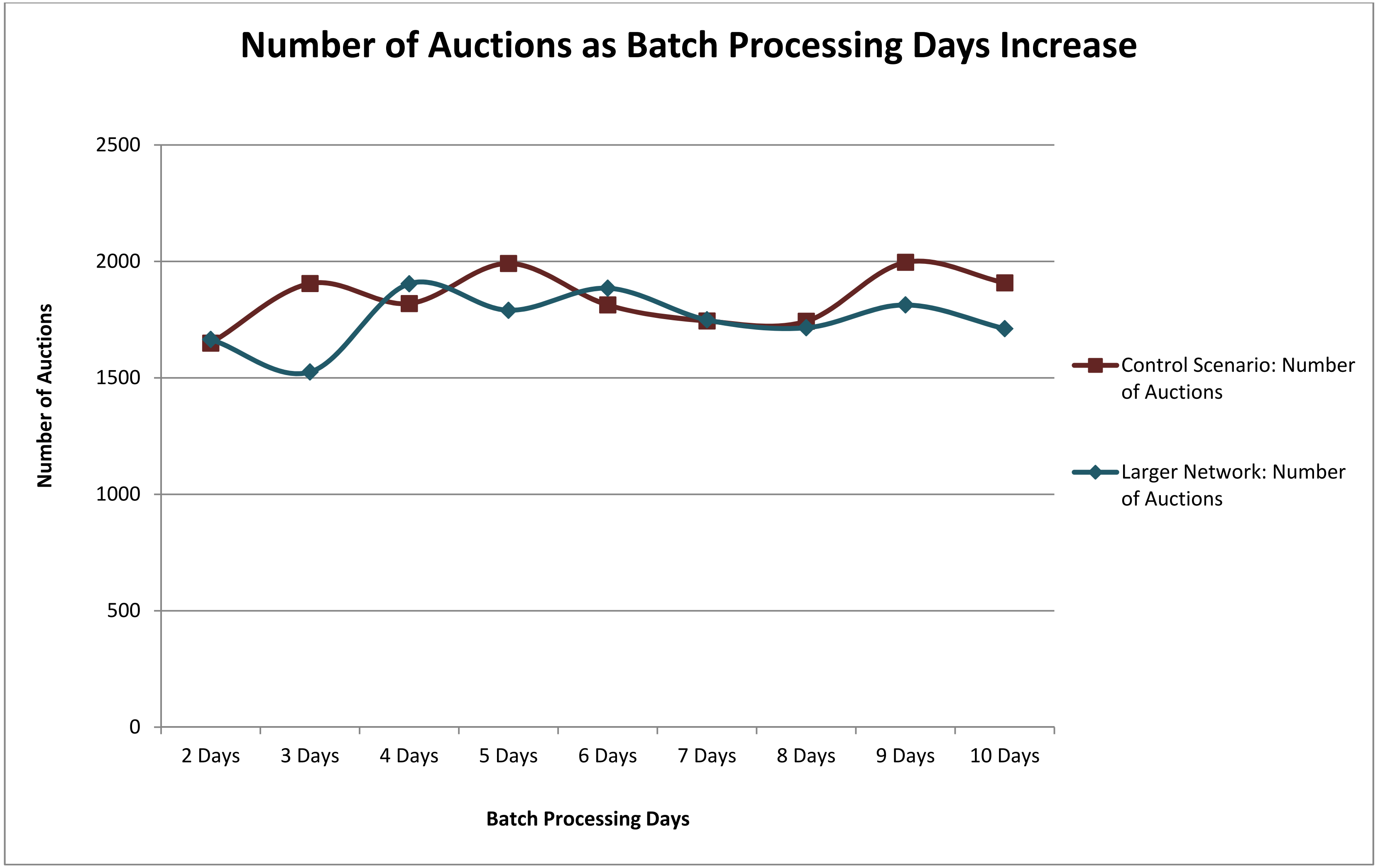
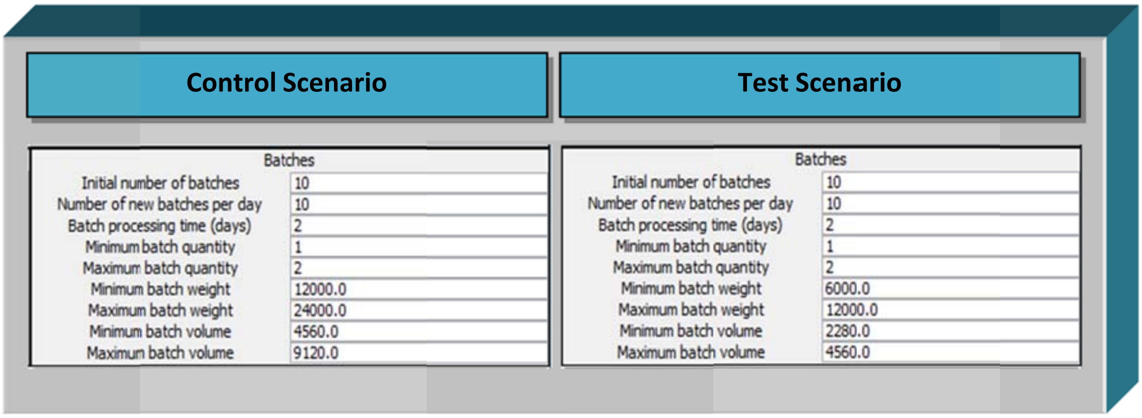


Figure 6.25 Number of Auctions as the Batch Processing Days Increase and the Network is Enlarged

6.7 **Smaller Batch Sizes**

The quantity and size of batches, which are introduced to the network will vary, particularly for a logistics service provider with a large portfolio of clients, across an array of industries with different product types. A smaller size of RTE batches is tested and the simulation is set up as per the parameters presented in Figure 6.26.



Control Scenario

Batches	
Initial number of batches	10
Number of new batches per day	10
Batch processing time (days)	2
Minimum batch quantity	1
Maximum batch quantity	2
Minimum batch weight	12000.0
Maximum batch weight	24000.0
Minimum batch volume	4560.0
Maximum batch volume	9120.0

Test Scenario

Batches	
Initial number of batches	10
Number of new batches per day	10
Batch processing time (days)	2
Minimum batch quantity	1
Maximum batch quantity	2
Minimum batch weight	6000.0
Maximum batch weight	12000.0
Minimum batch volume	2280.0
Maximum batch volume	4560.0

Figure 6.26 Control Scenario Versus Test Scenario

Figure 6.27 illustrates the impact of introducing smaller RTE batches into the network. It is noted that although the late batch cost is similar between the two scenarios, in most of the instances (with the exception of 50 batches), the truck operating cost is higher for smaller batch loads. Further investigation into the utilisation rates presented in Figure 6.28 illustrates that for smaller RTE batches, the volume and weight utilisation rates are significantly lower, whilst the time utilisation for the trucks is higher. Therefore, trucks are unable to pick up enough loads during the time available to increase the utilisation, although this does not impact on the late RTE batch cost. Furthermore, with smaller batches, the fleet costs rise, however the late RTE batch costs remain similar, leading to a good rate of RTE utilisation. The auction rates in Figure 6.29 are similar between the two scenarios but there is a case to be made to limit the amount of smaller batches in the system, asserting the requirement to batch RTE when scheduling instead of disseminating decision-making to each individual RTE, as this will lead to a rise in the truck operating cost.

Figure 6.30 illustrates the results of the impact of implementing smaller batch sizes across a range of batch processing days. The late batch cost is stable across all the batch processing days, however the fleet costs rise as the batch processing day increases. This indicates that with smaller batches, which wait at the customer facility for processing for a longer number of days, impacts the total cost on the fleet. Therefore, there is a case to be made to implement a strategy of a lower batch processing day (2-5 days) in order to benefit the fleet, by enabling them to schedule more efficient deliveries, and also return the RTE back quicker for re-use. This is asserted by Figure 6.31, where the time utilisation of the truck is higher as the batch processing day increases, foregoing the missed opportunity to use the available driving hours earlier. The lowest number of auctions is for the highest batch processing day, 10, where Figure 6.32 illustrates that as the time utilisation rate is reached quicker, the scheduling process is completed, due to the time constraint. Therefore a case is to be made to limit the batch processing day to facilitate a more responsive fleet.

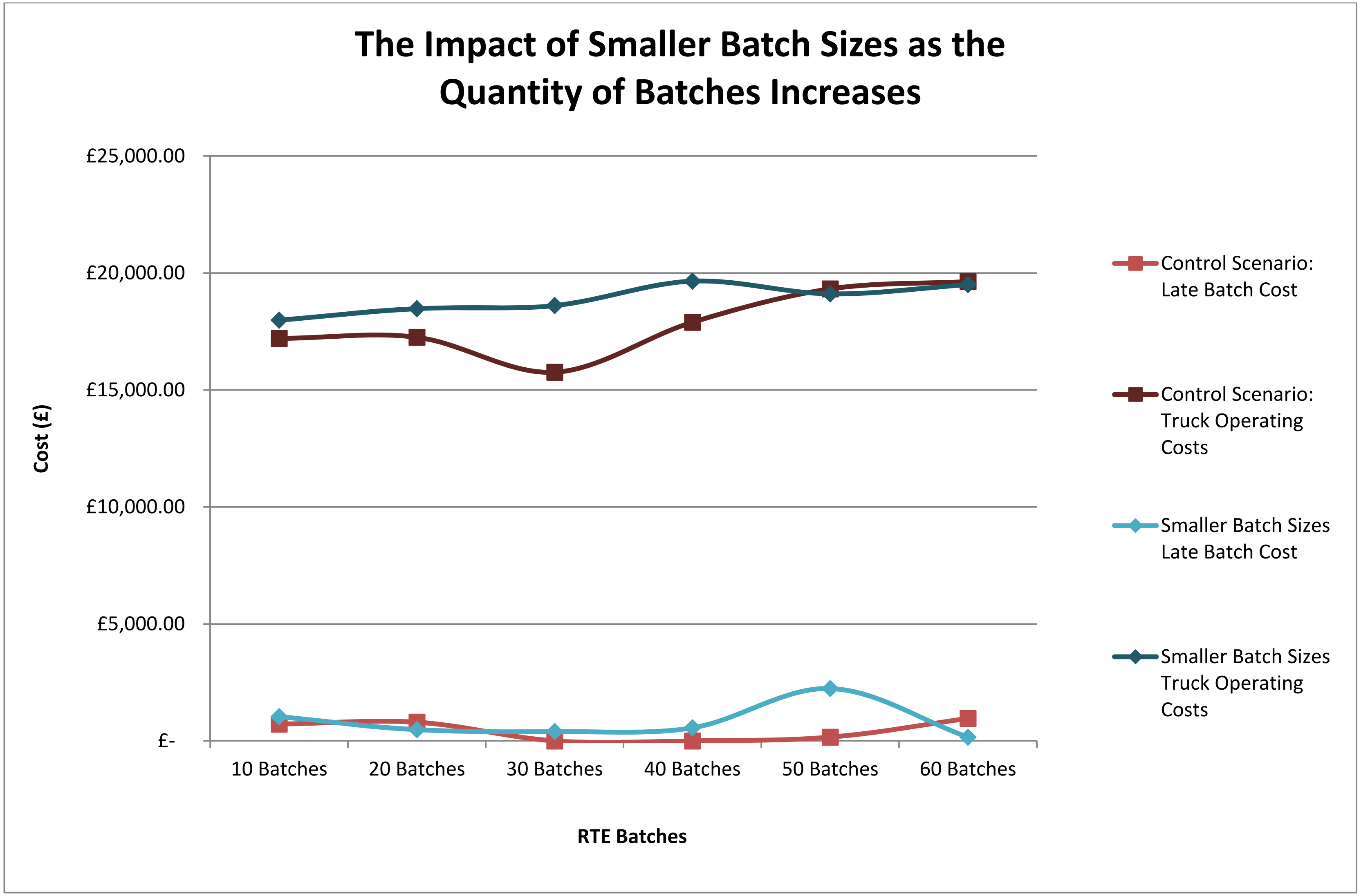


Figure 6.27 Impact of Implementing Smaller Batch Sizes as Batch Quantities Rise

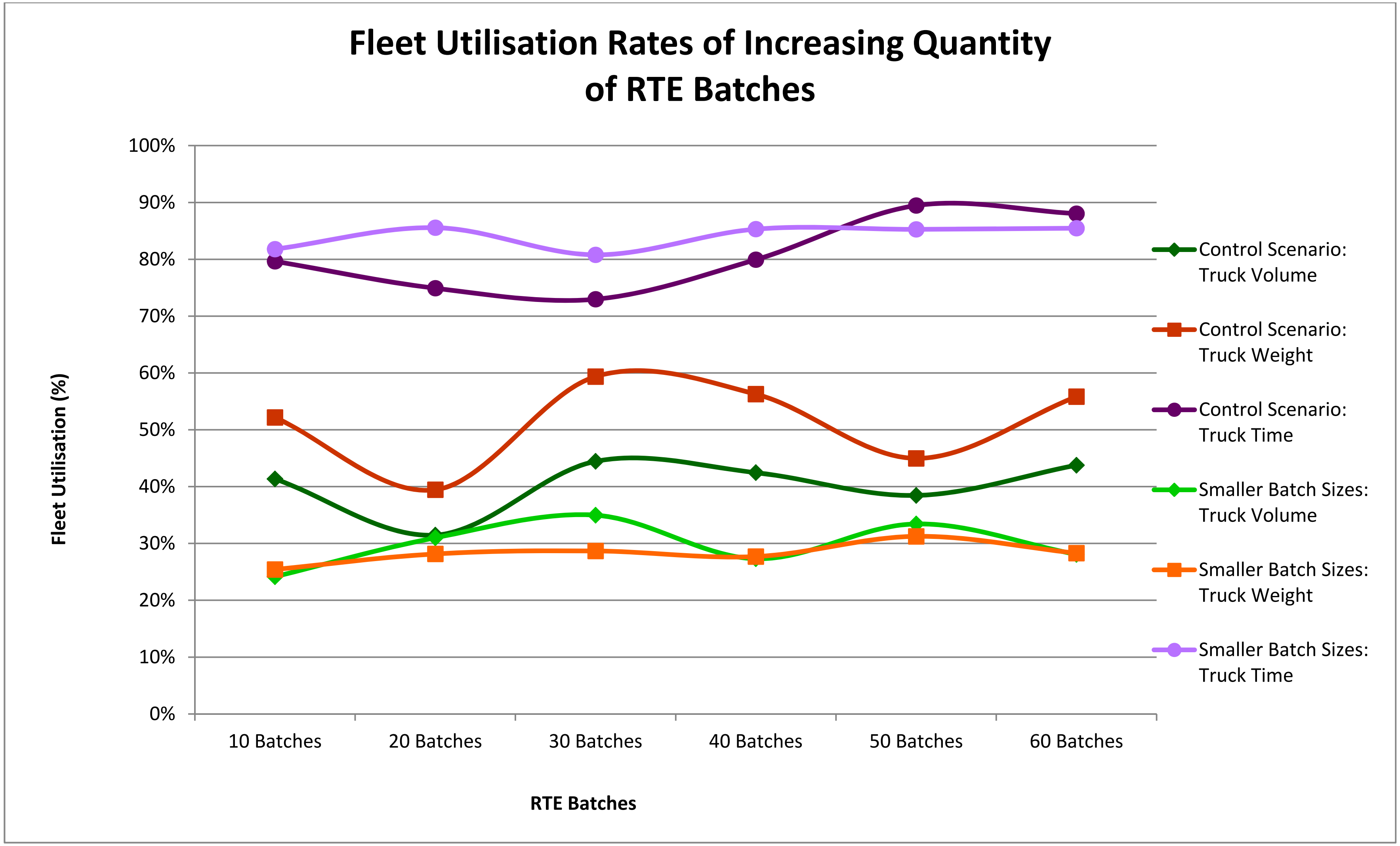


Figure 6.28 Fleet Utilisation Rates of Increasing Quantity the of RTE Batches with Smaller Batch Sizes

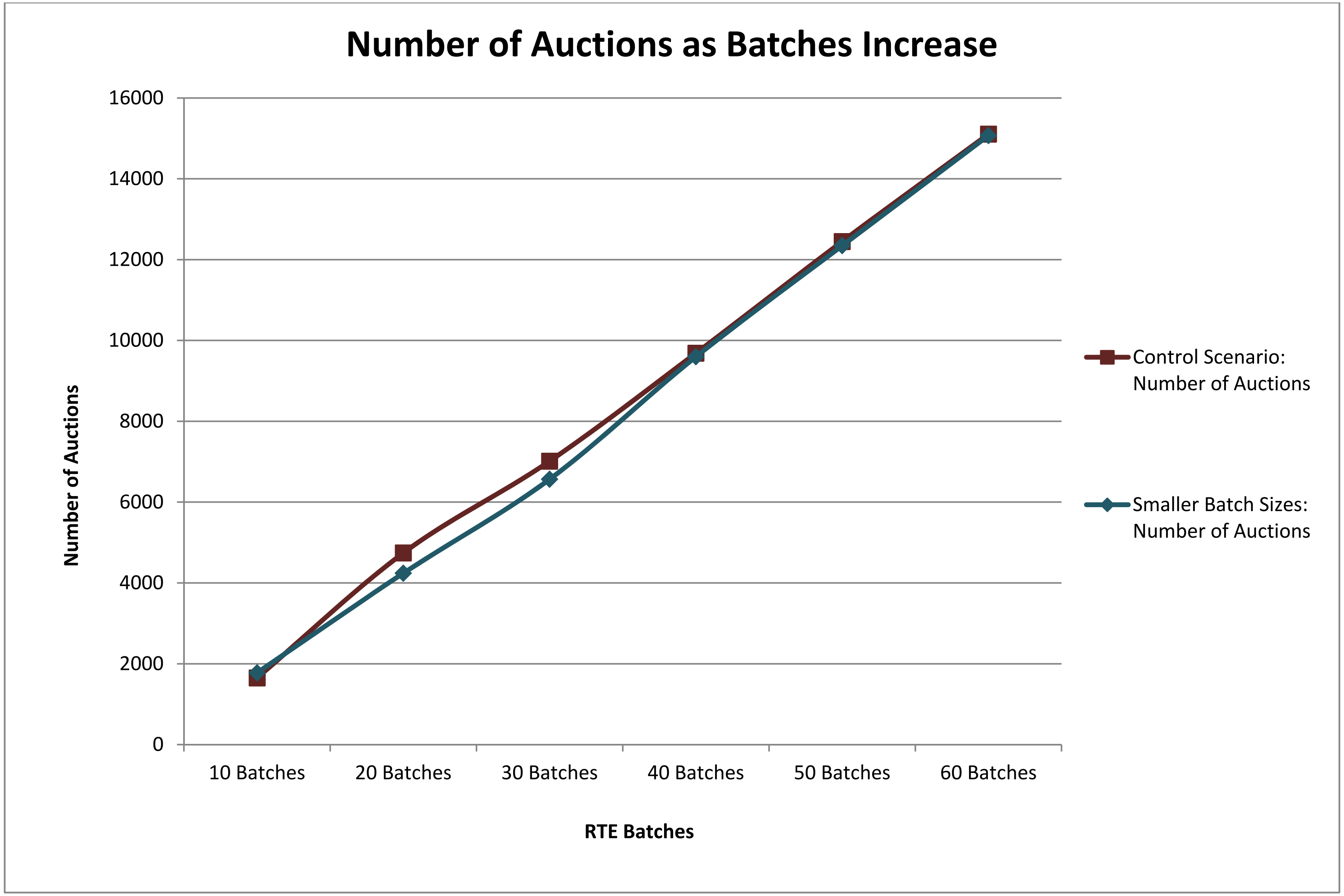


Figure 6.29 Number of Auctions as Batches Increase with Smaller Batch Sizes

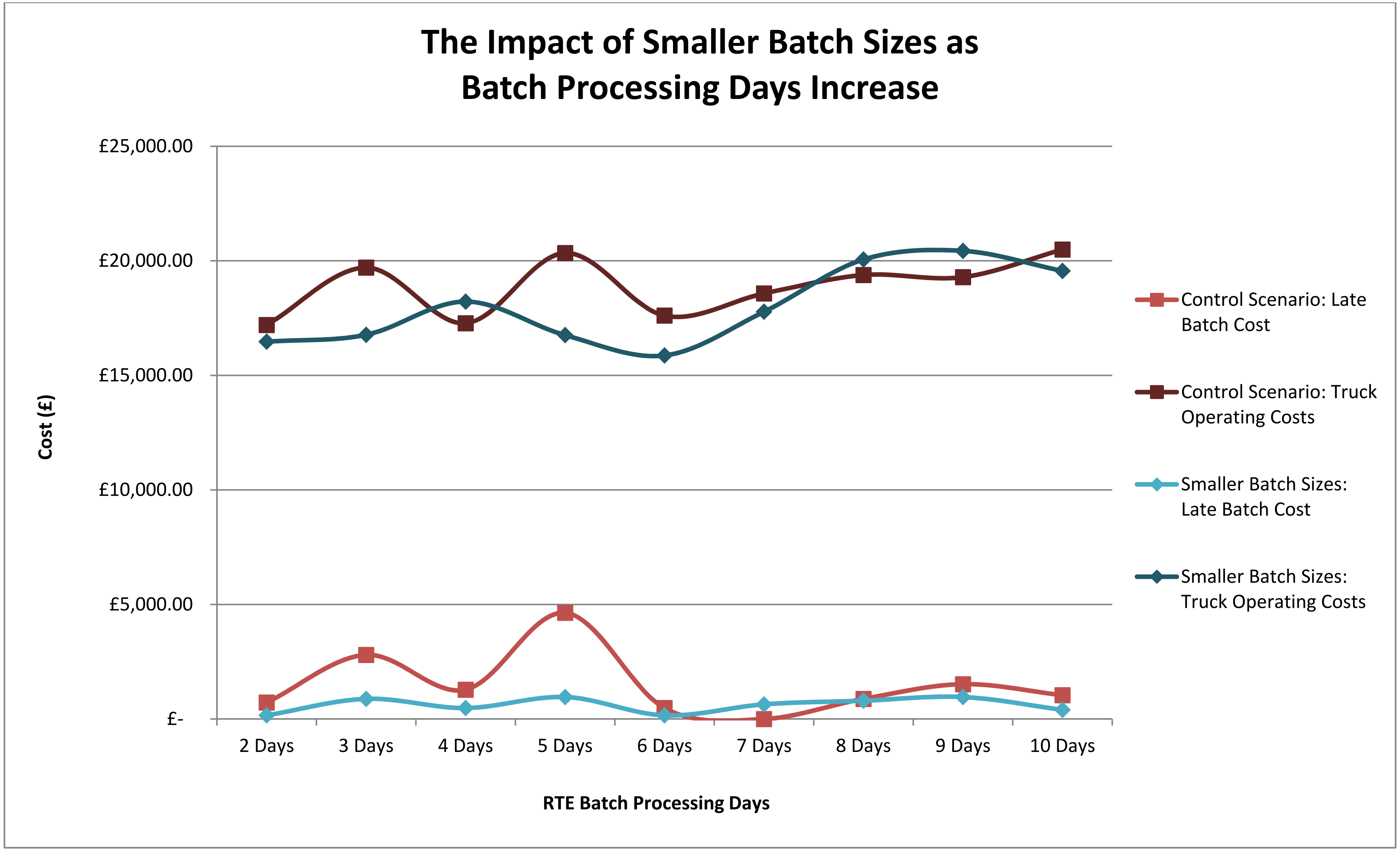


Figure 6.30 Impact of Smaller Batch Sizes as the Batch Processing Days Rise

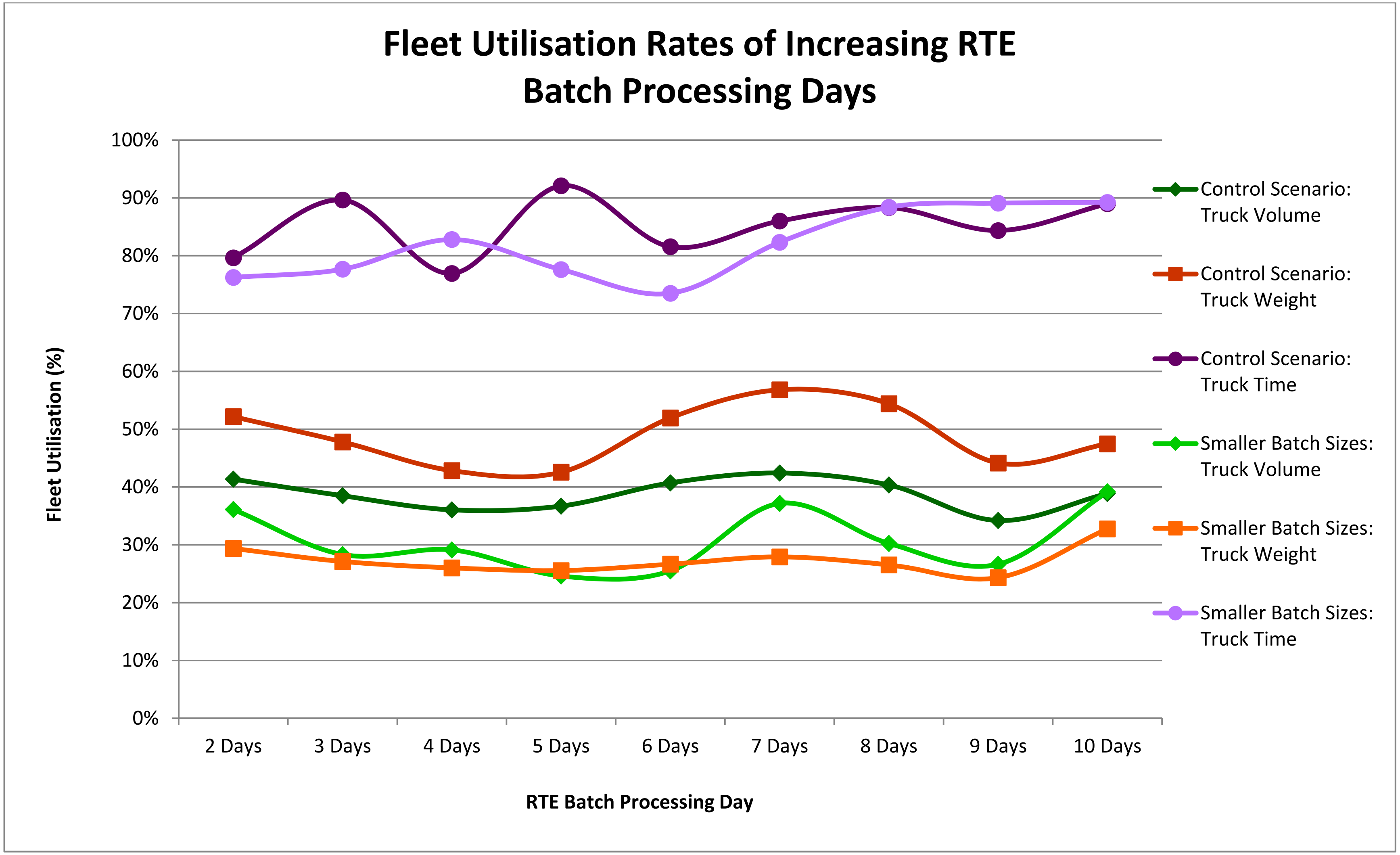


Figure 6.31 Fleet Utilisation Rates of Increasing the Batch Processing Days as the Quantity of Smaller Batch Sizes Increases

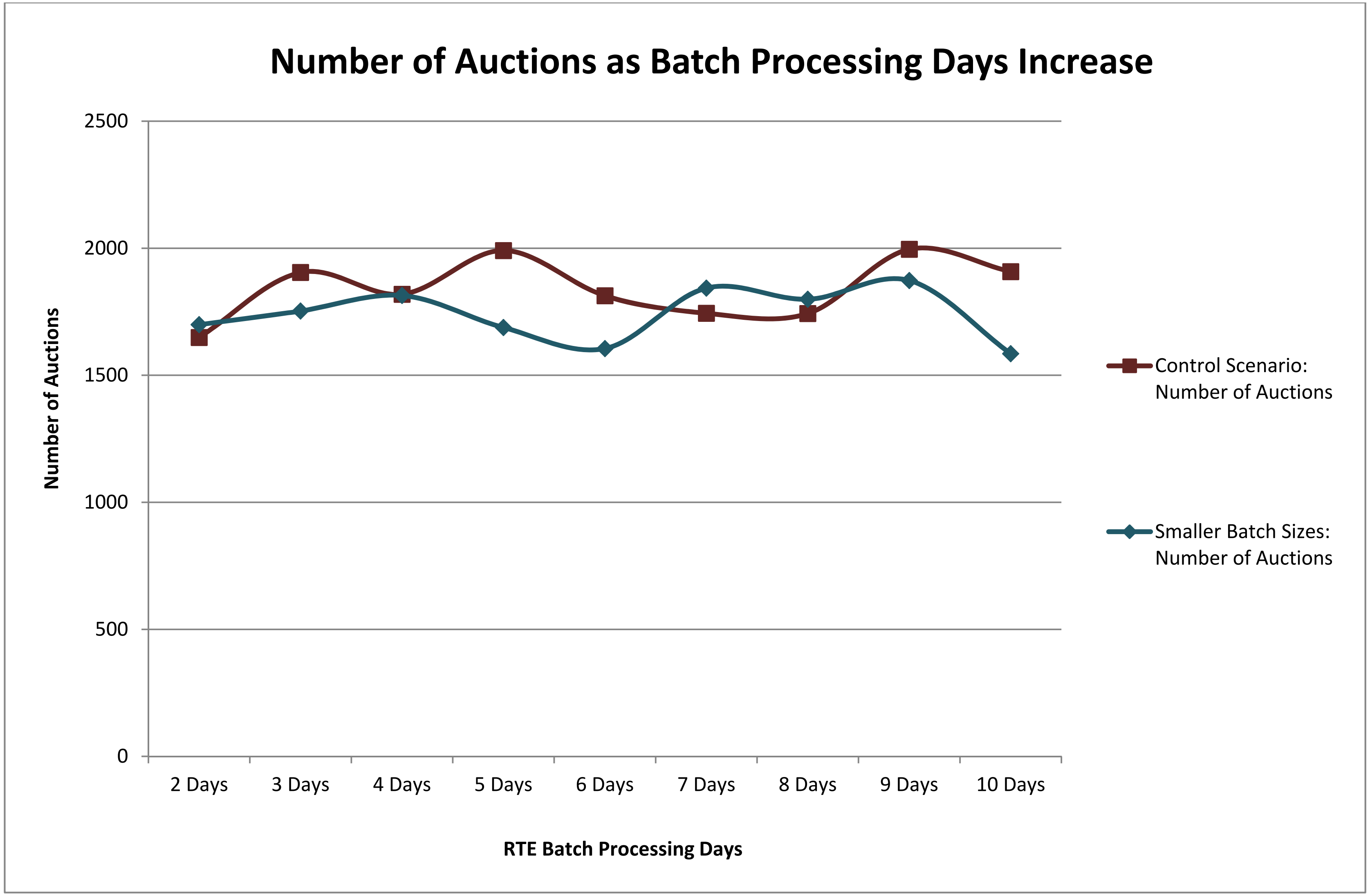


Figure 6.32 Number of Auctions as the Batch Processing Days and the Level of Smaller Batch Size is Increased

6.8 Bigger Range of RTE

The variety of products, which are carried by RTE may vary significantly in size, and therefore this section presents results of a bigger range of RTE with the parameters used illustrated in Figure 6.33.

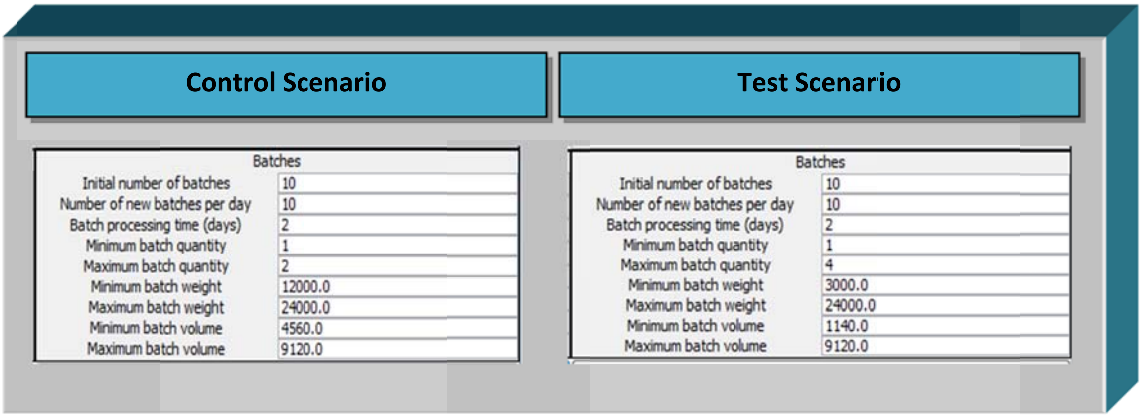


Figure 6.33 Control Scenario Versus Test Scenario

Figure 6.34 illustrates the impact of implementing a bigger range of RTE where it is noted that the late batch costs and truck operating costs are higher than the control scenario. In addition, Figure 6.35 demonstrates that the weight and volume utilisation rates for the bigger batch range are lower, whilst the time utilisation is higher. Therefore, although the fleet is used more, the number of late batches is higher, incurring higher costs compared to a network of a smaller batch range. Despite the difference, the number of auctions is similar as illustrated in Figure 6.36. There is a case to be made to limit the range of batch sizes in the system. This would be implemented in the rules, whereby a batch size between a certain range (for example, in the control scenario) needs to be reached, before it is put into the system. Therefore, a more standardised RTE network is favoured in regards to fleet and late RTE batch cost reductions.

Figure 6.37 illustrates that a lower batch processing day is favourable for a network with a bigger range of RTE as both the batch and truck costs are lower. Although the utilisation rates vary between 2 to 5 days, for the complex scenario as illustrated in Figure 6.38, the costs presented in Figure 6.37 indicate a better outcome for a lower utilisation rate. The auction rates in Figure 6.39 are similar in variation of 1600 to 2000, for both scenarios across the days. Therefore, although the late batch cost is not affected by a bigger range of RTE, the fleet operating cost is increased, as it is more challenging to develop a favourable schedule. Therefore, a contingency from the batches through introducing a constraint of a shorter batch processing day, aids in the reduction of the fleet costs, and in turn enables better utilisation of the RTE as they are returned to the facility quicker. For the final test case scenario in section 6.9, all the changes made in the previous sections are combined and tested to form a large and complex network with RTE variability.

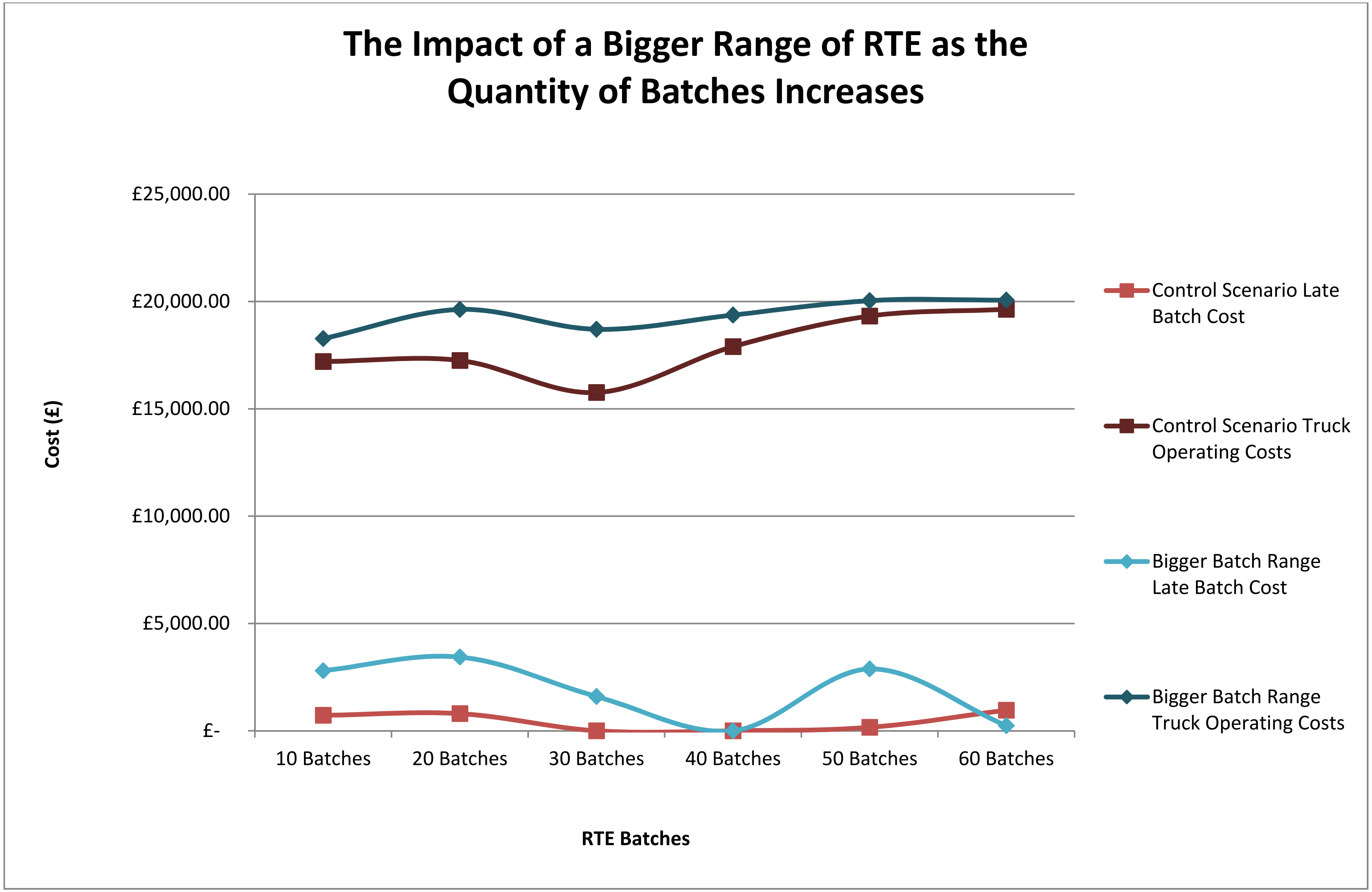


Figure 6.34 Impact of Increasing the Batch Size Range as Batch Quantities Rise with a Bigger Range of RTE

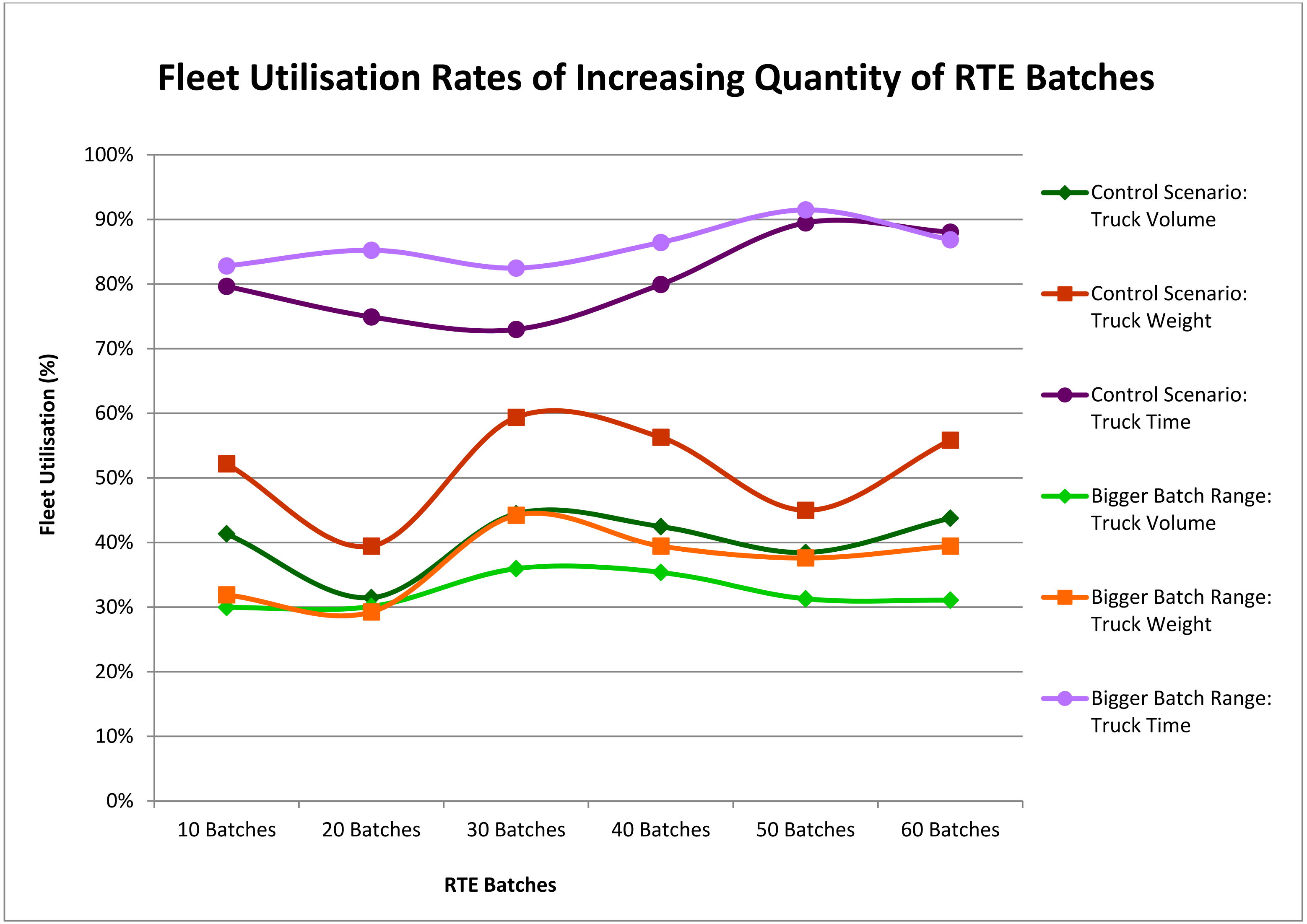


Figure 6.35 Fleet Utilisation Rates of Increasing Quantity of RTE Batches with a Bigger Range of RTE

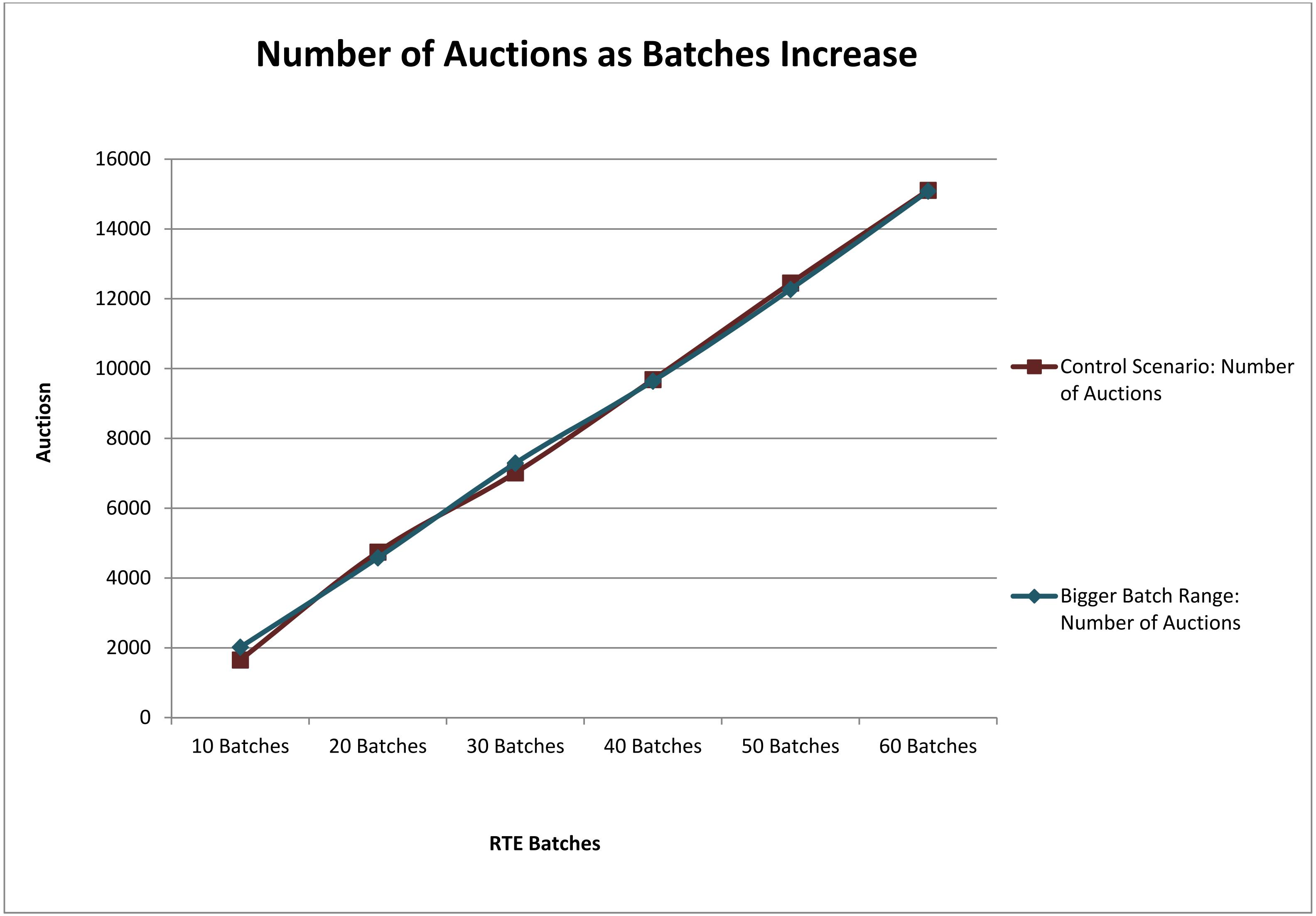


Figure 6.36 Number of Auctions as Batches Increase with a Bigger Range of RTE

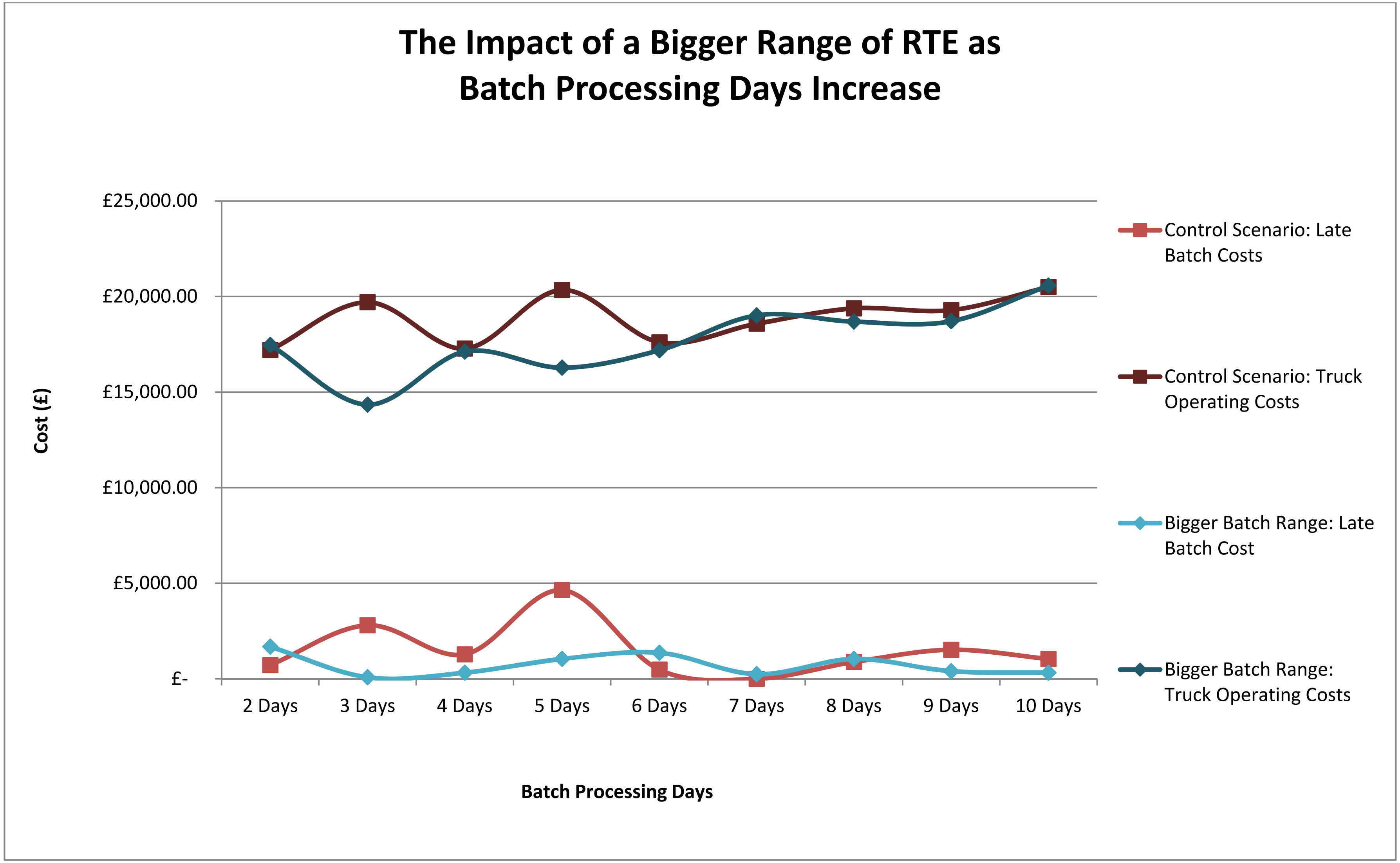


Figure 6.37 Impact of Implementing a Bigger Range of RTE as the Batch Processing Days Rise

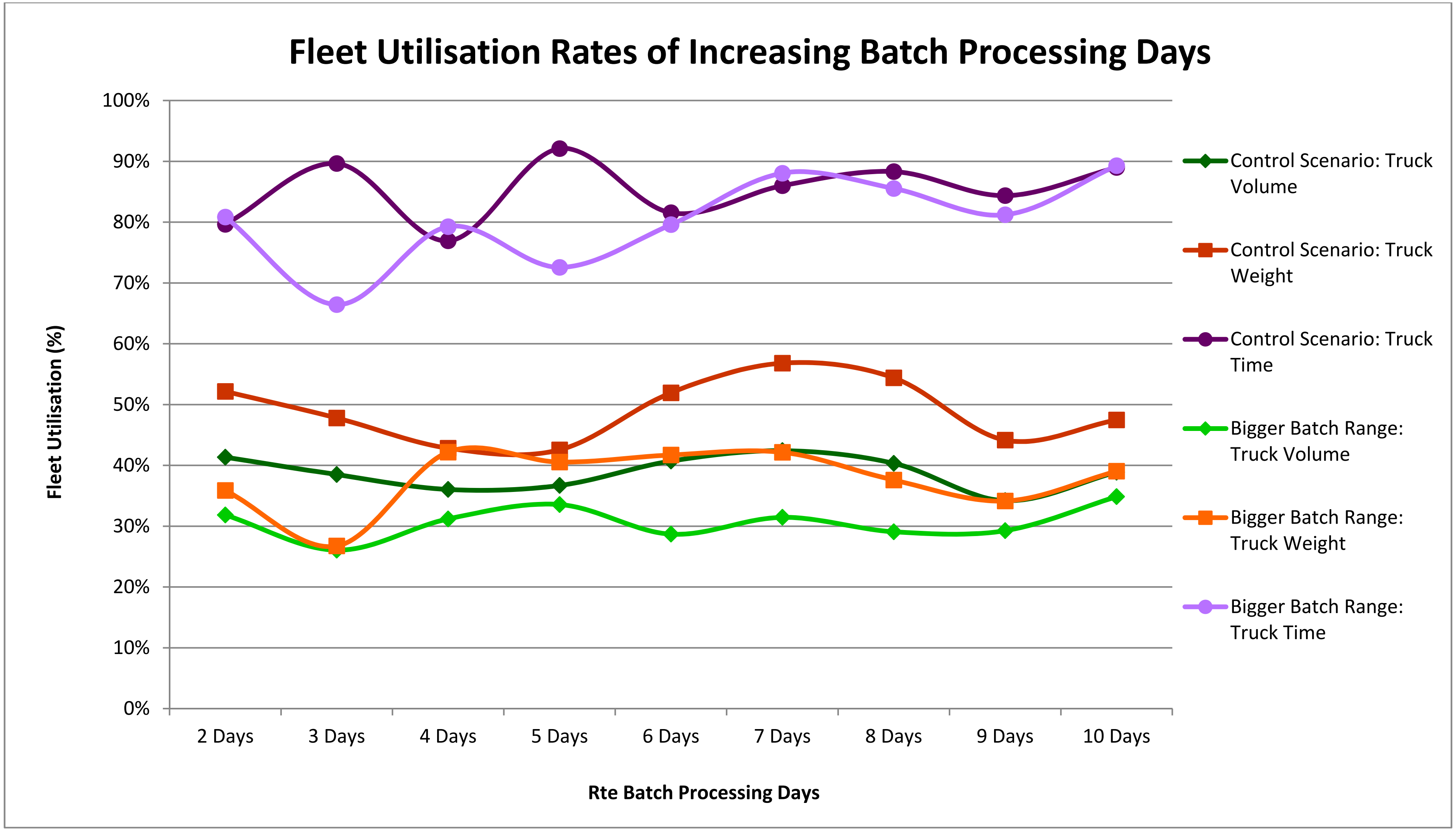


Figure 6.38 Fleet Utilisation Rates of Increasing the Batch Processing Days with Bigger Range of RTE

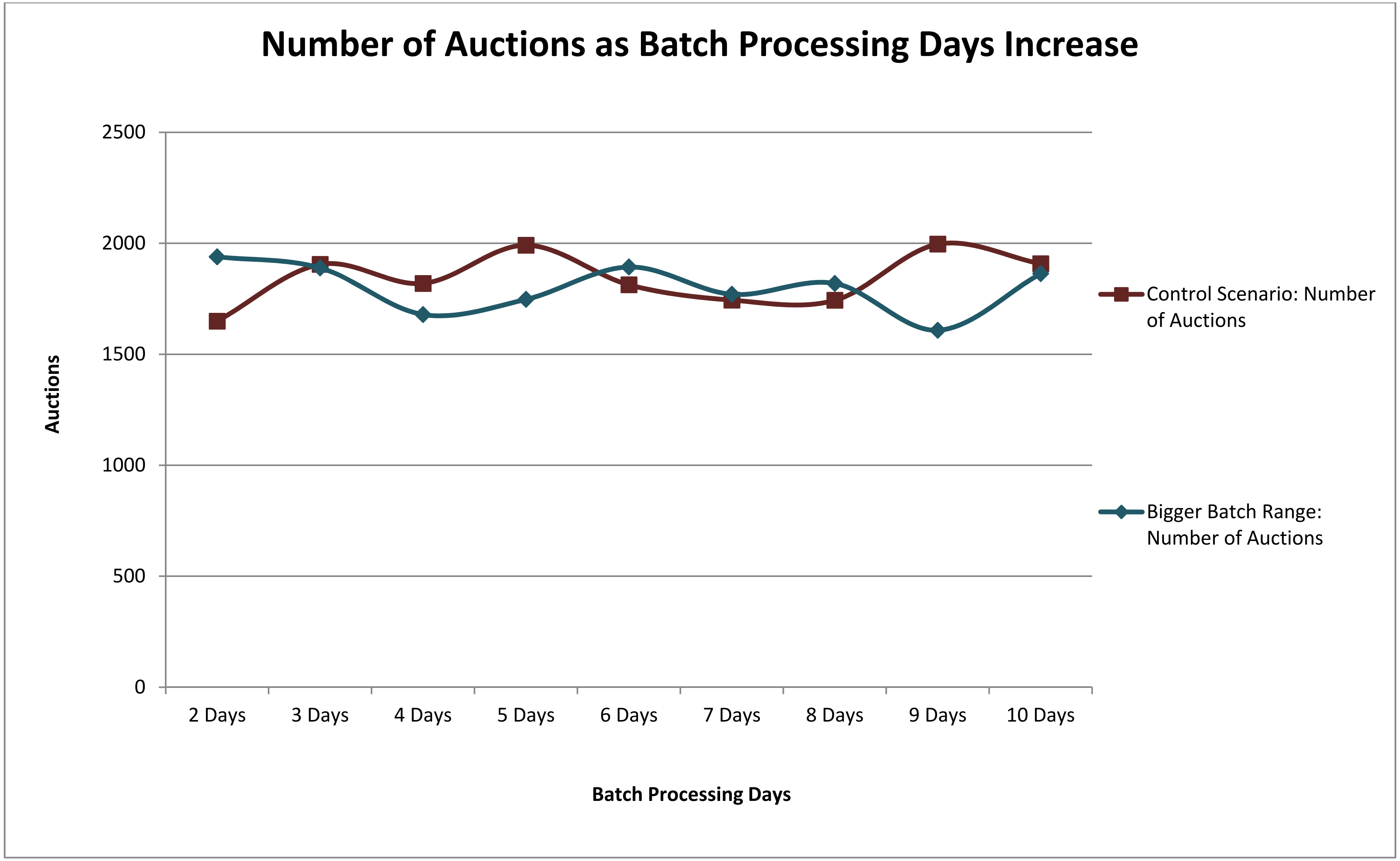


Figure 6.39 Number of Auctions as the Batch Processing Days with a Bigger Range of RTE Increase

6.9 Large and Complex Network

All levels of complexity, which have been described in sections 6.4 to 6.9 are combined in order to test the impact on fleet and RTE batch costs and rates. The network of the test scenario is illustrated in Figure 6.40 and the parameters used to set up the simulations in Figure 6.41.

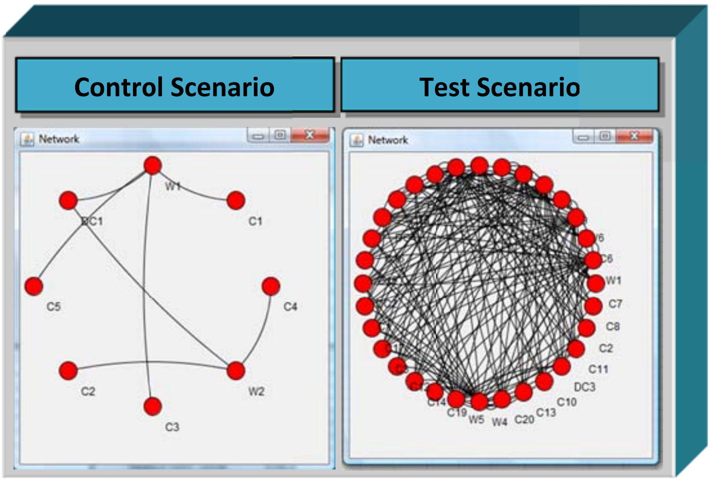
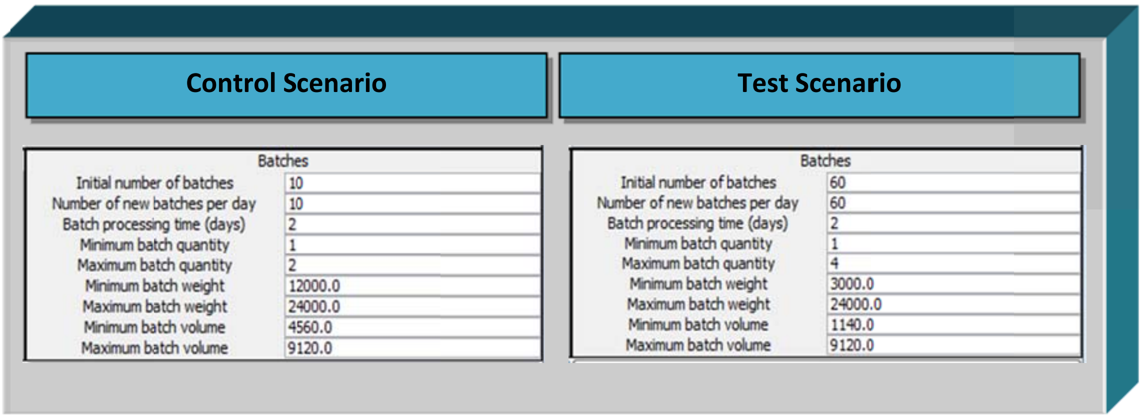


Figure 6.40 Control Scenario Versus Test Scenario



Control Scenario

Batches	
Initial number of batches	10
Number of new batches per day	10
Batch processing time (days)	2
Minimum batch quantity	1
Maximum batch quantity	2
Minimum batch weight	12000.0
Maximum batch weight	24000.0
Minimum batch volume	4560.0
Maximum batch volume	9120.0

Test Scenario

Batches	
Initial number of batches	60
Number of new batches per day	60
Batch processing time (days)	2
Minimum batch quantity	1
Maximum batch quantity	4
Minimum batch weight	3000.0
Maximum batch weight	24000.0
Minimum batch volume	1140.0
Maximum batch volume	9120.0

Figure 6.41 Control Scenario Versus Test Scenario

Figure 6.42 illustrates the impact of increasing the network size, network complexity and batch range. The late batch costs are slightly higher than the whole scenario but relatively stable, particularly when compared to the truck cost, which is higher for batches 20 to 40. This corroborates with the fleet utilisation time rate in Figure 6.43, which is higher for 20-40 batches despite the volume and weight utilisation rates being lower than the control scenario. In addition, the auction rates are lower for each batch quantity tested as illustrated in Figure 6.44.

The results indicate that all the trucks are required in order to lower the RTE batch late cost but there is a case to be made that as the size and complexity of the network increases and the batch range is higher, the truck sizes may be smaller as a quantity of trucks in the system will have more impact on quicker RTE returns and the adherence of fleet operating limits, than increasing the truck size.

When analysing the results of the batch processing days tested, for a complex network the costs are more stable then the control scenario as illustrated in Figure 6.45. In addition, the utilisation rates are also more stable; however a lower batch processing day is favoured for truck cost reduction as in many of the previous test case scenarios. In addition, Figure 6.47 illustrates a lower auction rate, as the limited truck driving hours are filled quicker than in a simple network.

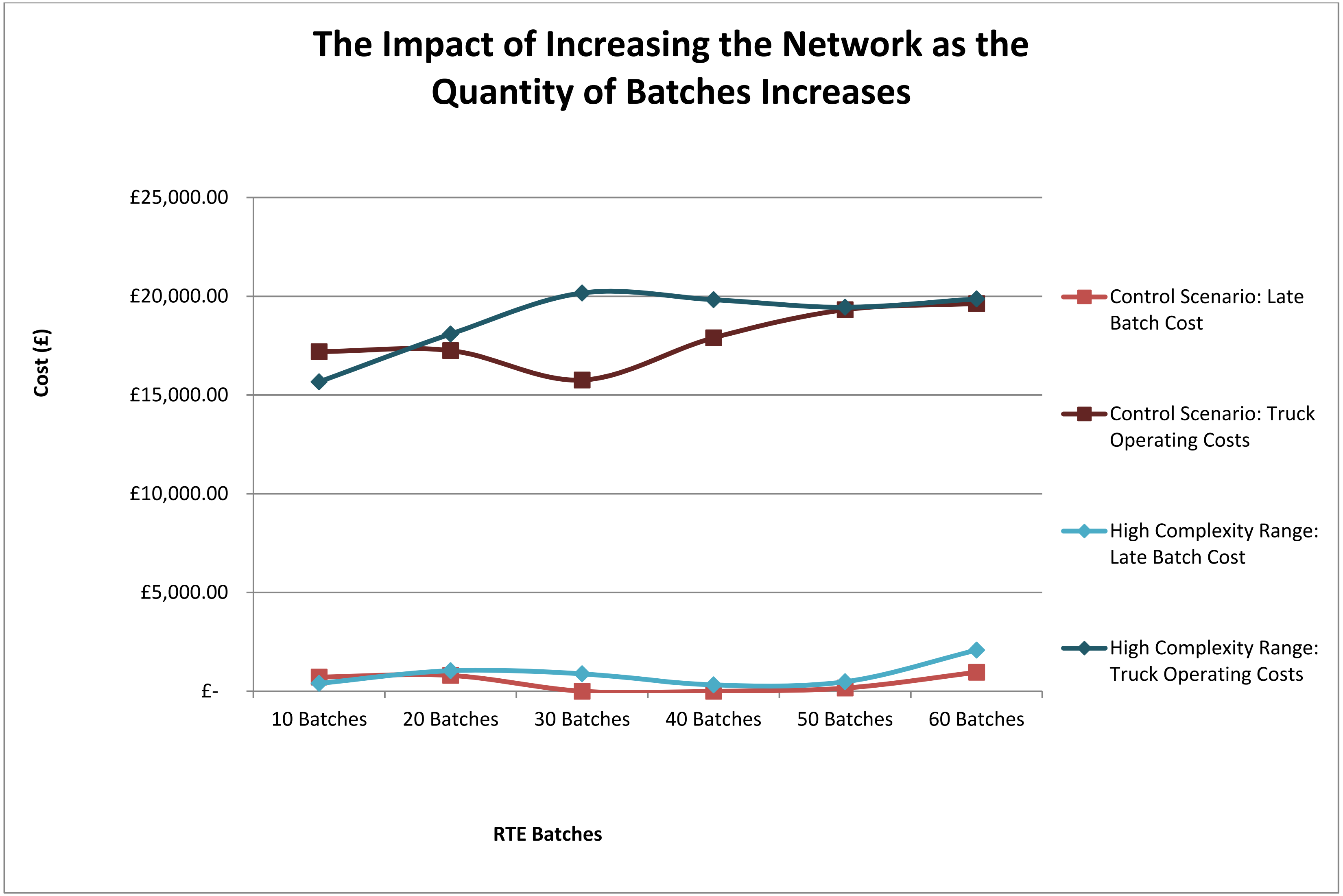


Figure 6.42 Impact of Increasing the Network Size as Batch Quantities Rise

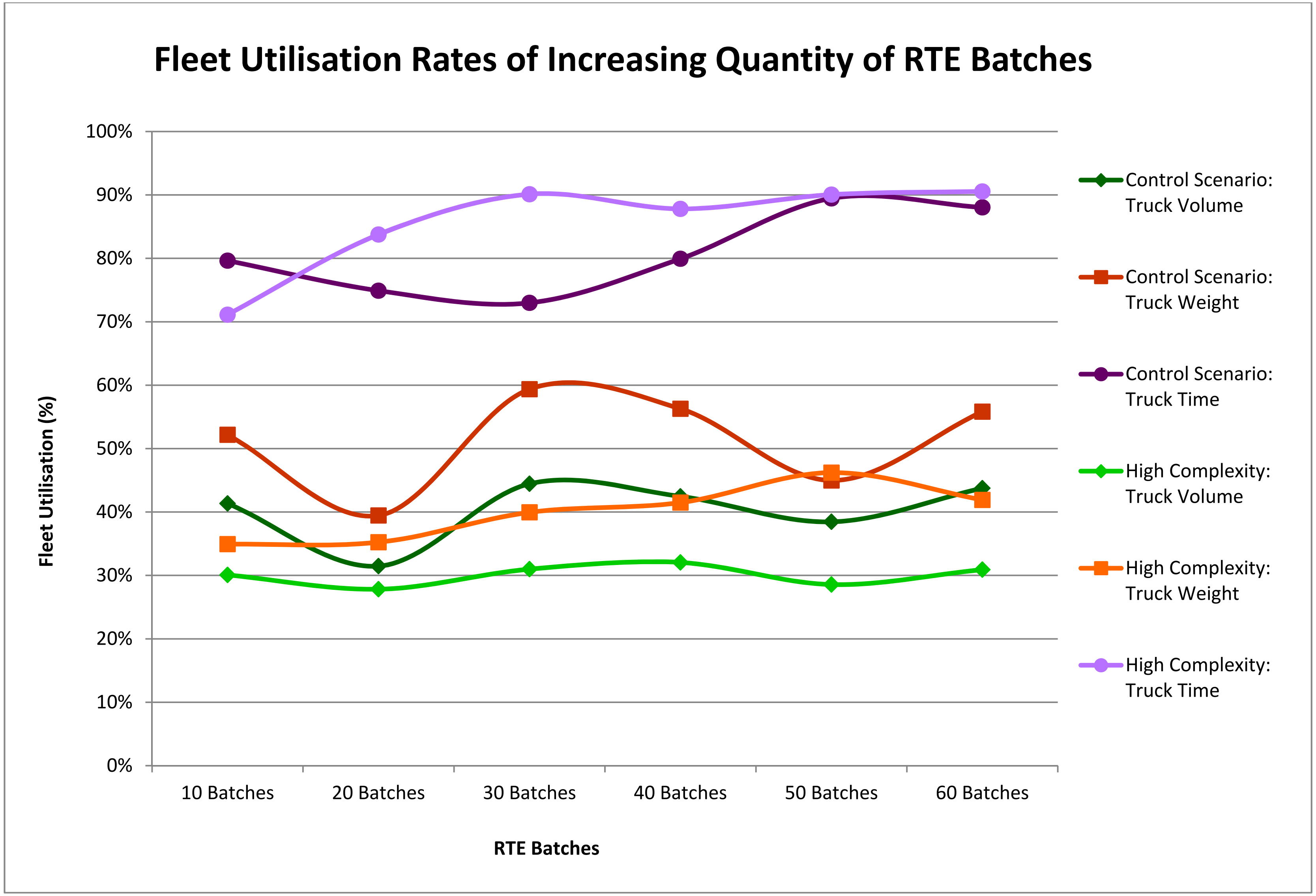


Figure 6.43 Fleet Utilisation Rates of Increasing Quantity of RTE Batches as the Network Size is Increased

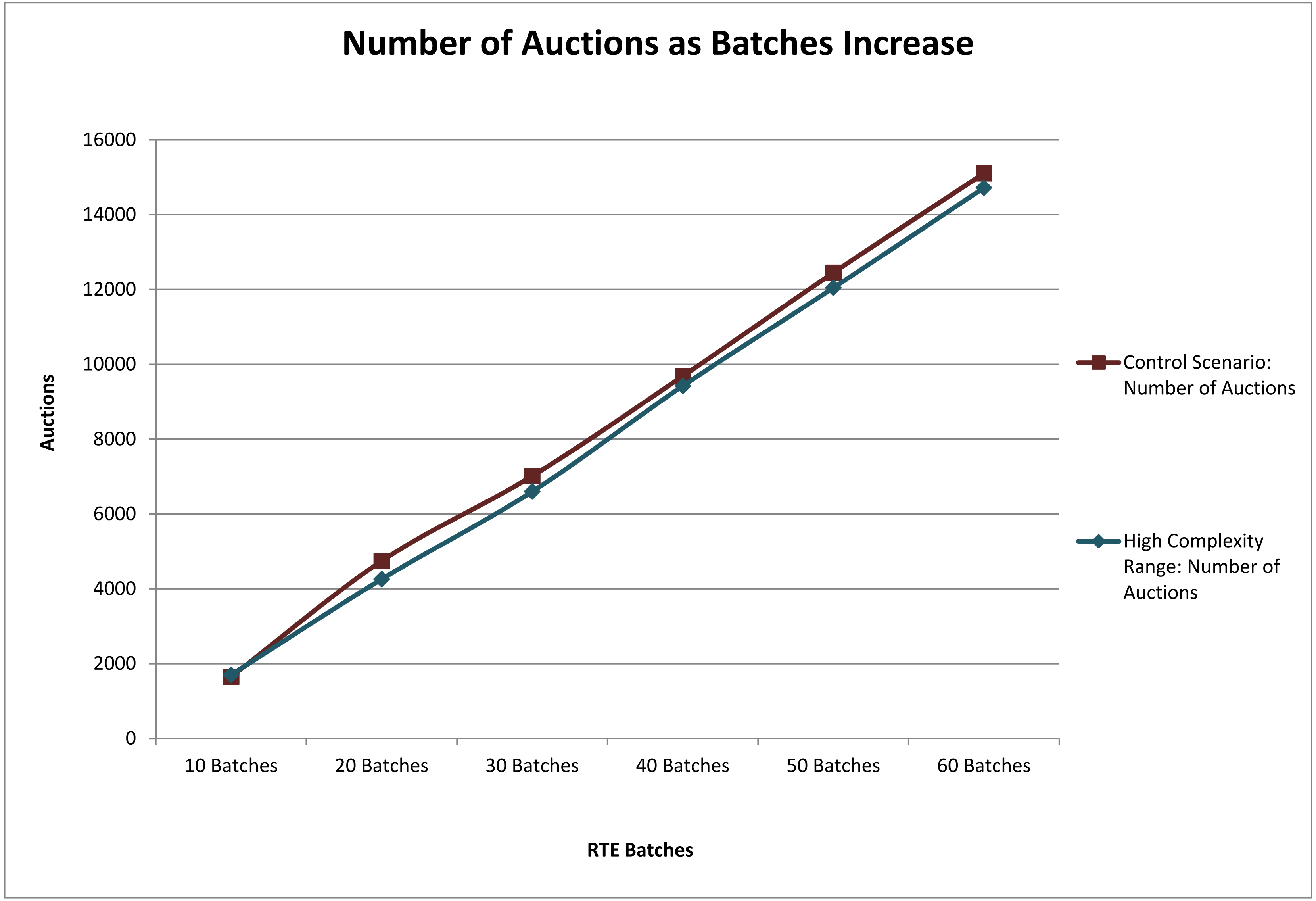


Figure 6.44 Number of Auctions as Batches Increase and the Network Size is Increased

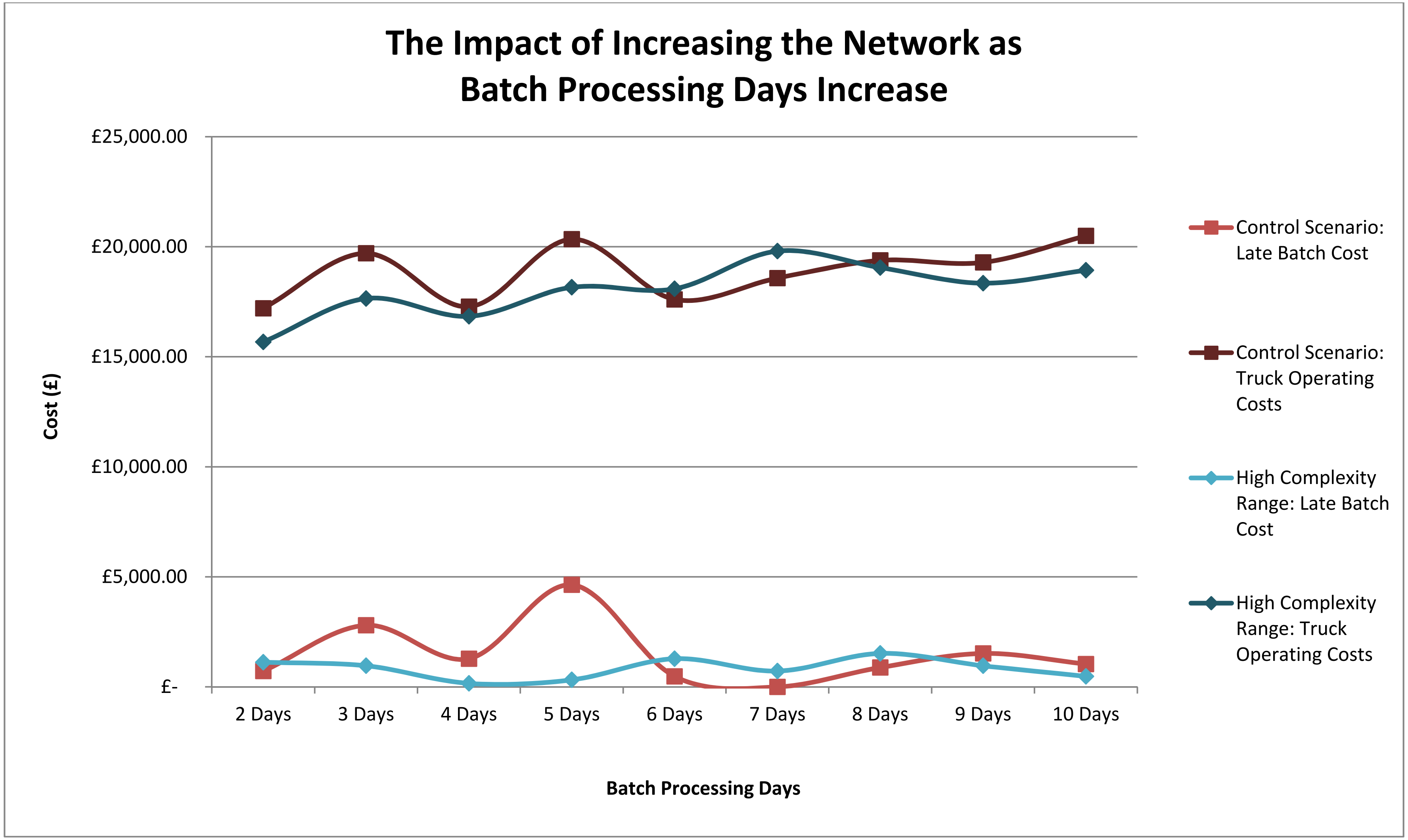


Figure 6.45 Impact of Increasing Road Complexity as the Batch Processing Days Rise with Increased Network Size

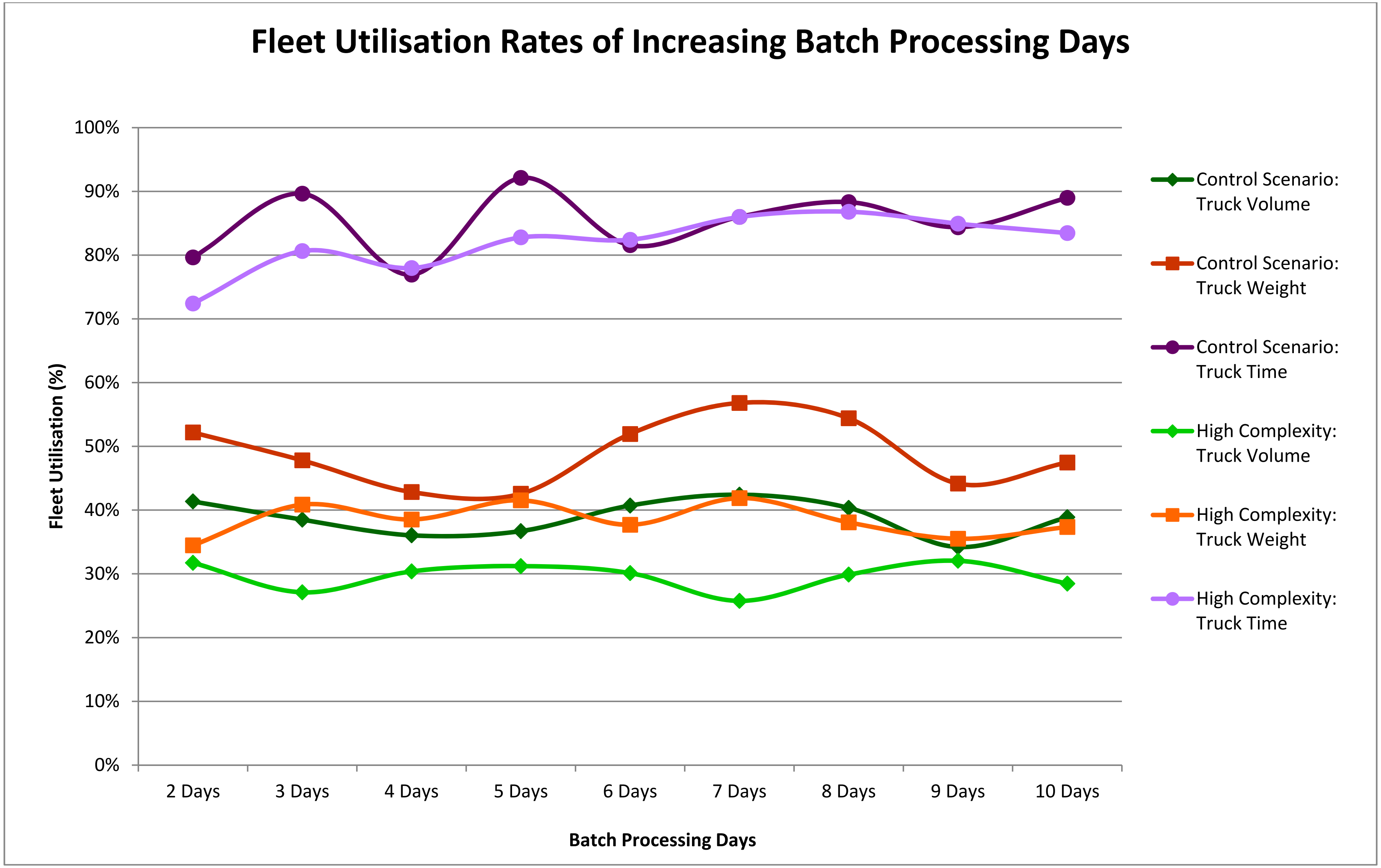


Figure 6.46 Fleet Utilisation Rates of Increasing the Batch Processing Days with Increased Network Size

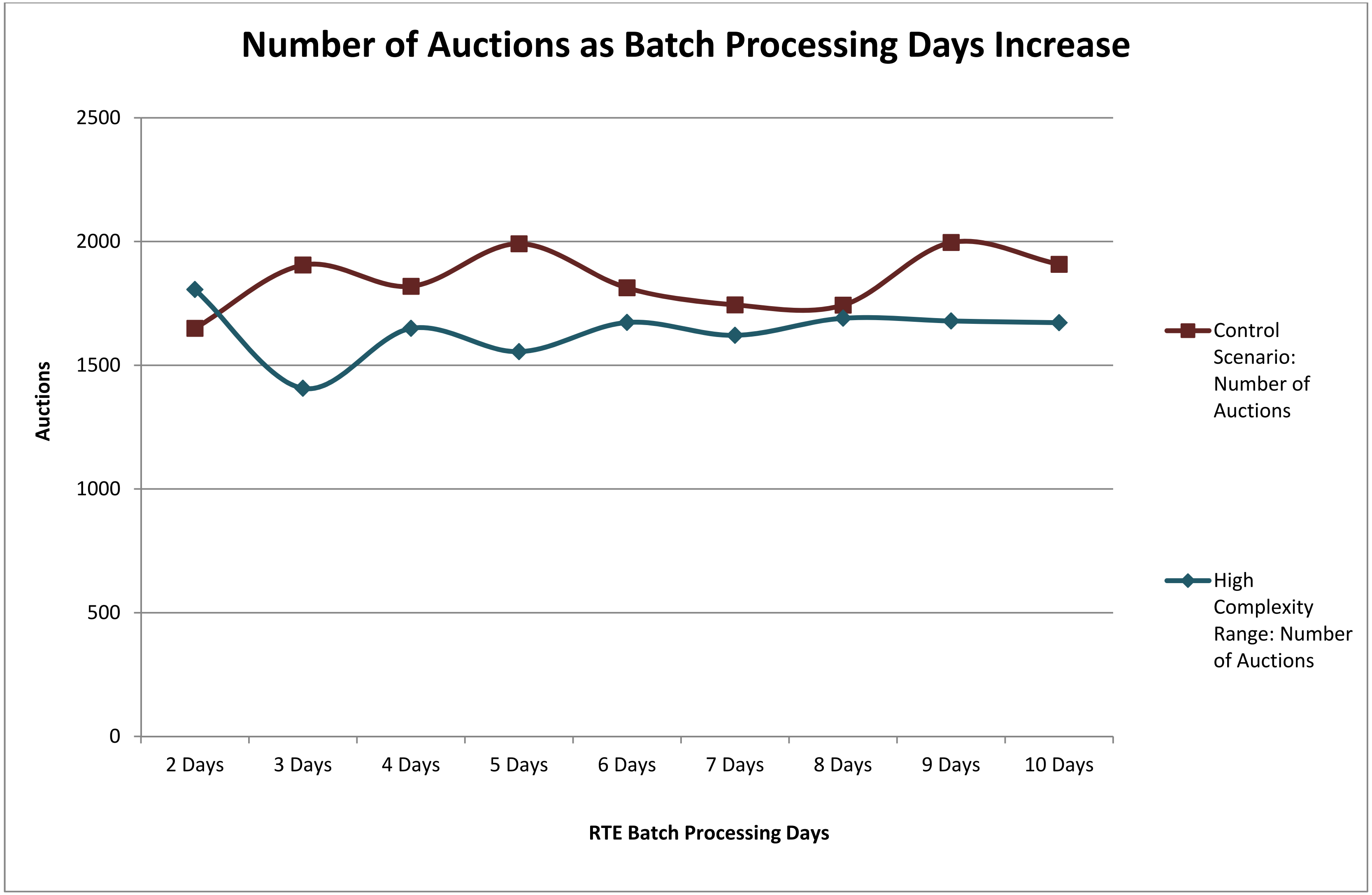


Figure 6.47 Number of Auctions as the Batch Processing Days Increase with Increased Network Size

6.10 **Summary**

The chapter presents results to test the competitive advantage objectives of the model discussed in chapter 5. The model developed focuses on utilising the current resources in the system to make improvements, and therefore the effect of gaining value advantage on costs is required. The model has various tools to monitor these effects and this chapter investigates the truck utilisation rates and the batch usage by exploring the impact of the auction process. The scenarios implemented in this chapter investigate the mechanics of the model and algorithms on the truck and batch utilisation rates. The simulation shows that the rules devised through the algorithms, which enable interaction between the Truck agent and RTE Batch agent, ensure all RTE are returned back to their final destination. The findings from the simulation are:

- The RFIDERTEAM ensures a quicker return and less cost loss of batch under-utilisation than a slow RTE return systems or networks with losses as reported by (Ilic et. al., 2009).
- Scenarios tested, demonstrate that certain batch processing days have a lower rate of late RTE batch costs than others and as the batch processing day is increased, the fleet costs rise.
- Offering trucks more variety of pick-up options (increasing the number of links between facilities), increases the number of late RTE batches.
- An increase of the number of facilities in the network will lead to a rise in the late RTE batch costs as the loads are spread across the network. In addition, the fleet has lower weight and volume utilisation as it picks-up smaller loads. Therefore, a strategy where the number of routes or the points of collection are limited, such as decreasing the number of facilities accessible, may be beneficial.
- The truck utilisation rates are significantly lower for smaller RTE batches as they are unable to pick-up enough loads within the time. Therefore, there is a need to reduce the truck size or increase the batch sizes in the scheduling process.

- A low batch processing day for smaller batches will lead to benefits for the fleet as the truck can schedule more efficient pick-ups, and therefore there is a case to be made to limit the batch processing day in order to implement a responsive fleet and quicker RTE return.
- As the range of RTE is increased, the fleet utilisation is decreased and the number of late RTE batches increases. There is a case to be made to limit the range of batch sizes in the system and standardise them to enable a better fleet collection and quicker return rate.

The simulations show that the set of rules devised results in efficient returns of RTE to increase utilisation and improves cost advantage for the RTE network. The model is also able to integrate forward and reverse flows through agents with negotiation capability, improving RTE management, whilst maintaining the fleet operating costs. The results demonstrate that the decentralised decision-making and interactions, which allows autonomy in the RTE management system, leads to a gain in competitive advantage throughout the network.

Highlighting the importance of the fleet in RTE management improvement, the truck is an essential unit to consider in the network as it incurs the majority of the total cost. The ability of the model to utilise the current fleet in the system is investigated through examining the weight, volume and time utilisation rates. These key performance indicators enable a valuation to be conducted as to the effect on the fleet of implementing agents, on the endeavour to integrate the reverse and forward flows. It also shows that the cost of late batches i.e. not using RTE are minimised, therefore creating a network improvement tool. Further research is required by using real-world networks for testing, and also a comparison to a non-autonomous approach without RFID for quantitative results, chapter 7 asserts the merit of this model in comparison to other systems presented in the literature.

Chapter 7

Discussion

In chapter 6, the RFID enhanced RTE Agent Model (RFIDERTEAM) was tested with different scenarios and the results of the simulations presented. In this chapter, a number of threads of research, which have been addressed throughout this thesis, are discussed. Aggarwal and Lim (2013b) assert a strong relationship between resource utilisation and RTE network competitive advantage, where it has been reported that introducing lean and agile methods is essential. Prior studies that have noted the importance of RFID technology application to RTE stress the significance of visibility as a critical feature as it enables competitive advantage achievement (Chuang and Shaw, 2007, Hellstrom and Saghir, 2007, Hellstrom, 2009, Ilic et al., 2009, Martínez-Sala et al., 2009). Through the enhancement of network visibility, the problems of RTE costs, shrinkage, ownership and cycle times are improved as described in section 2.1.3 and the DHL Exel Supply Chain case study in section 2.1.4. RFID technology when applied to RTE removes the need for manual handling and improves the efficiency of automatic identification as no line of sight is required when scanning tagged items. Shrinkage is reduced or even completely eliminated with RFID technology due to constant monitoring capability as stated in a study of Arla Foods (Hellstrom, 2009). Ownership problems in a LSP pool environment are alleviated by using the rewritable feature of the attached RFID tags, which automatically updates as the RTE changes between clients. The ease of keeping up-to-date records encourages an increase in the interchangeable rate of RTE between clients as the ownership of each RTE is clarified throughout the supply chain. Finally, with increased network visibility, bottlenecks are reduced, improving the RTE velocity, which leads to a reduction of the cycle times and a decrease in the RTE cycle costs, which in one study was reported at a cost reduction of 22%, assisting in the reduction of overall supply chain costs (Hellstrom, 2009, Ilic et al., 2009, Martínez-Sala et al., 2009).

As mentioned in the literature review, the increased supply chain visibility offered by RFID technology application is used to improve data retrieval and physical process automation. However, to utilise the RFID enabled visibility for more cognitive based methods, such as autonomous decision-making, further investigation is required. Little was found in the literature in regards to the potential of autonomous decision-making in the RTE problem domain. Furthermore, in reviewing the literature, there were limited studies on the impact of RFID data on creating autonomous objects and no studies addressing competitive advantage gain in the RTE problem domain in outbound logistics, from the perspective of the LSPs, where the RTE network becomes more complex and challenging as discussed in section 2.2.1. This study set out with the aim of assessing the impact of autonomous decision-making in improving the management of RFID enabled RTE. The scenarios used in chapter 6 to determine the impact of implementing an autonomous approach to create a leaner and more agile RTE network are presented in Table 7.1.

In this chapter, the merit of the research is established by interpreting the findings in section 7.1 and comparing the research conducted to other studies in section 7.2. Once the merit of the

RFIDERTEAM is established, the chapter assesses the limitations which may affect the conclusions presented in chapter 8 in section 7.3 and the chapter concludes with a summary.

Scenarios Tested
Comparison to RTE which return slowly
Comparison to a system with RTE losses
The batch processing day versus to truck operating costs
Impact of increasing the network complexity
Impact of using the model on a larger network
The effect of implementing smaller RTE batch sizes
Implementing a bigger range of RTE batches
Using the model to represent a bigger and more complex network with a larger quantity and range of batches

Table 7.1 List of Scenarios Used For Simulations

7.1. Interpretation of Findings

The results of this study show that agents provide the level of autonomy required to negotiate with flexibility, producing leaner schedules. When the results of simulations, which were conducted with scenarios using the RFID enabled Agent based RTE model (RFIDERTEAM) were compared to the RTE losses reported in the literature by (Ilic et al., 2009)**,** the model presented more favourable results as fewer late batches were scheduled, leading to lower late batch costs and a reduction in the total cost.

On the question of truck operating costs, this study found that with the implementation of the model, the RTE late batch costs are reduced, particularly when compared to RTE related costs reported in the literature, and therefore the fleet costs contribute to a large majority of the total cost. In cases where there were fluctuations of late batch costs, the total costs rise, and therefore the truck operating cost must be balanced against the return date of the RTE (returning the RTE quickly) as reducing the fleet cost will increase the late batch cost, and therefore in certain cases lead to an increase in the total cost, resulting in a trade-off.

It was found that when the model was tested with various scenarios, the agent interactions produced results, which enabled schedules to be developed, whilst the late batch cost is reduced. For example, when testing the results of the model against the RTE network presented in the literature, which is discussed in Chapter 6 in section 6.3, a significant reduction in the time to return batches is produced, reducing the late batch cost (see Figure 6.9). Other factors such as increasing the network complexity resulted in some test case scenarios to produce better utilisation of the fleet, whilst the RTE is returned back as soon as possible (see Section 6.5). These results support the mobility of agent interaction where the system provides an environment for the Truck agent and RTE Batch agent to negotiate the most favourable schedule. The ability of the model to cope with various levels of complexity, whilst reducing the late batch costs, asserts the model's robustness, whilst the complexity factor also affirms its applicability to the real-world.

The system provides an environment for the Truck agent and RTE Batch agent to negotiate a feasible schedule and this supports the benefits of agent interaction. The ability of the model to cope with various levels of complexity, whilst reducing the late batch costs asserts the model's robustness, whilst the ability to cope with an increase in the complexity factor affirms its applicability to the real-world. The model provides the flexibility to respond to dynamic changes such as truck constraints, enabling RTE network agility and the ability to factor in each client's needs and product requirements, such as differences in sizes and delivery dates.

The most interesting finding was that as the complexity factor of the network and scenarios were increased, the lowest batch processing day of 2 did not produce the lowest overall cost or the lowest late batch cost. For example, when the road complexity was increased, a batch processing day of 3 achieved the lowest level of late batches and truck operating costs when tested between a range of 2-9 batch processing days. Therefore, lying an extra day idle in the customer's facility or warehouse awaiting pick-up is more favourable in terms of overall cost reduction. This finding asserts that when scheduling trucks, RTE processing time is critical and requires investigation in order to assess the day, which will produce the lowest overall total cost.

Another important finding was that the higher the truck operating cost in each scenario the higher the time utilisation of the truck, however the volume and weight utilisation rates varied by batch load, batch processing day and the level of complexity in the scenario. Therefore, the model enables a truck to achieve time utilisation, and in the scenarios tested this is reached before the truck is able to achieve a higher load level. This is significant when the range and types of batches increase as the volume and weight utilisation is lower. Therefore, this stresses the challenges LSPs face as they have to transport an array of various RTE for different clients across complex and large networks. The RFIDERTEAM provides a platform for the LSP to test different scenarios, strategies and techniques in order to gain advantage for the LSP and their clients.

In the current study, comparing the truck operating costs, RTE batch costs and the utilisation rates, it is interesting to note that in all the scenarios tested, each factor varies in every instance. This is explained by the intricacy of each of these factors and the trade-offs, which are required in the RTE network. The contingencies through the RTE Agent Scheduling Algorithm (RTEASA) and the method and constraints such as the truck volume, perpetuate a trade-off action throughout the auction process. The bespoke agent model enables the agents to interact and negotiate the most appropriate solutions, whilst considering the trade-offs.

The model demonstrates that once a return date is set, on-time deliveries of RTE in the reverse flow is achieved with the RFIDERTEAM. Therefore, the model successfully integrates the scheduling of the reverse deliveries with the forward delivery requests simultaneously through the auction process to produce a feasible schedule. The data gathered for the scenario shows how intensively the bidding and auction process is with a high rate of auctions. This highlights that each step of the contingency is used and is essential in ensuring that all the reverse deliveries are integrated during the scheduling process, which enables a better utilisation of the truck time as new legs are added in the reverse flow.

An unanticipated finding was that there were certain batches, which were not scheduled and the algorithm would not select them due to the time and dimension constraints, hence making the batches undesirable. Although the number of undesirable batches is low and are collected at some point, further development could explore an incentive to pick-up undesirable loads before they are late. It was noted that certain network conditions such as increasing the number of links between facilities, improved the RTE batch pick-ups, therefore there is a case to be made to implement more route options for undesirable batches, preventing late deliveries. However, increasing the network complexity increases the fleet costs but the cost rise may be alleviated by lowering the RTE batch processing day. Another possibility is to standardise the size of the RTE batches. For example by amalgamating more batches together before the auction process, the batches become more desirable due to their size. The current algorithm works to split the batches during auction in order to secure a delivery, however based on the findings it would be interesting to observe the impact of increasing the batch size. Any conditions which are imposed must balance the trade-offs of cost increases in other areas, and therefore fleet costs would need to be considered in any simulated scenario.

The total cost calculated in the simulations conducted in Chapter 6 is a combination of the late batch cost, which is the number of days an RTE is un-used when it could be utilised and the truck operating costs. The time, weight and cost utilisation rates of the fleet are also monitored. The model facilitates the potential to run a variety of scenarios, which aids the LSP to determine the level of trade-off required between the costs and utilisation rates for each client and each RTE batch. This finding, while preliminary, suggests that in order to select the most appropriate level of trade-off between costs and utilisation rates of the fleet and RTE, the selection of the batch processing day is critical as illustrated in Figure 7.1. The batch processing day reduces both the late batch costs and the truck operating costs, if the appropriate day is selected based on the simulations conducted in the model. A reduction in the late batch cost will reduce the amount of time a piece of RTE awaits idle, increasing the utilisation rate, and therefore the return on investment for each piece of RTE is enhanced. The overall fleet utilisation rates are increased due to a combination of a reduction of the truck operating costs and an increase of RTE batch pick-ups during the reverse flow. These findings assert the significance of the batch processing day in reducing waste by removing under-utilisation of the RTE and fleet and reducing costs, which creates leaner RTE networks.

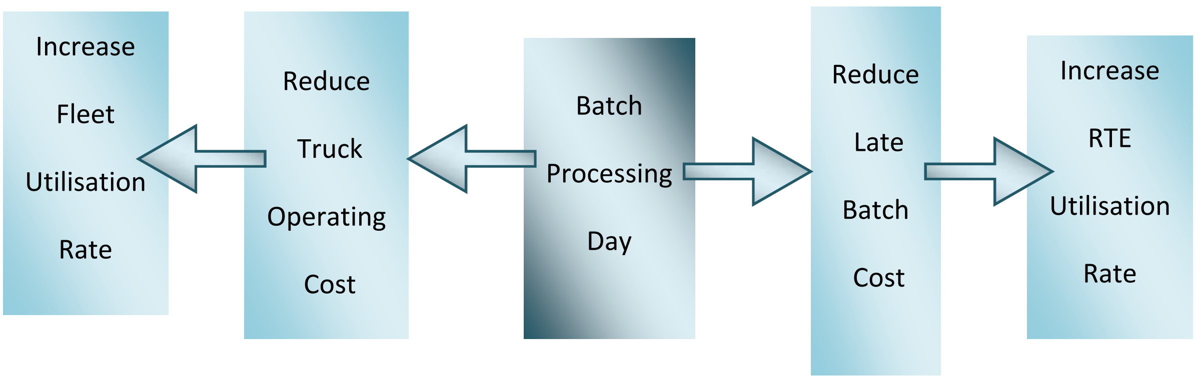


Figure 7.1 Significance of the Batch Processing Day in RTE Improvement

With the set of scenarios, which have been tested caution must be applied as these findings might not be transferable for every scenario, and therefore cannot be extrapolated to all RTE networks. However due to the strong correlation between the fleet, batch costs, utilisation rates and the significance of the batch processing day, it can conceivably be hypothesised that the agent interactions and bespoke algorithms developed for scheduling in the RFIDERTEAM offers insight into the RTE network and enables a reduction in overall network costs, improving the management of RFID enabled RTE and enhancing the competitive advantage of LSPs.

7.2. Comparison to Related work in the RTE Problem Domain

Ilic et al. (2009) stresses the lack of simulation models for RTE management improvement arguing that in the literature the only work, which has a model or simulation methodology for RTE management is by Kroon and Vrijens (1995) and Johansson and Hellstrom (2007), whose work is discussed further below. RFID application is argued by Ilic et al. (2009) as an essential technology to enable visibility to improve the RTE network, which is proven by the model and simulation presented. However the model is developed in order to do a cost analysis and cannot take various inputs and make decisions utilising real-time data from RFID enabled RTE unlike the RFID Enhanced RTE Agent Model (RFIDERTEAM), which uses the modelling and simulation process to develop a decentralised decision-making platform for RTE networks in real-time.

Johansson and Hellstrom (2007) use a discrete-event simulation (DES) model in order to examine the effect of asset visibility on RTE management. In this model it is not possible to observe any changes, and also any discrepancies are not well represented, for example when a server broke down in the case study discussed in their paper. In addition, the model cannot measure the effect of crucial parameters unlike the agents, which have been developed in the RFIDERTEAM. For the eight parameters, which are considered by Johansson and Hellstrom (2007) the fleet, which is the most important feature to implement as there is a high correlation between the effect of the fleet and the RTE costs as demonstrated by the RFIDERTEAM in chapter 6 and as discussed in section 7.1., is not included. In addition, the assumptions made in the DES are in a linear fashion, which is unrealistic in regards to representing the dynamism of the real-world RTE network. The research conducted by Johansson and Hellstrom (2007) is focussed mainly on the distribution centres despite transportation being an essential and costly part of the RTE network management process, hence the RFIDERTEAM developed integrates the transportation of RTE, and therefore a direct comparison cannot be made with this model. However it can be stated that the RFIDERTEAM provides a contribution to knowledge in the RTE domain as it captures the dynamism between the RTE and transport network.

Kroon and Vrijens (1995) present a quantitative model, which is used to aid in the planning process for a RTE management system. A real-time feed of data is not integrated into the model and simulation process, and therefore numerous assumptions are made, which do not enable the model to represent the dynamic nature of the RTE network. In addition, the model is developed using one particular test case, which means that it has a low level of transferability, and also the model's ability to respond to various changes has not been tested, which does not enable it to foster agility. The RFIDERTEAM on the other hand is developed from the "bottom-up" with agents representing each object, which react to real-world changes, creating a higher level of realism in comparison to the work conducted in this field thus far.

A comparison to previous studies affirms the importance of developing modelling and simulation based solutions in the RTE management problem domain and the research conducted and presented in this thesis stresses the contribution agent based modelling in conjunction with RFID technology brings to the decision-making process throughout the RTE network.

The studies described above are not fine-tuned to the RTE management problems whereas the RFIDERTEAM developed uses agents to represent objects in the network enabling the detail to be captured better. A more detailed and accurate representation of the RFID enabled RTE network allows the model to be robust, which is further strengthened by the model's ability to represent extreme cases and complex scenarios more adequately than the studies discussed. The RFIDERTEAM represents the problem sufficiently enough to capture the dynamic nature unlike the models above, which only function in controlled environments, and therefore outside of this will struggle to represent the real-world. The dynamic environment, such as the RTE network will have many stages and states, which the RFIDERTEAM captures and uses agent interaction to produce a feasible solution.

The modular nature of the agent based model through the object representation enables complex object orientated programming as it is an object friendly way to model a system. A model built through a non-agent representation is incompatible with object-orientated programming and the modular form of the provision for a solution, which enables localised decision-making, becomes inaccessible. Any non-agent representation would therefore require an intermediate component in order to develop advanced modular RTE network representation such as the RFIDERTEAM, which would be cumbersome, complex and unable to capture the detail and flexibility implemented though agent orientated programming. The representation of the RTE network through objects captures the detail and creates interactions, allowing the behaviour to be monitored. Agents enable the integration of reverse flows into the scheduling process and the utilisation of real-time data creates an information-centric environment for agents, which enable effective and realistic decision-making adding a valuable contribution to knowledge.

7.3. Limitations

One of the main limitations is the lack of equivalent models to compare the RFIDERTEAM for the specific RTE problem domain described, particularly as bespoke algorithms have been developed specifically to integrate the reverse and forward flows in order to ensure that all the return flows are scheduled on time. As the fleet cost and truck operating functions have not been implemented by the studies discussed in the previous section, a comparison is not possible. The important features to consider are the robustness and dynamism of the model based on the scenarios which have been implemented and tested, and have proven to integrate the reverse flows into the scheduling process, enabling the RTE to return quicker for re-use, increasing utilisation.

There are a few limitations in the current RFID enhanced RTE Agent Model (RFIDERTEAM) firstly, due to the lack of availability of real-world data from project partners DHL, the data for scenarios are generated using random numbers. The model implements a nominal set of costs in order to test the model with different scenarios as presented in Chapter 6. The costs can be modified, and therefore the workings of the model will remain the same as real-world costs are implemented but the output of the balance between the importance of the fleet or RTE batch return rates may change. With the scenarios tested, for each day a batch is not being used when it could have been, is given the same weighting as a full days cost as a truck operating cost. If this ratio changes, for example, if the cost of not using a batch is higher (expensive bespoke RTE) then the results will indicate a more significant impact on the late RTE batch costs across all the scenarios tested, and therefore initiatives to return RTE back quicker, such as low batch processing days, would become favourable. On the other hand, if the cost of the fleet increases a strategy to limit the amount of pick-ups may be required, and therefore the late batch costs would rise and the RTE awaits pick-up. Future research can incorporate the data from the real-world in order to compare to the test scenario developed and enable further understanding as to the RTE scheduling process for specific networks.

Secondly, in the test case scenario all the trucks in the simulation are of the same size in order to offer the simplest solution and allow comparisons to be made between test cases and examine the ability of the algorithm and model developed. However the methodology selected allows a range of different sizes and parameters to be tested, which will be utilised when applied to a real-world situation of an LSP. For example, in some test cases such as when a bigger range of RTE batches were simulated, the fleet utilisation rates where lower when compared to the control scenarios, therefore a smaller fleet or trucks of differing sizes may present a more favourable outcome for fleet utilisation and the impact on the RTE batch cost can be observed.

Thirdly, the model does not allow deliveries to be conducted between customers due to the design of the network during the controller class process. By adding links between customers, the problem becomes more complex and the model development more challenging. It was observed in the test case scenario of increased network complexity that adding more links has a negative impact on late batch costs as the probability of being selected for pick-up decreases as the truck has more options. Therefore, by adding links between customers, the rules would need to be devised carefully in order to assess the impact of enabling the trucks to go to other customers' facilities during the return process versus the rise in late batch costs. An extension of this work would be to observe a real network and represent the patterns of movement and real-world deliveries in order to develop more realistic simulations. The model has the flexibility to design the network as required and modify ad hoc as tested in Chapter 6 by increasing complexity.

During the scenario, certain assumptions such as the unloading and loading times were integrated into the simulations as the simplest solution. The scenario was used to demonstrate the interactions between the agents and establish whether the algorithms developed enable integration of reverse flow with the forward flow during scheduling, increasing RTE utilisation rates. The model proved to be able to ensure that the RTE is scheduled on-time to the distribution centres, allowing the RTE to be re-used.

This model considers the fleet and RTE batches, however the facilities are treated as boxes with inputs and outputs able to accommodate all flows. The purpose was to develop a RTE return policy and create an interaction between the fleet and RTE. By adding facility based constraints such as limits to the loadings or time windows, the outcomes for the routes, and therefore the costs would be different as the challenge of developing a schedule becomes greater. Instead of just adding constraints, a facility agent, which has rules and is integrated into the negotiations, may result in a more flexible schedule as features such as the time-window and storage constraints can be negotiated and a solution satisfying all parties reached. The inclusion of a facility agent would add more constraints to the trucks and RTE during negotiation but in some cases may enable a higher incentive. For example, if a batch has to be picked up immediately as it has not secured another day at a facility, then an urgent delivery would be requested and the weighting for collection would be favoured for this batch.

Also, as the network becomes more complex, with 4 distribution centres, 8 warehouse and 20 customers (compared to the control scenario of 1 dc, 2 warehouse, 5 customers), the simulation time for each run increases considerably with a simulation of high complexity network taking approximately 5 hours. This limits the amount of further simulations, which are conducted with complex networks and when representing real-world networks. Precautions will need to be taken to enable enough time for the runs to be completed before the final schedule date. Therefore, the java and eclipse implementation should be compared to other options such as C* for future development, particularly if further rules and interactions are added between agents such as the inclusion of a storage facility agent.

The bidding rules are devised so that if a batch cannot secure a delivery, it will split and another agent is formulated, in order to try and secure a pick-up. However, after the results of the simulation, it was noted that a pick-up may be encouraged by increasing the RTE batch loads. Therefore, modifications to the algorithm to include steps where batches can amalgamate to become bigger either before or after being split would make for interesting investigation.

The RTE batch is an amalgamation of a number of RTE. This was implemented in order to reduce the complexity of the agent system. In fact, with the outcome of the results, batching RTE is more favourable as smaller batch sizes and a range of batches, lead to higher fleet and late RTE batch costs, in addition to lower fleet utilisation rates. Therefore, it would be beneficial to tag a batch of RTE rather than each individual RTE for the purposes of the model. Or if each batch is tagged, then a further set of rules can be implemented before the auction process begins, where the batches use their own knowledge to join with other RTE in order to get selected for deliveries. By including further negotiations prior to the scheduling system developed, the process becomes very complex and the simulation very difficult and challenging, especially as the more complex scenarios conducted (section 6.9). Therefore, RTE batching is an important feature to implement when designing an efficient RTE return system.

Despite the inability to clearly compare the model to other research methods due to the lack of similarity, there is a clear contribution to knowledge of the RFIDERTEAM. The model enables a system of visibility, which is modular and allows ad hoc component development. Also, the model provides flexible, transparent, robust architecture, and therefore in the long-run encourages flexibility for application rather than going for short-term efficiency from commercial solutions by using tool-kits.

7.4 Summary

This chapter discussed the results of the RFID enhanced RTE Agent model (RFIDERTEAM) of which various scenarios were tested in Chapter 6. The findings have been interpreted where the RFIDERTEAM has demonstrated that reverse and forward flow scheduling has been successfully integrated with an autonomous approach and the model has been developed for real-world testing. The amount of time that a batch is processed and awaiting for return is critical in determining the overall total cost of operating the network. Although it will be beneficial to return the RTE as soon as possible for re-use, this may not always produce the most beneficial results for the network in terms of cost, therefore the date for which the RTE is to be returned may not be the lowest possible.

The utilisation of the truck and RTE demonstrates that the decentralised decision-making system has enabled a level of autonomy between the RTE Batch agents and Truck agents when they are scheduled. These methods effectively allow flexibility during the auction process of scheduling routes due to the contingencies implemented. This has led to achieving the objective of returning the RTE to a point of re-use as soon as possible. This enables a strategic change in the network by either modifying the fleet or the LSP could tender for more business in order to utilise the current fleet more effectively, creating profitable routes. The workings of the agent based modelling and simulation, supports the hypothesis that ABM is a natural application of RTE networks and that RFID technology is essential to close the gap between the physical and virtual worlds. Based on the results, discussion and limitations, Chapter 8 presents the conclusions and addresses the research questions presented in Chapter 1.

Chapter 8

Conclusions

Can the implementation of RFID technology and autonomous agents, improve RTE management and enhance competitive advantage for logistics service providers? This can be broken down into a number of smaller questions.

- Investigate the feasibility of the application of agent based modelling to model the RTE network utilising RFID based data;
- Develop a methodology to enable the agents to integrate forward and reverse flows of RTE simultaneously, generating a feasible schedule;
- Develop an algorithm which can be used to promote RTE utilisation i.e. return RTE back as soon as possible for re-use;

- Construct a platform for LSPs to use to assess various strategies for their client base and utilise the increased visibility, which RFID enabled RTE provides in comparison to a non-RFID enabled network to enhance their competitiveness.

The research questions presented in Chapter 1 are addressed, followed by a summary of the major findings. The contributions of this study are discussed and an investigation into future work is documented. The chapter is concluded with a summary.

8.1. Addressing Research Questions

Returning to the questions posed at the beginning of this study, this section addresses each question:

i) How to improve the RFID Enabled RTE network;

The combination of agent based modelling and RFID technology application to the RTE network has proven to be advantageous in improving network efficiency and enhancing RTE management. As studies in the literature review asserted, RFID improves data and physical process automation, which when applied to RTE, results in significant improvements on areas such as RTE loss prevention and tracking improvement. Agents utilise the increased data visibility further, by conducting autonomous decision-making.

ii) How the data gathered from RFID tags on RTE can be utilised in real-time;

Each RTE is tagged and at every checkpoint (i.e., distribution facility, truck loading/unloading), the status will be updated. The work conducted in this study utilises information collected such as location and destination, feeding it into the agent environment, enabling the agents to negotiate the best possible schedule with the most up-to-date data. The data frames stored on the RFID tag for each object (RTE and trucks) are crucial in providing the knowledge required in the agent based decision support system, which is able to utilise the data gathered from the RFID tags at its most advantageous, in real-time.

iii) How to model the RTE network and capture the distributed nature;

The RFID enhanced RTE agent model (RFIDERTEAM) captures the functionality of the RTE and trucks by implementing two types of agents: the Truck agent and the RTE batch agent. The agents are programmed and the model and simulation functions are developed using Java programming on the integrated development environment, Eclipse. The modular nature of formulating agents enables each part of the network to be represented and extensions to be added, for example if in the future the system is developed to include maintenance at specified facilities, then a maintenance agent can be added to the current system.

iv) How to enable decentralised decision-making from the perspective of the RTE;

Each agent is programmed with rules, which enable it to make decisions for each object it represents. Once the simulations begin, each agent is formulated in the model for each truck in the network and each RTE batch. The RTE in the batch are going to the same destination on the same date. The agents adhere to their rules, using the knowledge gathered from their information pool to conduct decisions. Once a batch is scheduled or a truck has completed its route and the auction is closed, the agent dissolves. Therefore, each object in the network has a separate decision unit, which is formulated and dissolved during each simulation run.

v) *What rules and procedures are required in order to ensure that the RTE is utilised to its maximum;*

Algorithms such as the specific one developed to integrate RTE scheduling of the reverse flow, the RTE Agent Scheduling Algorithm (RTEASA), and other algorithms such as Dijsktra's shortest path algorithm are used to enable interactions and negotiation features between agents. There are also rules such as trucks with maximum time, volume and weight constraints. The contingencies in the RTEASA and the constant interactions between the agents during the scheduling process enables the RTE Batch agent to negotiate a return journey where the agent continues through the auction process until each RTE is scheduled for return deliveries as well as forward delivery. This ensures that the RTE is scheduled for return, improving RTE utilisation. The agent based system provides a RTE scheduling environment for LSPs where this type of tool is currently unavailable, and is able to improve RFID enabled RTE management.

vi) How to simultaneously improve the client's supply chain, increasing competitiveness of the LSP through RFID enabled RTE.

The RFID enhanced RTE agent model (RFIEDERTEAM) facilitates each RTE to be monitored, preventing losses, whilst the agent based system implements a return flow, which ensures that the RTE are utilised as much as possible, whilst considering fleet constraints. The LSP therefore utilises the resources it owns efficiently and improves its client's network, whilst also improving their own competitive advantage by offering further value added services to increase their business portfolio. Further explanation of the multiple levels of advantage which are gained by LSPs through RFID enabled RTE management is discussed by Aggarwal and Ming (2013b) (Appendix A-1).

8.2. Summary and Significance of Findings

One of the more significant findings to emerge from this study is that the batch processing day, which is the amount of time the RTE is being processed in the forward flow and awaiting pick-up, needs to be varied in order to achieve lower overall total costs, which depends on network factors such as road complexity, batch numbers or network size. The shortest batch processing time may increase RTE utilisation, but total network costs will increase. The RFIDERTEAM provides an autonomous agent based platform to simulate the most cost effective solution to integrating the reverse flow of RTE.

The fleet is scheduled by the model so that the RTE is returned within a specific time. Batches which are not returned in a specified time for re-use or at all, have a significant impact on the total cost. The importance of the fleet is clearly supported by the current findings, as the research has shown that the majority of total costs are the truck operating costs when the RFIDERTEAM is implemented. However, fluctuations of late batch costs can lead to a rise in the total cost, therefore the batch processing day is critical in balancing the late batch cost and truck operating cost.

The study has gone some way towards enhancing our understanding of the relationship between batch costs, truck operating costs and RTE and fleet utilisation. Taken together, these findings suggest a role for the batch processing day in reducing the overall cost of operating the RTE network. The return of RTE batches should form an integral part of the scheduling process. The present study however makes a noteworthy contribution as it provides a RTE scheduling tool which reduces the impact of poor RTE management (as discussed in the literature in chapter 2) by integrating the reverse flow scheduling with the forward flow scheduling.

In addition to a contribution to knowledge with the RFID enabled RTE linking to a specifically developed agent model, findings from this study make several contributions to the current literature as presented in the publications in Appendix A. Further contribution of this research is in section 5.3.

8.3. Contribution of the Research

The RFID enabled RTE network domain is an under-researched area, therefore the methods of the model have been developed. The aim of the model is to improve the RFID enabled RTE network, whilst considering the fleet, which is facilitated by integrating the reverse deliveries into the forward flow during scheduling. The different levels of objectives as described in Chapter 2 are integrated into the model, each of which requires a decentralised decision-making format in order for it to be implemented. A number of contributions emerge from this research.

LSP Competitive Advantage via RFID enabled RTE

The value-addition aspect of LSPs as a competitive advantage from the perspective of theory (Christopher 2005) was identified in this thesis based on literature review. Moreover, the better management of RTE (as elaborated in Research Question 1) by using RFID will further reduce the cost and increase value-addition of LSP (as elaborated in Research Question 6). The idea of using decentralised decision making of RTE (using MAS) as a potential unique selling point in gaining LSP's competitive advantage is an additional thesis contribution to the theory.

Aggarwal and Lim (2013a) assert that systems need to be developed in order to increase the utilisation rates of RTE, which in turn are used to increase the competitiveness of LSPs. The RFID enhanced RTE Agent Model (RFIDERTEAM) developed enables RTE to be moved from network loss makers to a network improving tool. The model increases competitive advantage for LSPs by developing a scheduling arena, which involves negotiation of both forward deliveries and reverse deliveries of the RTE in order to return them as soon as possible for re-use. This enables the LSP to utilise the RTE to its maximum potential, take advantage of the constant tracking offered by the application of RFID, which in itself prevents losses, and utilising the decision-making functionality from the agents to ensure the RTE is in use.

Agent-based Modelling Approach to the RTE Network Management

The research conducted fills a gap in the knowledge by providing an agent based platform for RTE management improvement. The toolkits currently available are too rigid or need too many modifications, which result in the development of a model which has too many assumptions and compromises. Although it is a very arduous and challenging task to build agents from first principles using Java code, the result has been the development of a platform with modular functionality addressing a specific problem area. Therefore, additions and changes are easy to implement depending on the network being modelled and the parameters which need to be tested within the RFID enabled RTE problem domain. The agents which have been developed provide an autonomous system for RTE management whereby the agents utilise real-time data to conduct effective decisions, providing an essential contribution to the RTE problem domain.

Information Centric Environment: Radio-Frequency Identification Application to RTE

The model utilises the data frames which are made available through RFID technology application throughout the RTE network, taking advantage of the data in order to utilise it for higher levels than just a tracking facility and as a point solution (for example, a reduction in RTE losses). Thus far, in the literature presented in Chapter 2, the utilisation of RFID data has been limited to data and physical process automation. RFID as a standalone technology has proven to be advantageous due to its properties, such as no line of sight required during scanning and readable through materials. The functionality is further increased when it is combined with technologies such as GPS and sensors, enabling a more superior form of tracking of items. Agent technology further utilises the superiority of RFID technology by utilising the real-time information to enable effective decision-making, taking the technology beyond just a tracking tool. The total network visibility that RFID tags offer to the items that they are tagged to, creates a real-time information-centric environment for the agent, which is a valuable tool for effective and efficient decision-making when combined with agents. The data gathered in real-time provides each agent with essential information, which is used to enable the agent to create agility and flexibility in the network.

Graphical User Interface (GUI)

A graphical user interface was developed so that the model and simulation is accessible to LSPs. This is an essential interface for non-programmers as the code is overwhelming. This allows the model to be explained easily to logistics practitioners when gathering the required data to develop scenarios and conduct simulations.

Decentralised Decision-Making in RFID enabled RTE Networks

Aggarwal and Ming (2012) assert that a move to a decentralised decision-making format in supply chains would be more aligned with the distributed nature of supply chains and increases competitiveness. The use of a system where the control is centralised has limited and ineffective application in the RTE network domain, and therefore the model developed has implemented agents to represent RTE and trucks filling a gap in the knowledge. The RTE based model developed allows each object in the network to be represented by an agent, which is programmed with rules and interaction based algorithms in order to enable the agent to make decisions in their best interest. The model and simulation has provided a platform for further research to be conducted by academics and industry practitioners.

Algorithm Development: Return Flow Integration

The scheduling algorithm (RTEASA) developed specifically addresses a problem, which has not been tackled in depth in the literature. The algorithm implemented integrates the return flows of RTE deliveries simultaneously with the forward flow. This goes well beyond the capability of current models for the RFID enabled RTE problem addressed. The outputs of the model such as costs and utilisation rates have been presented in Chapter 6, which highlights the contribution that the platform developed makes not only in research but also when applied to industry.

Industry Application

The agent based model provides a platform for industry to test how to gain competitive advantage on operational, tactical and strategic levels by simulating changes in parameters and conditions. The costs and benefits of implementing different strategies such as different sizes in fleet or varying numbers of RTE is tested to develop results, which offer improved profitability. In addition, the model considers the multiple levels of clients with different requirements allowing different types of RTE and trucks to flow through the same network as required by a LSP in order to test in a real-world application.

Development in the Area of Multi-Agent System Applications

The model is a contribution not only to the area of RTE management but also delivers a MAS in RFID enabled RTE management. One of the key requirements for further development in the emerging field of MAS is the development of applications. The research presented in this thesis addresses this by building a foundation for MAS development in RTE management.

8.4. Future Work

A number of possible future studies using the same experimental set-up are apparent. Firstly, further investigation and experimentation into the significance of the batch processing day is strongly recommended, by comparing the current results to real-world data and scenarios and then conducting simulations to assess the impact. Secondly, it would be interesting to assess the effects of adding facility agents. The agent system will become more complex with added algorithms and rules and further agent interaction, therefore the impact on features such as the running time would need to be assessed. Thirdly, the issue of implementing RTE cycles with built-in maintenance pit-stops is an intriguing one, which could be usefully explored in further research. This could examine whether the current agents would have a maintenance module with extended data frames, or whether another group of agents would be more effective. Fourthly, it would be interesting to investigate the impact of including pick-ups from other customers during the reverse flow and carrying loads for example, integrating a reverse logistics network. Finally, further research might explore the development of the agent based platforms as a toolkit for RTE focussed management.

8.5. Summary

This chapter has concluded the thesis by summarising the findings and highlighting the contribution of the research conducted. Addressing the four reasons for why the project was undertaken firstly, agent based modelling provides an autonomous tool, which is effective in modelling RFID enabled RTE in a decentralised format utilising the real-time data facility. Secondly, the RFIDERTEAM enables autonomous agent interaction, which leads to a feasible schedule integrating both forward and reverse flows for each RTE batch. Thirdly, the RTEASA developed promotes the utilisation of RTE by including an automatic return flow for each batch of RTE, whilst considering the fleet costs and utilisation. Finally, the research conducted contributes an agent based platform which LSPs use in order to assess the most appropriate strategies to implement for each of their clients when assessing the impact of RTE management.

Taken together, the findings support strong recommendations to LSPs to observe their current RTE networks and investigate the application of RFID technology. If this technology is already in place, agent based modelling improves their own competitive advantage and their client's even further. Profitable clients will lead to profitable LSPs and in the current challenging economic climate, the role of the LSP to deliver innovative ways to achieve competitive advantage has never been so important.

References

ACAR, Y., KADIPASAOGLU, S. N. & DAY, J. M. 2009. Incorporating uncertainty in optimal decision making: Integrating mixed integer programming and simulation to solve combinatorial problems. *Computers & Industrial Engineering,* 56**,** 106-112.

AGGARWAL, R. & FOSSO WAMBA, S. In Preparation. Transforming Logistics Service Provider's Activities Using IT: The Case of RFID Technology. *Production Planning & Control: The Management of Operations*.

AGGARWAL, R. & LIM, M. 2009. Impact of RFID in Reverse Logistics. *11th Int. Conf. on The Modern Information Technology in the Innovation Processes of the Industrial Enterprise.* Bergamo, Italy.

AGGARWAL, R. & LIM, M. 2010. Utilisation of RFID to Increase Efficiency in the Returnable Equipment Supply Chain. *12th Int. Conf. on The Modern Information Technology in the Innovation Processes of the Industrial Enterprise.* Aalborg, Denmark.

AGGARWAL, R. & LIM, M. 2011. Increasing competitiveness of third party logistics with RFID. *IEEE Int. Conf. on RFID - Technology and Applications.* Barcelona, Spain.

AGGARWAL, R. & LIM, M. 2012a. Advances in Decision Support Systems on the Endeavour of Supply Chain Optimality. *International Conference on Industrial Logistics.* Croatia.

AGGARWAL, R. & LIM, M. 2012b. Improving Returnable Transport Equipment Management with an Autonomous Approach. *3rd International Conference on Dynamics in Logistics.* Bremen, Germany.

AGGARWAL, R. & LIM, M. 2013a. Aligining Physical and Virtual Logistical Spheres with Radio-Frequency Identification and Agent-Based Modelling. *International Jounral of Agile Systems and Management,* 6**,** 66-82.

AGGARWAL, R. & LIM, M. 2013b. Enhancing competitive advantage with radio-frequency identification (RFID) enabled returnable transport equipment (RTE). *Int. J. Information Technology and Management,* 12**,** 3-26.

AGGARWAL, R. & LIM, M. In Preparation. RFID Innovation: An Exploration of the Impact of RFID Technology in Outbound Logistics. *International Journal for RF Technologies: Research and Applications* Accepted with corrections.

AGGARWAL, R., LIM, M. & TAN, K. 2011. A Three-level Automation Approach for Radio-frequency Identification Enabled Returnable Transport Equipment to Enhance Supply Chain Competitiveness. *21st International Conference on Production Research.* Stuttgart, Germany.

ANGELES, R. 2005. RFID Technologies: Supply-chain Applications and Implementation Issues. *Information Systems Management,* 22**,** 51.

ARLBJØRN, J. S. & FREYTAG, P. V. 2013. Evidence of lean: a review of international peer-reviewed journal articles. *European Business Review,* 25**,** 174-205.

ARMSTRONG & ASSOCIATES, I. 2010a. *Global 3PL Market Size Estimates* [Online]. Available: http://www.3plogistics.com/3PLmarketGlobal.htm [Accessed 27/03/2012 2012].

ARMSTRONG & ASSOCIATES, I. 2010b. *U.S. and Global Third-Party Logistics Market Analysis is Released* [Online]. Armstrong & Associates, Inc. Available: http://www.3plogistics.com/PR_3PL_Financial-2010.htm [Accessed 7/04/2011 2011].

ATTARAN, M. 2007. RFID: an enabler of supply chain operations. *Supply Chain Management: An International Journal,* 12**,** 249-257.
AUSTIN, J. L. 1962. *How to Do Things with Words.,* Oxford, Oxford University Press.
BACHELDOR, B. 2006. *CHEP RFID-enables Reusable Containers for Auto Industry* [Online]. RFID Journal. Available: http://www.rfidjournal.com/article/view/2859 [Accessed 06/04/2011 2011].
BACHELDOR, B. 2007. *Aggressive Sales Growth Predicted for Passive Tags in North America* [Online]. RFID Journal. Available: http://www.rfidjournal.com/article/view/3573 [Accessed 27/03/12 2012].
BAÏNA, S., PANETTO, H. & MOREL, G. 2009. New paradigms for a product oriented modelling: Case study for traceability. *Computers in Industry,* 60**,** 172-183.
BANKS, J., HANNY, D., PACHANO, M., A. & THOMPSON, L., G. 2007. *RFID Applied,* New Jersey, John Wiley & Sons.
BARDAKI, C., KOUROUTHANASSIS, P. & PRAMATARI, K. 2011. Modeling the information completeness of object tracking systems. *The Journal of Strategic Information Systems,* In Press, Corrected Proof.
BARDI, E., J. & TRACEY, M. 1993. Transportation Outsourcing: A Survey of US Practices. *International Journal of Physical Distribution & Logistics Management,* 21**,** 15-21.
BARNEY, J. 1991. Firm resources and sustained competitive advantage. *Journal of management,* 17**,** 99-120.
BASK, A., H. 2001. Relationships among TPL providers and members of supply chains â€“ a strategic perspective. *Journal of Business & Industrial Marketing,* 16**,** 470-486.
BATES, J. 1994. The role of emotion in believable agents. *Commun. ACM,* 37**,** 122-125.
BAUER, B., MÜLLER, J., P. & ODELL, J. 2001. Agent UML: A Formalism for Specifying Multiagent Software Systems. *International Journal on Software Engineering and Knowledge Engineering,* 11**,** 1-24.
BAUER, B. & ODELL, J. 2005. UML 2.0 and Agents: How to Build Agent-based Systems with the new UML. *Journal of Engineering Applications of Artificial Intelligence,* 18**,** 141-157.
BEDEMAN, M. & GATTORNA, J. L. 2003. Third-and Fourth-party logistics service providers. *Gower Handbook of Supply Chain Management,* 5**,** 469-485.
BERGLUND, M., LAARHOVEN, P. V., SHARMAN, G. & WANDEL, S. 1999. Third-party logistics: Is there a future? *International Journal of Logistics Management,* 10.
BHATNAGAR, R., SOHAL, A., S. & MILLEN, R. 1999. Third party logistics services: a Singapore perspective. *International Journal of Physical Distribution & Logistics Management,* 29**,** 569-587.
BORDINI, R., BRAUBACH, L., DASTANI, M., SEGHROUCHNI, A. E. F., GOMEZ-SANZ, J., LEITE, J., O'HARE, G., POKAHR, A. & RICCI, A. 2006. A Survey of Programming Languages and Platforms for Multi-Agent Systems. *Informatica,* 30**,** 33-44.
BORSHCHEV, A. & FILIPPOV, A. 2004. From System Dynamics and Discrete Event to Practical Agent Based Modeling: Reasons, Techniques, Tools. *22nd International Conference of the System Dynamics Society.* Oxford, England.
BOSE, F., PIOTROWSKI, J. & SCHOLZ-REITER, B. 2009. Autonomously controlled storage management in vehicle logistics - applications of RFID and mobile computing systems. *International Journal of RF Technologies: Research and Applications,* 1**,** 57-76.
BOSE, I. & PAL, R. 2005. Auto-ID: Managing Anything, Anywhere, Anytime in the Supply Chain. *Communications of the ACM,* 48**,** 100-106.
BOTTANI, E. & BERTOLINI, M. 2009. Technical and Economic Aspect of RFID Implementation for Asset Tracking. *International Jounral of RF Technologies: Research and Appications,* 1**,** 169-193.
BOWMAN, C. & TOMS, S. 2010. Accounting for competitive advantage: the resource-based view of the firm and the labour theory of value. *Critical Perspectives on Accounting,* 21**,** 183-194.
BRYSON, J. 2000. Cross-paradigm analysis of autonomous agent architecture. *Journal of Experimental & Theoretical Artificial Intelligence,* 12**,** 165-189.

CAO, J., WANG, J., LAW, K., ZHANG, S. & LI, M. 2006. An interactive service customization model. *Information and Software Technology,* 48**,** 280-296.

CARRASCO-GALLEGO, R., PONCE-CUETO, E. & DEKKER, R. 2012. Closed-loop supply chains of reusable articles: a typology grounded on case studies. *International Journal of Production Research,* 50**,** 5582-5596.

CHAIB-DRAA, B., MOULIN, B., MANDIAU, R. & MILLOT, P. 1992. Trends in distributed artificial intelligence. *Artificial Intelligence Review,* 6**,** 35-66.

CHAO, C. C., YANG, J. M. & JEN, W. Y. 2007. Determining technology trends and forecasts of RFID by a historical review and bibliometric analysis from 1991 to 2005. *Technovation,* 27**,** 268-279.

CHEN, R.-S. & TU, M. 2009. Development of an agent-based system for manufacturing control and coordination with ontology and RFID technology. *Expert Systems with Applications,* 36**,** 7581-7593.

CHOW, H. K. H., CHOY, K. L. & LEE, W. B. 2007. A dynamic logistics process knowledge-based system - An RFID multi-agent approach. *Knowledge-Based Systems,* 20**,** 357-372.

CHRISTOPHER, M. 2000. The Agile Supply Chain : Competing in Volatile Markets. *Industrial Marketing Management,* 29**,** 37-44.

CHRISTOPHER, M. 2005. *Logistics and Supply Chain Management: Creating Value-Adding Networks*, Pearson Education Limited.

CHUANG, M.-L. & SHAW, W. 2007. RFID: Integration Stages in Supply Chain Management. *Engineering Management Review, IEEE,* 35**,** 80-87.

COHEN, P. R. & LEVESQUE, H. J. 1990. Intention is choice with commitment. *Artificial Intelligence,* 42**,** 213-261.

COLLINS, J. 2005a. *Carmakers Study RFID's ROI for Racks* [Online]. RFID Journal. Available: http://www.rfidjournal.com/article/view/1569 [Accessed 11/11/2011.

COLLINS, J. 2005b. *EU Logistics Providers Shirk RFID Role* [Online]. RFID Journal. Available: http://www.rfidjournal.com/article/view/1350 [Accessed 27/03/2012 2012].

COLLINS, J. 2006. *Tesco Revises RFID plans* [Online]. RFID Journal. Available: http://www.rfidjournal.com/article/articleview/2243/1/1/ 19/02/2010].

COYLE, J. J., BARDI, E., J. & LANGLEY, C. J. 2003. *Management of Business Logistics: A Supply Chain Perspective,* Ohio, South-Western.

CURTIN, J., KAUFFMAN, R., J. & RIGGINS, F., J. 2007. Making the 'MOST' out of RFID technology: a research agenda for the study of the adoption, usage and impact of RFID. *Information Technology and Management,* 8**,** 87.

D'INVERNO, M., LUCK, M., MICHAEL, G., KINNY, D. & WOOLDRIDGE, M. 2004. The dMARS Architecture: A Specification of the Distributed Multi-Agent Reasoning System. *Autonomous Agents and Multi-Agent Systems,* 9**,** 5-53.

DAMME, D. A. V. & PLOOS VAN AMSTEL, M. J. 1996. Outsourcing Logistics Management Activities. *The International Journal of Logistics Management,* 7**,** 85-94.

DANIEL, H. 2009. The cost and process of implementing RFID technology to manage and control returnable transport items. *International Journal of Logistics Research and Applications,* 12**,** 20.

DAPIRAN, P., LIEB, R., MILLEN, R. & SOHAL, A. 1996. Third party logistics services usage by large Australian firms. *International Journal of Physical Distribution & Logistics Management,* 26**,** 36.

DAVIDSSON, P., HENESEY, L., RAMSTEDT, L., TÖRNQUIST, J. & WERNSTEDT, F. 2005. An analysis of agent-based approaches to transport logistics. *Transportation Research Part C: Emerging Technologies,* 13**,** 255-271.

DELEN, D., HARDGRAVE, B. C. & SHARDA, R. 2007. RFID for Better Supply-Chain Management through Enhanced Information Visibility. *Production & Operations Management,* 16**,** 613-624.

DEMIR, A. 2010. The value of radio frequency identification (RFID) technology for managing pools of returnable transport items *In:* TUE: DEPARTMENT TECHNOLOGY MANAGEMENT, S. M. T. O. M. A. L. O. (ed.). Eindhoven : Technische Universiteit Eindhoven.

DHL. 2013. *The Comeback of the Robots* [Online]. Available: http://www.dhl-discoverlogistics.com/cms/en/course/technologies/reinforcement/reinforcement.jsp [Accessed 19/03/2011.

DHL, D. P. 2011. *Intelligent Transport Hits the Road* [Online]. Deutsche Post AG. Available: http://www.dp-dhl.com/en/logistics_around_us/from_our_divisions/dhl_smarttrucks.html 2013].

DIAS, J. C. Q., CALADO, J. M. F., OSORIO, A. L. & MORGADO, L. F. 2009. RFID together with multi-agent systems to control global value chains. *Annual Reviews in Control,* 33**,** 185-195.

DOMDOUZIS, K., KUMAR, B. & ANUMBA, C. 2007. Radio-Frequency Identification (RFID) applications: A brief introduction. *Advanced Engineering Informatics,* 21**,** 350-355.

DUHAIME, R., RIOPEL, D. & LANGEVIN, A. 2001. Value Analysis and Optimization of Reusable Containers at Canada Post. *Interfaces,* 31.

EPCGLOBAL, G. 2012. *Electronic Product Code (EPC): An Overview* [Online]. GS1. Available: http://www.gs1.org/docs/epcglobal/an_overview_of_EPC.pdf [Accessed 17/01/2012 2012].

ERGEN, E., AKINCI, B. & SACKS, R. 2007. Life-cycle data management of engineered-to-order components using radio frequency identification. *Advanced Engineering Informatics,* 21**,** 356-366.

ETZIONI, O. 1996. Moving up the information food chain: deploying softbots on the world wide web. *Proceedings of the thirteenth national conference on Artificial intelligence - Volume 2.* Portland, Oregon: AAAI Press.

FERBER, J. 1999. *Multi-Agent Systems: An Introduction to Distributed Artificial Intelligence,* Harlow, Addison-Wesley.

FINKENZELLER, K. 2010. *RFID Handbook: Fundamentals and Applications in Contactless Smart Cards, Radio Frequency Identification and Near-Field Communication,* Chichester, Wiley.

FLEISCHMANN, M., BEULLENS, P., BLOEMHOF-RUWAARD, J., M. & VAN WASSENHOVE, L., N. 2001. The impact of product recovery on logistics network design. *Production and Operations Management,* 10**,** 156.

FORGET, P., D'AMOURS, S., FRAYRET, J.-M. & GAUDREAULT, J. 2009. Study of the performance of multi-behaviour agents for supply chain planning. *Computers in Industry,* 60**,** 698-708.

FORUM, A. 1994.

FOX, M. S., BARBUCEANU, M. & TEIGEN, R. 2000. Agent-oriented supply-chain management. *International Journal of Flexible Manufacturing Systems,* 12**,** 165-188.

FRANCE, G. & PARTNERS, W. 2007. Returnable transport items: requirements to improve reusable asset management. *In:* IST-2005-033546, E. C. C. N. (ed.) *Building Radio frequency IDentification for the Global Environment.*

FRANKLIN, S. & GRAESSER, A. 1997. Is it an Agent, or Just a Program?: A Taxonomy for Autonomous Agents. *Proceedings of the Workshop on Intelligent Agents III, Agent Theories, Architectures, and Languages.* Springer-Verlag.

GARCÍA-FLORES, R., WANG, X. Z. & GOLTZ, G. E. 2000. Agent-based information flow for process industries' supply chain modelling. *Computers & Chemical Engineering,* 24**,** 1135-1141.

GARCÍA-ARCA, J. & PRADO-PRADO, J. C. 2006. Logistics improvement through packaging rationalization: a practical experience. *Packaging technology and science,* 19**,** 303-308.

HEATH, B., HILL, R. & CIARALLO, F. 2009. A Survey of Agent-Based Modeling Practices (January 1998 to July 2008). *Journal of Artificial Societies and Social Simulation,* 12**,** 9.

HELLSTROM, D. 2009. The cost and process of implementing RFID technology to manage and control returnable transport items. *International Journal of Logistics Research and Applications: A Leading Journal of Supply Chain Management,* 12**,** 1 - 21.

HELLSTROM, D. & JOHANSSON, O. 2010. The impact of control strategies on the management of returnable transport items. *Transportation Research Part E-Logistics and Transportation Review,* 46**,** 1128-1139.
HELLSTROM, D. & SAGHIR, M. 2007. Packaging and logistics interactions in retail supply chains. *Packaging Technology and Science,* 20**,** 197-216.
HERTZ, S. & ALFREDSSON, M. 2003. Strategic development of third party logistics providers. *Industrial Marketing Management,* 32**,** 139-149.
HINES, P., HOLWEG, M. & RICH, N. 2004. Learning to evolve: a review of contemporary lean thinking. *International Journal of Operations & Production Management,* 24**,** 994-1011.
HOLLOWAY, S. 2009. *RFID: An Introduction*.
HOLMSTRÖM, J., KAJOSAARI, R., FRÄMLING, K. & LANGIUS, E. 2009. Roadmap to tracking based business and intelligent products. *Computers in Industry,* 60**,** 229-233.
HOMS, C., METCALFE, D. & TAKAHASHI, S. 2004. Exposing the myth of the 5-cent RFID tag. *Forrester Research Inc., Cambridge, MA*.
HUBER, N., MICHAEL, K. & MCCATHIE, L. Barriers to RFID Adoption in the Supply Chain. RFID Eurasia, 2007 1st Annual, 5-6 Sept. 2007 2007. 1-6.
HUHNS, M. N. & STEPHENS, L. M. 1999. Multiagent systems and societies of agents. *In:* GERHARD, W. (ed.) *Multiagent systems.* MIT Press.
HÜLSMANN, M., GRAPP, J. & LI, Y. 2008. Strategic adaptivity in global supply chains—Competitive advantage by autonomous cooperation. *International Journal of Production Economics,* 114**,** 14-26.
HÜLSMANN, M., WINDT, K. & BÖSE, F. 2007. Catalogue of Criteria for Autonomous Control in Logistics. *Understanding Autonomous Cooperation and Control in Logistics.* Springer Berlin Heidelberg.
HUNT, S. D. & DAVIS, D. F. 2008. Grounding Supply Chain Management In Resource-Advantage Theory. *Journal of supply chain management,* 44**,** 10-21.
ILIC, A., NG, J. W. P., BOWMAN, P. & STAAKE, T. 2009. The value of RFID for RTI management. *Electronic Markets,* 19**,** 125-135.
ILIE-ZUDOR, E., KEMÉNY, Z., VAN BLOMMESTEIN, F., MONOSTORI, L. & VAN DER MEULEN, A. 2011. A survey of applications and requirements of unique identification systems and RFID techniques. *Computers in Industry,* 62**,** 227-252.
IRANI, Z., GUNASEKARAN, A. & DWIVEDI, Y. K. 2010. Radio frequency identification (RFID): research trends and framework. *International Journal of Production Research,* 48**,** 2485-2511.
IVANOV, D. 2010. An Adaptive Framwork for Aligning (Re)Planning Decisions on Supply Chain Strategy, Design, Tactics and Operations. *International Jounral of Production Research,* 48**,** 3999-4017.
JAYARAMAN, V., ROSS, A. D. & AGARWAL, A. 2008. Role of information technology and collaboration in reverse logistics supply chains. *International Journal of Logistics-Research and Applications,* 11**,** 409-425.
JEDERMANN, R., BEHRENS, C., WESTPHAL, D. & LANG, W. 2006. Applying autonomous sensor systems in logistics--Combining sensor networks, RFIDs and software agents. *Sensors and Actuators A: Physical,* 132**,** 370-375.
JENNINGS, N. R. 2000. On Agent-Based Software Engineering. *Artificial Intelligence,* 117**,** 277-296.
JOHANSSON, O. & HELLSTRÖM, D. 2007. The effect of asset visibility on managing returnable transport items. *International Journal of Physical Distribution & Logistics Management,* 37**,** 799.
JONATHAN, C. 2005. *Europe Finds Its Own Path to ROI* [Online]. RFID Journal. Available: http://www.rfidjournal.com/article/purchase/1933 [Accessed 06/04/2011 2011].
JOURNAL, R. 2005. *A Summary of RFID Standards* [Online]. RFID Journal. Available: http://www.rfidjournal.com/article/view/1335 [Accessed 17/01/2012.
JUNG, H., FRANK CHEN, F. & JEONG, B. 2008. Decentralized supply chain planning framework for third party logistics partnership. *Computers & Industrial Engineering,* 55**,** 348-364.

KARKKAINEN, M., ALA-RISKU, T. & FRAMLING, K. 2004. Efficient tracking for short-term multi-company networks. *International Journal of Physical Distribution & Logistics Management,* 34**,** 545.

KÄRKKÄINEN, M., HOLMSTRÖM, J., FRÄMLING, K. & ARTTO, K. 2003. Intelligent products--a step towards a more effective project delivery chain. *Computers in Industry,* 50**,** 141-151.

KHAN, F. 2004. *Chip Maker Tries 'Snack and Trace'* [Online]. RFID Jounral. Available: http://www.rfidjournal.com/articles/view?964 [Accessed 12/04/2013.

KIRITSIS, D. 2011. Closed-loop PLM for intelligent products in the era of the Internet of things. *Computer-Aided Design,* 43**,** 479-501.

KNEMEYER, M., A. & MURPHY, P., R. 2005. Is the glass half full or half empty?: An examination of user and provider perspectives towards third-party logistics relationships. *International Journal of Physical Distribution & Logistics Management,* 35**,** 708-727.

KO, H. J., KO, C. S. & KIM, T. 2006. A hybrid optimization/simulation approach for a distribution network design of 3PLS. *Computers & Industrial Engineering,* 50**,** 440-449.

KROON, L. & VRIJENS, G. 1995. Returnable containers: An example of reverse logistics. *International Journal of Physical Distribution & Logistics Management,* 25**,** 56.

KYE, D., LEE, J. & LEE, K.-D. 2013. The perceived impact of packaging logistics on the efficiency of freight transportation (EOT). *International Journal of Physical Distribution & Logistics Management,* 43**,** 707-720.

LABARTHE, O., ESPINASSE, B., FERRARINI, A. & MONTREUIL, B. 2007. Toward a methodological framework for agent-based modelling and simulation of supply chains in a mass customization context. *Simulation Modelling Practice and Theory,* 15**,** 113-136.

LANGER, N., FORMAN, C., KEKRE, S., SCHELLER-WOLF & ANDREW, A. 2007. Assessing the Impact of RFID on Return Center Logistics. *Interfaces,* 37.

LANGLEY, J. 2012. Third Party Logistics Study 2012.

LANGLEY, J. & CAPGEMINI 2007. The state of logistics outsourcing: 2007 third-party logistics. *12th Annual Study of Third Party Logistics.*

LANGLEY, J. C. 2010. The State of Logistics Outsourcing,Results and Findings of the 15th Annual Study: 2010 Third-party Logistics. Capgemini.

LEE, C. K. M. & CHAN, T. M. 2009. Development of RFID-based Reverse Logistics System. *Expert Systems with Applications,* 36**,** 9299-9307.

LEE, D. & PARK, J. 2008. RFID-based traceability in the supply chain. *Industrial Management & Data Systems,* 108**,** 713-725.

LEE, H. & OZER, O. 2007. Unlocking the value of RFID. *Production and Operations Management,* 16**,** 40-64.

LEE, H. & WHANG, S. 1999. Decentralized Multi-Echelon Supply Chains: Incentives and Information. *Management Science,* 45**,** 633-640.

LEWIS, I. & TALALAYEVSKY, A. 2000. Third-party logistics: leveraging information technology. *Journal of Business Logistics,* 20**,** 373-87.

LI, S., VISICH, J., K. , KHUMAWALA, B., M. & ZHANG, C. 2006. Radio frequency identification technology: applications, technical challenges and strategies. *Sensor Review,* 26**,** 193.

LIAO, W.-P., LIN, T. M. Y. & LIAO, S.-H. 2011. Contributions to Radio Frequency Identification (RFID) research: An assessment of SCI-, SSCI-indexed papers from 2004 to 2008. *Decision Support Systems,* 50**,** 548-556.

LIEB, R. & BENTZ, B. A. 2005. The North American third party logistics industry in 2004: the provider CEO perspective. *International Journal of Physical Distribution & Logistics Management,* 35**,** 595-611.

LIN, C.-Y. 2008. Factors affecting the adoption of radio frequency identification technology by logistics service providers: an empirical study. *International Journal of Management,* 25**,** 488-99.

LIN, L. C. 2009. An integrated framework for the development of radio frequency identification technology in the logistics and supply chain management. *Computers & Industrial Engineering,* 57**,** 832-842.

LOGICACMG 2004. Making Waves: Rfid adoption in returnable packaging.

LOU, P., ZHOU, Z.-D., CHEN, Y.-P. & AI, W. 2004. Study on multi-agent-based agile supply chain management. *The International Journal of Advanced Manufacturing Technology,* 23**,** 197-203.

LU, S. Y. P., LAU, H. Y. K. & YIU, C. K. F. 2012. A hybrid solution to collaborative decision-making in a decentralized supply-chain. *Journal of Engineering and Technology Management,* 29**,** 95-111.

LUGER, G. 2005. *Artificial Intelligence: Structures and Strategies for Complex Problem Solving,* London, Addison-Wesley.

MAES, P. 1994. Agents that reduce work and information overload. *Commun. ACM,* 37**,** 30-40.

MARASCO, A. 2008. Third-party logistics: A literature review. *International Journal of Production Economics,* 113**,** 127-147.

MARTIN, R., C. 2003. *UML for Java Programmers,* USA, Prentice Hall Professional Technical Reference.

MARTÍNEZ-SALA, A. S., EGEA-LÓPEZ, E., GARCÍA-SÁNCHEZ, F. & GARCÍA-HARO, J. 2009. Tracking of Returnable Packaging and Transport Units with active RFID in the grocery supply chain. *Computers in Industry,* 60**,** 161-171.

MASON-JONES, R., NAYLOR, B. & TOWILL, D. R. 2000. Lean, agile or leagile? Matching your supply chain to the marketplace. *International Journal of Production Research,* 38**,** 4061-4070.

MCCARTHY, J. 1979. Ascribing mental qualities to machines. *In:* RINGLE, M. (ed.) *Philosophical Perspectives in Artificial Intelligence.* Humanities Press.

MCFARLANE, D., SARMA, S., CHIRN, J. L., WONG, C. Y. & ASHTON, K. 2003. Auto ID systems and intelligent manufacturing control. *Engineering Applications of Artificial Intelligence,* 16**,** 365-376.

MCFARLANE, D. & SHEFFI, Y. 2003. The impact of automatic identification on supply chain operations. *International Journal of Logistics Management,* 14**,** 1.

MCKERROW, D. 1996. What makes reusable packaging systems work. *Logistics Information Management,* 9**,** 3.

MEHRJERDI, Y. Z. 2008. RFID-enabled systems: a brief review. *Assembly Automation,* 28**,** 235-245.

MELSKI, A., MULLER, J., ZEIER, A. & SCHUMANN, M. Improving supply chain visibility through RFID data. Data Engineering Workshop, 2008. ICDEW 2008. IEEE 24th International Conference on, 7-12 April 2008 2008. 102-103.

MENG, S., CHIU, D., KAFEZA, E., WENYIN, L. & LI, Q. 2010. Automated management of assets based on RFID triggered alarm messages. *Information Systems Frontiers,* 12**,** 563-578.

MES, M., VAN DER HEIJDEN, M. & VAN HARTEN, A. 2007. Comparison of agent-based scheduling to look-ahead heuristics for real-time transportation problems. *European Journal of Operational Research,* 181**,** 59-75.

MEYER, G. G., FRÄMLING, K. & HOLMSTRÖM, J. 2009. Intelligent Products: A survey. *Computers in Industry,* 60**,** 137-148.

MICHEL, F., FERBER, J. & DROGOUL, A. 2009. Multi-Agent Systems and Simulation: A Survey from the Agent Community's Perspective. *In:* UHRMACHER, A., M. & DANNY, W. (eds.) *Multi-Agent Systems.* Boca Raton: CRC Press.

MIN, H. 2013. Examining logistics outsourcing practices in the United States: from the perspectives of third-party logistics service users. *Logistics Research,* 6**,** 133-144.

MIN, H. & JOO, S. J. 2006. Benchmarking the operational efficiency of third party logistics providers using data envelopment analysis. *Supply Chain Management-an International Journal,* 11**,** 259-265.

MO, J., P. T., GAJZER, S., FANE, M., WIND, G., SNIOCH, T., LARNACH, K., SEITAM, D., SAITO, H., BROWN, S., WILSON, F. & LERIAS, G. 2009. Process integration for paperless delivery using

EPC compliance technology. *Journal of Manufacturing Technology Management,* 20**,** 866-886.
MONTEIRO, T., ROY, D. & ANCIAUX, D. 2007. Multi-site coordination using a multi-agent system. *Computers in Industry,* 58**,** 367-377.
MOORE, R. C. 1984. A Formal Theory of Knowledge and Action.
MOYAUX, T., CHAIB-DRAA, B. & D'AMOURS, S. 2007. Information Sharing as a Coordination Mechanism for Reducing the Bullwhip Effect in a Supply Chain. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on,* 37**,** 396-409.
MULLER, J. P. 1999. The Right Agent (Architecture) to do the Right Thing. *Proceedings of the 5th International Workshop on Intelligent Agents V, Agent Theories, Architectures, and Languages.* Springer-Verlag.
NEELY, B. 2009. *DHL Says Its SmartTrucks Save Money, Time and CO2* [Online]. RFID Journal. Available: http://www.rfidjournal.com/articles/view?5261 [Accessed 23/03/2013.
NGAI, E. W. T., MOON, K. K. L., RIGGINS, F. J. & YI, C. Y. 2008. RFID research: An academic literature review (1995-2005) and future research directions. *International Journal of Production Economics,* 112**,** 510-520.
NIELSEN, I., LIM, M., NIELSEN, P. & AGGARWAL, R. 2012. A Decision Support System for Waste Collection Management and its Potential Improvement with Radio-frequency Identification Technology (RFID)", . *International Journal of Environmental Technology and Management,* In Print.
NIKOLAI, C. & MADEY, G. 2009. Tools of the Trade: A Survey of Various Agent Based Modeling Platforms. *Journal of Artificial Societies and Social Simulation,* 12**,** 2.
NILSSON, N. J. 2012. John McCarthy 1927-2011 A Biographical Memoir by Nils J. Nilsson.
NWANA, H. S. & NDUMU, D. T. 1999. A perspective on software agents research. *Knowl. Eng. Rev.,* 14**,** 125-142.
O'CONNOR, M. C. 2009. *Volkswagen de Mexico Employing RFID to Improve Parts Distribution* [Online]. RFID Journal. Available: http://www.rfidjournal.com/article/view/4908 [Accessed 26/08/2011.
OLAVARRIETA, S. & ELLINGER, A. E. 1997. Resource-based theory and strategic logistics research. *International Journal of Physical Distribution & Logistics Management,* 27**,** 559-587.
PANAIT, L. & LUKE, S. 2005. Cooperative Multi-Agent Learning: The State of the Art. *Autonomous Agents and Multi-Agent Systems,* 11**,** 387-434.
PARUNAK, H., DYKE , SVEN, B., MITCH, F. & JAMES, O. 2004. A Design Taxonomy of Multi-agent Interactions. *In:* GIORGINI, P., MÜLLER, J. & ODELL, J. (eds.) *Agent-Oriented Software Engineering IV.* Springer Berlin Heidelberg.
PENROSE, E. T. 1995. *The Theory of the Growth of the Firm*, Oxford University Press.
PEPPA, V. P. & MOSCHURIS, S. J. 2013. RFID technology in supply chain management: a review of the literature and prospective adoption to the Greek marke" t. *Global Journal of Engineering Education,* 15.
PORTER, M. E. 1985. *Competitive Advantage Creating and Sustaining Superior Performance,* New York, The Free Press.
POWER, D., SHARAFALI, M. & BHAKOO, V. 2007. Adding value through outsourcing: Contribution of 3PL services to customer performance. *Management Research News,* 30**,** 228.
PRENDERGAST, G., P. 1995. The EC directive on packaging and packaging waste: current status and logistical implications. *Logistics Information Management,* 8**,** 10-17.
QRUNFLEH, S. & TARAFDAR, M. 2013. Lean and agile supply chain strategies and supply chain responsiveness: the role of strategic supplier partnership and postponement. *Supply Chain Management: An International Journal,* 18**,** 571-582.
RAILSBACK, S., LYTINEN, S. & JACKSON, S. 2006. Agent-based Simulation Platforms: Review and Development Recommendations. *SIMULATION,* 82**,** 609-623.

RAJESH, R., PUGAZHENDHI, S., GANESH, K., MURALIDHARAN, C. & SATHIAMOORTHY, R. 2010. Influence of 3PL service offerings on client performance in India. *Transportation Research Part E: Logistics and Transportation Review,* 47**,** 149-165.

RAO, A. S. & GEORGEFF, M. P. 1995. BDI Agents: From Theory to Practice. *International Conference on Multi-agent Systems.*

RAZZAQUE, M. A. & SHENG, C. C. 1998. Outsourcing of logistics functions: a literature survey. *International Journal of Physical Distribution & Logistics Management,* 28**,** 89-107.

ROFF, J., T. 2003. *UML: A Beginner's Guide,* California, McGraw-Hill/Osbourne.

RÖNKKÖ, M., KÄRKKÄINEN, M. & HOLMSTRÖM, J. 2007. Benefits of an item-centric enterprise-data model in logistics services: A case study. *Computers in Industry,* 58**,** 814-822.

ROSENAU, W. V., TWEDE, D., MAZZEO, M. A. & SINGH, S. P. 1996. Returnable/reusable logistical packaging: A capital budgeting investment decision framework. *Journal of Business Logistics,* 17**,** 139-165.

RUSSELL, S. J. & NORVIG, P. 2003. *Artificial Intelligence A Modern Approach,* New Jersey, Pearson Education, Inc.

SARAC, A., ABSI, N. & DAUZÈRE-PÉRÈS, S. 2010. A literature review on the impact of RFID technologies on supply chain management. *International Journal of Production Economics,* 128**,** 77-95.

SARIMVEIS, H., PATRINOS, P., TARANTILIS, C. D. & KIRANOUDIS, C. T. 2008. Dynamic modeling and control of supply chain systems: A review. *Computers & Operations Research,* 35**,** 3530-3561.

SCHMIDT, G. & WILHELM, W. E. 2000. Strategic, tactical and operational decisions in multi-national logistics networks: a review and discussion of modelling issues. *International Journal of Production Research,* 38**,** 1501-1523.

SCHNEEWEISS, C. 2003. Distributed decision making in supply chain management. *International Journal of Production Economics,* 84**,** 71-83.

SCHOLZ-REITER, B., WINDT, K. & FREITAG, M. Autonomous logistic processes: new demands and first approaches. 37th CIRP International Seminar on Manufacturing Systems, May 19–21 2004 Budapest, Hungary. 357-362.

SCHONBERGER, R. J. 2007. Japanese production management: An evolution—With mixed success. *Journal of Operations Management,* 25**,** 403-419.

SCOTT, K. 2004. *Fast Track UML 2.0,* USA, Springer-Verlag.

SEARLE, J. R. 1969. *Speech Acts: An Essay in the Philosophy of Language,* Cambridge, Cambridge University Press.

SELVIARIDIS, K. & SPRING, M. 2007. Third party logistics: a literature review and research agenda. *International Journal of Logistics Management, The,* 18**,** 125-150.

SHAD, D. 2000. Developing a theory of reverse logistics. *Interfaces,* 30**,** 143.

SHOHAM, Y. 1993. Agent-oriented programming. *Artif. Intell.,* 60**,** 51-92.

SOHAIL, M. S., ROHIT, B. & AMRIK, S. S. 2006. A comparative study on the use of third party logistics services by Singaporean and Malaysian firms. *International Journal of Physical Distribution & Logistics Management,* 36**,** 690.

SPEKMAN, R. E. & SWEENEY, P. J. 2006. RFID: from concept to implementation. *International Journal of Physical Distribution & Logistics Management,* 36.

SWAFFORD, P. M., GHOSH, S. & MURTHY, N. 2008. Achieving supply chain agility through IT integration and flexibility. *International Journal of Production Economics,* 116**,** 288-297.

SWAMINATHAN, J., M. , SMITH, S., F. & SADEH, N., M. 1998. Modeling supply chain dynamics: A multiagent approach. *Decision Sciences,* 29**,** 607.

SWEDBERG, C. 2009a. *Familia Sancela, RFID Reduces Shrinkage* [Online]. RFID Jounral. Available: http://www.rfidjournal.com/articles/view?4630 [Accessed 11/04/2013.

SWEDBERG, C. 2009b. *PLS Uses RFID to Track Pallets, Containers* [Online]. RFID Journal. Available: http://www.rfidjournal.com/article/view/5043 [Accessed 06/04/2011 2011].

SWEDBERG, C. 2009c. *Study Finds Disposable Tags Can Be Reused* [Online]. RFID Journal. Available: http://www.rfidjournal.com/article/view/5017 [Accessed 04/04/2011 2011].
SWEDBERG, C. 2010. *Axios MA Launches Tagged Pallets and Real-Time Tracking Solution* [Online]. RFID Journal. Available: http://www.rfidjournal.com/article/view/8014 [Accessed 06/04/2011 2011].
SWEDBERG, C. 2011a. *Monroe Americana Tracks Reusable Containers* [Online]. RFID Journal. Available: http://www.rfidjournal.com/article/view/8316/ [Accessed 06/04/2011 2011].
SWEDBERG, C. 2011b. *Robots and RFID help KH Lloreda Distribute Its Products* [Online]. RFID Journal. Available: http://www.rfidjournal.com/article/view/8180 [Accessed 10/03/2011.
SWEDBERG, C. 2011c. *TrakLok Offers Hybrid System for Tracking Containers* [Online]. RFID Journal. Available: http://www.rfidjournal.com/article/view/8504 [Accessed 26/08/2011.
T MOYAUX, B CHAIB-DRAA & D'AMOURS, S. 2006. Supply Chain Management and Multiagent Systems: An Overview. *In:* CHAIB-DRAA, B. & MÜLLER, J. (eds.) *Multiagent based Supply Chain Management.* Verlag Berlin Heidelberg: Springer.
TAKO, A. A. & ROBINSON, S. 2012. The application of discrete event simulation and system dynamics in the logistics and supply chain context. *Decision Support Systems,* 52.
THOMAS, H., POLLOCK, T. & GORMAN, P. 1999. Global strategic analyses: Frameworks and approaches. *The Academy of Management Executive,* 13**,** 70-82.
TRAPPEY, A. J. C., LU, T.-H. & FU, L.-D. 2009. Development of an intelligent agent system for collaborative mold production with RFID technology. *Robotics and Computer-Integrated Manufacturing,* 25**,** 42-56.
TRAPPEY, A. J. C., TRAPPEY, C. V. & WU, C. R. 2010. Genetic algorithm dynamic performance evaluation for RFID reverse logistic management. *Expert Systems with Applications,* 37**,** 7329-7335.
TWEDE, D. 1992. The Process of Logistical Packaging Innovation. *Journal of Business Logistics,* 13**,** 69-94.
TWEDE, D. & CLARKE, R. 2005. Supply Chain Issues in Reusable Packaging. *Journal of Marketing Channels,* 12**,** 7 - 26.
UM, W., LU, H. & HALL, T. 2010. A Study of Multi-Agent Based Supply Chain Modeling and Management. *iBusiness,* 2**,** 333-341.
USTUNDAG, A. & CEVIKCAN, E. 2007. Vehicle route optimization for RFID integrated waste collection system. *Proceedings of the 1st International Conference on Risk Analysis and Crisis Response,* 2**,** 155-160

933.
VALCKENAERS, P., GERMAIN, B. S., VERSTRAETE, P., VAN BELLE, J., HADELI & VAN BRUSSEL, H. 2009. Intelligent products: Agere versus Essere. *Computers in Industry,* 60**,** 217-228.
VIOLINO, B. 2003. *Linking RFID with Web services* [Online]. RFID Journal. Available: http://www.rfidjournal.com/article/purchase/606 [Accessed 11/11/2011.
VIOLINO, B. 2011. *Soft-Drink Manufacturer Finds Myriad Benefits From RFID Solution* [Online]. RFID Journal. Available: http://www.rfidjournal.com/article/purchase/8819 [Accessed 15/11/2011.
VISICH, J., K. , LI, S. & KHUMAWALA BASHEER, M. 2007. Enhancing Product Recovery Value in Closed-loop Supply Chains with RFID. *Journal of Managerial Issues,* 19**,** 436.
VOSA 2011. Rules on Driver's Hours and Tacographs: Goods Vehicles in GB and Europe. *In:* TRANSPORT, D. F. (ed.). VOSA.
WANG, M., WANG, H., VOGEL, D., KUMAR, K. & CHIU, D. K. W. 2009. Agent-based negotiation and decision making for dynamic supply chain formation. *Engineering Applications of Artificial Intelligence,* 22**,** 1046-1055.
WASSERMAN, E. 2007. *Beaver Street Fisheries Automates RFID Tagging* [Online]. RFID Journal. Available: http://www.rfidjournal.com/article/purchase/3060 [Accessed 08/03/2011 2011].

WASSERMAN, E. 2010. *Automakers Seek Fast ROI From RFID* [Online]. RFID Journal. Available: http://www.rfidjournal.com/article/purchase/7460 [Accessed 06/04/2011 2011].

WEGNER, P. 1997. Why interaction is more powerful than algorithms. *Commun. ACM,* 40**,** 80-91.

WEIMING, S. 2002. Distributed manufacturing scheduling using intelligent agents. *Intelligent Systems, IEEE,* 17**,** 88-94.

WERNZ, C. & DESHMUKH, A. 2010. Multiscale decision-making: Bridging organizational scales in systems with distributed decision-makers. *European Journal of Operational Research,* 202**,** 828-840.

WESSEL, R. 2008. *Metro Group, DHL to Roll Out RFID in France* [Online]. RFID Journal. Available: http://www.rfidjournal.com/article/view/4277/ [Accessed 7/04/2011 2011].

WINDT, K., BÖSE, F. & PHILIPP, T. 2008. Autonomy in production logistics: Identification, characterisation and application. *Robotics and Computer-Integrated Manufacturing,* 24**,** 572-578.

WOMACK, J., JONES, D. & ROOS, D. 1990. The Machine That Changed the World. Free Press.

WOOLDRIDGE, M. 2009. *An Introduction to MultiAgent Systems,* Chichester, Wiley.

WOOLDRIDGE, M. & JENNINGS, N. R. 1995. Intelligent Agents: Theory and Practice. *Knowledge Engineering Review,* 10**,** 115-152.

WU, H.-J. & DUNN, S., C. 1995. Environmentally responsible logistics systems. *International Journal of Physical Distribution & Logistics Management,* 25**,** 20-38.

XIAO, Y., YU, S., WU, K., NI, Q., JANECEK, C. & NORDSTAD, J. 2007. Radio frequency identification: technologies, applications, and research issues. *Wireless Communications and Mobile Computing,* 7**,** 457-472.

YANG, X., MOORE, P. & CHONG, S. K. 2009. Intelligent products: From lifecycle data acquisition to enabling product-related services. *Computers in Industry,* 60**,** 184-194.

YEH, K.-C., CHEN, R.-S. & CHEN, C.-C. 2011. Intelligent service-integrated platform based on the RFID technology and software agent system. *Expert Systems with Applications,* 38**,** 3058-3068.

YING, W. & DAYONG, S. 2005. Multi-agent framework for third party logistics in E-commerce. *Expert Systems with Applications,* 29**,** 431-436.

ZAEH, M. F., REINHART, G., OSTGATHE, M., GEIGER, F. & LAU, C. 2010. A holistic approach for the cognitive control of production systems. *Advanced Engineering Informatics,* 24**,** 300-307.

ZELBST, P., J. , GREEN, J., KENNETH, W. , SOWER, V., E. & BAKER, G. 2010. RFID utilization and information sharing: the impact on supply chain performance. *Journal of Business & Industrial Marketing,* 25**,** 582-589.

ZHANG, Y., HUANG, G. Q., QU, T., HO, O. & SUN, S. 2011. Agent-based smart objects management system for real-time ubiquitous manufacturing. *Robotics and Computer-Integrated Manufacturing,* 27**,** 538-549.

ZHOU, W. 2009. RFID and item-level information visibility. *European Journal of Operational Research,* 198**,** 252-258.

ZHU, X., MUKHOPADHYAY, S. K. & KURATA, H. 2012. A review of RFID technology and its managerial applications in different industries. *Journal of Engineering and Technology Management,* 29**,** 152-167.

Appendices

VOL. 2

Contents

Appendix A – Publications

A-1: Aggarwal, R. and Lim, M.K., (2013) "Enhancing competitive advantage with radio-frequency identification (RFID) enabled returnable transport equipment (RTE)", Int. J. Information Technology and Management, Vol. 12, Nos. 1/2, pp.3–26.

A-2: Aggarwal, R. and Lim, M.K., (2013) "Aligning physical and virtual logistical spheres with radio-frequency identification and agent-based modelling", Int. J. Agile Systems and Management, Vol. 6, No. 1, pp.66–82.

A-3: Aggarwal, R. and Lim, M. K., (2012), "Advances in decision support systems on the endeavour of supply chain optimality", International Conference on Industrial Logistics, Croatia, 14-16 June 2012, 472-477

A-4: Aggarwal, R. and Lim, M. K., (2012) "Improving returnable transport equipment management with an autonomous approach", 3rd International Conference on Dynamics in Logistics, 27 Feb - 1 Mar 2012

Page removed for copyright restrictions.

A-5: Nielsen, I., Lim, M., Nielsen, P. and Aggarwal, R., (2012), "A decision support system for waste collection management and its potential improvement with radio-frequency identification technology (RFID)", International Journal of Environmental Technology and Management, Vol. 15, No.s 3-6, pp. 305-324.

A-6: Aggarwal, R. and Lim, M., (2011) "Increasing competitiveness of third party logistics with RFID", Proceedings of the IEEE Int. Conf. on RFID - Technology and Applications, Barcelona, Spain, 437-444

A-7: Aggarwal, R., Lim, M. and Tan, K., (2011), ''A three-level automation approach for radio-frequency identification enabled returnable transport equipment to enhance supply chain competitiveness", 21st International Conference on Production Research, July 31 - August 4, 2011, Stuttgart, Germany, ISBN 978-3-8396-0293-5

Page removed for copyright restrictions.

A-8: Aggarwal, R. and Lim, M. K., (2010), "Utilisation of RFID to increase efficiency in the returnable equipment supply chain", Proc. of the 12th Int. Conf. on The Modern Information Technology in the Innovation Processes of the Industrial Enterprise, 30-31 Oct 2010, Aalborg Denmark, 284-292

Aston University

Page removed for copyright restrictions.

A-9: Aggarwal, R. and Lim, M. K., (2009), "Impact of RFID in reverse logistics", Proc. of the 11th Int. Conf. on The Modern Information Technology in the Innovation Processes of the Industrial Enterprise, 15-16 Oct 2009, Bergamo Italy, 185 – 192

Aston University

A-10: Aggarwal, R. and Lim, M. K., RFID Innovation: An Exploration of the Impact of RFID Technology in Outbound Logistics Accepted with correction in International Journal for RF Technologies: Research and Applications

A-11: Aggarwal, R. and Fosso Wamba, S., Transforming Logistics Service Provider's Activities Using IT: The Case of RFID Technology, In Preparation for Production Planning & Control: The Management of Operations

APPENDIX B – Project Partners: DHL Exel Supply Chain Documents

B-1: Transcript between Homebase and DHL for RTE management

B-2: A presentation discussing technology and innovation management and the development of Auto-ID and standards titled "RFID in the Logistic" by Marcel Schirmer.

B-3: Exel's exploits into RFID application in the UK

B-4: Detail’s of the logistics at Marks & Spencer Long Eaton RDC

Aston University

Page removed for copyright restrictions.

APPENDIX C – Class Diagrams for Selected Classes from the Agent Based Model

C-1: Location

Location
-name: String -batches: ArrayList<Batch>
+HashMap<Location, Integer>(connectedLocations) +toString(name:String)

C-2: Routing

Routing
+Routing: wtTransformer (double)
+getPath () +getShortestPath ()

C-3: Average Stats Writer

Average Stats Writer
-filename: String -out: BufferedWriter -date: String
+AverageStatsWriter (filename: String) +writeToFile()

C-4: Date Formatter

Date Formatter
+simpleDateFormat()

C-5: Routes Writer

Average Stats Writer
-filename: String -out: BufferedWriter -date: String
+RoutesWriter (filename: String) +writeToFile()

C-6: GUI

Average Stats Writer
-gui: GUI -frmSimulator: JFrame -numberOfTrucksField: JTextField -minTruckWeightField: JTextField -maxTruckWeightField: JTextField -minTruckVolumeField: JTextField -maxTruckShiftField: JTextField -maxTruckShiftField: JTextField -minTruckShiftField: JTextField -initialBatchesField: JTextField -batchesPerDayField: JTextField -minBatchWeightField: JTextField -maxBatchWeightField: JTextField -minBatchVolumeField: JTextField -maxBatchVolumeField: JTextField -numberOfDaysField: JTextField -minBatchQuantField: JTextField -maxBatchQuantField: JTextField -maxTruckVolumeField: JTextField -resultsDirectory: String -seedField: JTextField -distCentresField: JTextField -warehousesField: JTextField -customersField: JTextField -horizonField: JTextField

-batchProcessingField: JTextField -runButton: JButton -createNetworkButton: JButton -mntmViewNetwork: JMenuItem -extraRoadsField: JTextField -runningFrame: JFrame -finishedFrame: JFrame -minRoadTravField: JTextField -maxRoadTravField: JTextField
+run() +GUI() +initialize() +parseValues() +createFinishedFrame() +getResultDirectory()

C-7: Random Generator

RandomGenerator
-random: Random -seed: int
+initialiseWithSeed (s: int) +getRandom() +incrementSeed() +getSeed()

APPENDIX D – Java Code for Each Class

D-1: Batch Agent

```
package agents;

import java.util.ArrayList;

import routes.Route;

import control.Controller;
import control.ModelVariables;

import bidding.Auction;
import bidding.Bid;

import locations.Location;


public class Batch {
	private  String name;
	private int quantity;
	private double weight;
	private double volume;
	private int deliveryDate;
	private int returnDate;
	private int daysPostponed;
	private int daysLateOnDelivery;
	private int daysLateOnReturn;
```

```
	private Location destination;
	private Location origin;
	private Location acceptableWarehouse;
	private Location currentLocation;
	private boolean hasHalved;
	private boolean auctionPostponed = false;
	private  Bid provisionalForwardBid = null;
	private  Bid provisionalReverseBid = null;


	/**
	 * A class to model a single Batch agent. A batch agent represents a quantity of RTE with an origin, a destination, an intermediate warehouse that the
	 * batch will accept indirect delivery to, a quantity, volume, weight and preferred delivery and return dates.
	 * @param name
	 * @param origin
	 * @param destination
	 * @param acceptableWarehouse
	 * @param quantity
	 * @param volume
	 * @param weight
	 * @param deliveryDate
	 * @param returnDate
	 * @param batchValue
	 */
	public Batch(String name, Location origin, Location destination, Location acceptableWarehouse, int quantity, double volume, double weight, int deliveryDate, int returnDate) {
		this.setName(name);
		this.setQuantity(quantity);
		this.setWeight(weight);
		this.setVolume(volume);
		this.setDeliveryDate(deliveryDate);
		this.setOrigin(origin);
		this.setAcceptableWarehouse(acceptableWarehouse);
		this.setReturnDate(returnDate);
		this.setDestination(destination);
		this.setDaysPostponed(0);
		this.setDaysLateOnDelivery(0);
		this.setDaysLateOnReturn(0);
		setHasHalved(false);
	}


	/**
```

```java
         * Runs auctions according to the multi-stage process outlined in the spec.
         * @param trucks
         * @return True if the auction found an acceptable set of bids, otherwise false.
         */
        public boolean runAuction(ArrayList<Truck> trucks) {
                Auction auction = new Auction(this);

                System.out.println(Controller.getController().getDay() + " " + getDaysPostponed ()* 40 +" BatchCost " + " " +
"AuctionDay " +  " Batch " + name  + " SetDeliveryDate " +
                deliveryDate + " SetReturnDate " + returnDate + " DaysPostponed " + getDaysPostponed ()+  " DaysLateOnDelivery " +
getDaysLateOnDelivery () + " DaysLateOnReturn " + getDaysLateOnReturn () + " Origin "
                + origin + " Destination " + destination);


// control.Controller.day

                // Stage 1 - Combination bids
                Bid b = auction.combinationBids(trucks);
                if (b != null) {
                        b.getBidder().bidAccepted(b);
                        //batchValue = daysPostponed;
                        //batchValue = 0;
                        return true;

                }
                else {
                        // Stage 1 - Seek separate direct forward/reverse bids
                        Bid forwardBid = auction.forwardBids(trucks, false, false,false);
                        Bid reverseBid = auction.reverseBids(trucks, false,false,false);
                        if (forwardBid != null && reverseBid != null) {
                                forwardBid.getBidder().bidAccepted(forwardBid);
                                reverseBid.getBidder().bidAccepted(reverseBid);
                        //      batchValue = daysPostponed;
                        //      batchValue = 0;
                                return true;
                        }
                        // Stage 2d - Acceptable forward bid, seek reverse bids to warehouse
                        else if (forwardBid != null && reverseBid == null) {

                                        Bid reverseBidToWarehouse = auction.reverseBids(trucks, false, true, false);
                                        // Stage 2d contingency 1: halve weight and volume
                                        if (reverseBidToWarehouse == null && !this.hasHalved) {
                                                Batch newBatch = this.createCopy(true);
                                                Controller.getController().addSplitBatch(newBatch);
```

```
				reverseBidToWarehouse = auction.reverseBids(trucks, false, true, false);
				// Stage 2d contingency 2: open return date
/*			if (reverseBidToWarehouse == null) {
				reverseBidToWarehouse = auction.reverseBids(trucks,true,true,false);*/
					// Bidding failed, try again tomorrow
					if (reverseBidToWarehouse == null) {
						provisionalForwardBid = null;
						provisionalReverseBid = null;
						return false;
			}


			if (forwardBid != null && reverseBidToWarehouse != null) {
				// Stage 2e - acceptable forward bid and reverse bid to warehouse, seek reverse bid to 
distribution centre
				setDaysLateOnReturn(getDaysPostponed()+(reverseBidToWarehouse.getReverseDate()-
getReturnDate()));
				setReturnDate(reverseBidToWarehouse.getReverseDate());
				setProvisionalReverseBid(reverseBidToWarehouse);
				Bid reverseBidToDistCentre = auction.reverseBids(trucks, false, false, true);
				// Stage 2e contingency 1 : halve weight and volume
				if (reverseBidToDistCentre == null && !this.hasHalved) {
					Batch newBatch = this.createCopy(true);
					Controller.getController().addSplitBatch(newBatch);
					reverseBidToDistCentre = auction.reverseBids(trucks, false, false, true);
					// Stage 2e contingency 2 : open return
/*					if (reverseBidToDistCentre == null) {
						reverseBidToDistCentre = auction.reverseBids(trucks, true, false, true);*/
						// Bidding failed, try again tomorrow
						if (reverseBidToDistCentre == null) {
							provisionalForwardBid = null;
							provisionalReverseBid = null;

							return false;
					}

				if (reverseBidToDistCentre != null) {
					setDaysLateOnReturn(getDaysPostponed()+(reverseBidToDistCentre.getReverseDate()-
getReturnDate()));
					setReturnDate(reverseBidToDistCentre.getReverseDate());
					forwardBid.getBidder().bidAccepted(forwardBid);
					reverseBidToWarehouse.getBidder().bidAccepted(reverseBidToWarehouse);
					reverseBidToDistCentre.getBidder().bidAccepted(reverseBidToDistCentre);
					return true;
```

```
                    }
                    else {
                          provisionalForwardBid = null;
                          provisionalReverseBid = null;
                          return false;
                    }
              }
        }
        // Stage 2a - No acceptable forward bids, seek forward bids from distribution centre to warehouse
        if (forwardBid == null && reverseBid == null || forwardBid == null && reverseBid != null) {
              Bid forwardBidToWarehouse = auction.forwardBids(trucks, false, true, false);
              // Stage 2a contingency 1: halve weight and volume
              if (forwardBidToWarehouse == null && !this.hasHalved) {
                    Batch newBatch = this.createCopy(true);
                    Controller.getController().addSplitBatch(newBatch);
                    forwardBidToWarehouse = auction.forwardBids(trucks, false, true, false);
              }
                    // Stage 2a contingency 2: open delivery date
                    /*if (forwardBidToWarehouse == null) {
                          forwardBidToWarehouse = auction.forwardBids(trucks, true, true, false);*/
                          // Bidding failed, try again tomorrow
                          if (forwardBidToWarehouse == null) {
                                provisionalForwardBid = null;
                                provisionalReverseBid = null;
                                return false;
                          }


              // Stage 2b - Acceptable bid from distribution centre to warehouse, seek forward bids from warehouse
to customer
              if (forwardBidToWarehouse != null) {
                    setProvisionalForwardBid(forwardBidToWarehouse);
                    setDaysLateOnDelivery(getDaysPostponed()+(forwardBidToWarehouse.getForwardDate()-
getDeliveryDate()));
                    this.deliveryDate = forwardBidToWarehouse.getForwardDate();
                    Bid forwardBidToCustomer = auction.forwardBids(trucks, false, false, true);
                    // Stage 2b contingency 1: halve weight and volume
                    if (forwardBidToCustomer == null && !this.hasHalved) {
                          Batch newBatch = this.createCopy(true);
                          Controller.getController().addSplitBatch(newBatch);
                          forwardBidToCustomer = auction.forwardBids(trucks, false, false, true);
                    }
                          // Stage 2b contingency 2 : open delivery date
        /*                if (forwardBidToCustomer == null) {
```

```
				forwardBidToCustomer = auction.forwardBids(trucks, true, false, true);*/
				// Bidding failed, try again tomorrow
				if (forwardBidToCustomer == null) {
					provisionalForwardBid = null;
					provisionalReverseBid = null;
					return false;
				}


			// Stage 2c - Acceptable forward bids, seek direct reverse bids
			if (forwardBidToWarehouse != null && forwardBidToCustomer != null) {
				setDaysLateOnDelivery(getDaysPostponed()+(forwardBidToCustomer.getForwardDate()-
getDeliveryDate()));
				this.deliveryDate = forwardBidToCustomer.getForwardDate();
				this.returnDate =
forwardBidToCustomer.getForwardDate()+ModelVariables.batchProcessingDays;
				Bid directReverseBids = auction.reverseBids(trucks, false, false, false);
				// Stage 2c contingency 1: halve weight and volume
				if (directReverseBids == null && !this.hasHalved) {
					Batch newBatch = this.createCopy(true);
					Controller.getController().addSplitBatch(newBatch);
					directReverseBids = auction.reverseBids(trucks, false, false, false);
				}
					if (directReverseBids != null) {
						forwardBidToWarehouse.getBidder().bidAccepted(forwardBidToWarehouse);
						forwardBidToCustomer.getBidder().bidAccepted(forwardBidToCustomer);
						directReverseBids.getBidder().bidAccepted(directReverseBids);
						return true;
					}

			}
			// Stage 2d - Acceptable forward bids, seek reverse bids to warehouse
			if (forwardBidToWarehouse != null && forwardBidToCustomer != null) {
				setDaysLateOnDelivery(getDaysPostponed()+(forwardBidToCustomer.getForwardDate()-
getDeliveryDate()));
				this.deliveryDate = forwardBidToCustomer.getForwardDate();
				this.returnDate =
forwardBidToCustomer.getForwardDate()+ModelVariables.batchProcessingDays;
				Bid reverseBidToWarehouse = auction.reverseBids(trucks, false, true, false);
				// Stage 2b contingency 1: halve weight and volume
				if (reverseBidToWarehouse == null && !this.hasHalved) {
					Batch newBatch = this.createCopy(true);
					Controller.getController().addSplitBatch(newBatch);
					reverseBidToWarehouse = auction.reverseBids(trucks, false, true, false);
```

```
				}
					// Stage 2d contingency 2 : open delivery date
		/*			if (reverseBidToWarehouse == null) {
						reverseBidToWarehouse = auction.reverseBids(trucks,true,true,false);*/

					// Bidding failed, try again tomorrow
					if (reverseBidToWarehouse == null) {
						provisionalForwardBid = null;
						provisionalReverseBid = null;
						return false;
					}

				//Stage 2e - Acceptable forward bids and acceptable reverse bid to warehouse, seek
reverse bid to distribution centre
				if (forwardBidToWarehouse != null && forwardBidToCustomer != null &&
reverseBidToWarehouse != null) {
					setProvisionalReverseBid(reverseBidToWarehouse);
					setDaysLateOnReturn(getDaysPostponed()+(reverseBidToWarehouse.getReverseDate()-
getReturnDate()));
					this.setReturnDate(reverseBidToWarehouse.getReverseDate());
					Bid reverseBidToDistCentre = auction.reverseBids(trucks, false, false, true);
					//Stage 2e contingency 1 : halve weight and volume
					if (reverseBidToDistCentre == null && !this.hasHalved) {
						Batch newBatch = this.createCopy(true);
						Controller.getController().addSplitBatch(newBatch);
						reverseBidToDistCentre = auction.reverseBids(trucks, false, false, true);
					}
						// Stage 2e contingency 2: open return
				/*		if (reverseBidToDistCentre == null) {
						reverseBidToDistCentre = auction.reverseBids(trucks, true, false, true);*/

						//Bidding failed, try again tomorrow
						if (reverseBidToDistCentre == null) {
							provisionalForwardBid = null;
							provisionalReverseBid = null;
							return false;
						}
						// Stage 2e completed successfully, accept all offers
						else {
	setDaysLateOnDelivery(getDaysPostponed()+(forwardBidToCustomer.getForwardDate()-getDeliveryDate()));
	setDaysLateOnReturn(getDaysPostponed()+(reverseBidToDistCentre.getReverseDate()-getReturnDate()));
							this.setReturnDate(reverseBidToDistCentre.getReverseDate());
```

```
                                                                forwardBidToWarehouse.getBidder().bidAccepted(forwardBidToWarehouse);
                                                                forwardBidToCustomer.getBidder().bidAccepted(forwardBidToCustomer);
                                                                reverseBidToWarehouse.getBidder().bidAccepted(reverseBidToWarehouse);

        reverseBidToDistCentre.getBidder().bidAccepted(reverseBidToDistCentre);
                                                                return true;

                                                        }

                                                }
                                        }
                                }

                        }

                provisionalForwardBid = null;
                provisionalReverseBid = null;
                return false;

                }


                }

        /**
         * Selects the best bid that offers both forward and reverse deliveries with dates according to the batch's preferred
delivery/return dates.
         * @param auction
         * @return The acceptable bid if there is one, otherwise null.
         */
        public Bid selectBestClosedCombinationOffer(Auction auction) {
                for (Bid b : auction.getBids()) {
                        if (b.isForward() && b.isReverse() && b.getForwardDate() == getDeliveryDate() && b.getReverseDate() ==
getReturnDate()) {
                                return b;
                        }
                }
                return null;
        }

        /**
         * Selects the best bid that offers forward delivery only with a date according to the batch's preferred delivery date.
         * @param auction
         * @return The acceptable bid if there is one, otherwise null.
```

```java
     */
    public Bid selectBestClosedForwardOffer(Auction auction) {
        for (Bid b : auction.getBids()) {
            if (b.isForward() && !b.isReverse() && b.getForwardDate() == getDeliveryDate()) {
                return b;
            }
        }
        return null;
    }


    /**
     * Selects the best bid that offers reverse delivery only with a date according to the batch's preferred reverse date.
     * @param auction
     * @return The acceptable bid if there is one, otherwise null.
     */
    public Bid selectBestClosedReverseOffer(Auction auction) {
        for (Bid b : auction.getBids()) {
            if (!b.isForward() && b.isReverse() && b.getReverseDate() == getReturnDate()) {
                return b;
            }
        }
        return null;
    }

    /**
     * Selects the best bid that offers forward delivery only with the closest date to the batch's preferred delivery date.
     * @param auction
     * @return The acceptable bid if there is one, otherwise null.
     */
    public Bid selectBestOpenForwardOffer(Auction auction) {
        int forwardDate = Controller.getController().getDay()+99999;
        //int forwardDate = Batch.this.getDaysPostponed();
        Bid bestBid = null;
        for (Bid b : auction.getBids()) {
            if (b.isForward() && !b.isReverse() && b.getForwardDate() < forwardDate ) {
                bestBid = b;
                forwardDate = b.getForwardDate();
            }
        }
        return bestBid;
    }

    /**
```

```
     * Selects the best bid that offers reverse delivery only with the closest date to the batch's preferred reverse date.
     * @param auction
     * @return The acceptable bid if there is one, otherwise null.
     */
    public Bid selectBestOpenReverseOffer(Auction auction) {
        int reverseDate = Controller.getController().getDay()+99999;

        //int reverseDate = Batch.this.getDaysPostponed();
        Bid bestBid = null;
        for (Bid b : auction.getBids()) {
            if (b.isForward() && !b.isReverse() && b.getReverseDate() < reverseDate) {
                bestBid = b;
                reverseDate = b.getReverseDate();
            }
        }
        return bestBid;
    }


/*  *//**
     * Set the value of the batch.
     * @param batchValue
     *//*

    public void setBatchValue(int batchValue){
        this.batchValue = batchValue;

    }

    public int getBatchValue(){
        return batchValue;

    }*/
    /**
     * Set the quantity of the batch.
     * @param quantity
     */
    public void setQuantity(int quantity) {
        this.quantity = quantity;
    }

    /**
     * @return The quantity of the batch.
     */
```

```
public int getQuantity() {
	return quantity;
}

/**
 * Set the weight of the batch.
 * @param weight
 */
public void setWeight(double weight) {
	this.weight = weight;
}

/**
 * @return The weight of the batch.
 */
public double getWeight() {
	return weight;
}

/**
 * Set the volume of the batch.
 * @param volume
 */
public void setVolume(double volume) {
	this.volume = volume;
}

/**
 * @return The volume of the batch.
 */
public double getVolume() {
	return volume;
}

/**
 * Set the delivery date of the batch.
 * @param deliveryDate
 */
public void setDeliveryDate(int deliveryDate) {
	this.deliveryDate = deliveryDate;
}

/**
 * @return The delivery date of the batch.
```

```
 */
public int getDeliveryDate() {
	return deliveryDate;
}

/**
 * Set the return date of the batch.
 * @param returnDate
 */
public void setReturnDate(int returnDate) {
	this.returnDate = returnDate;
}

/**
 * @return The return date of the batch.
 */
public int getReturnDate() {
	return returnDate;
}

/**
 * Set the destination of the batch.
 * @param destination
 */
public void setDestination(Location destination) {
	this.destination = destination;
}


/**
 * @return The destination of the batch.
 */
public Location getDestination() {
	return destination;
}

/**
 * Set the origin of the batch.
 * @param origin
 */
public void setOrigin(Location origin) {
	this.origin = origin;
}
```

```java
	/**
	 * @return The origin of the batch.
	 */
	public Location getOrigin() {
		return origin;
	}

	/**
	 * Halve in size, weight and volume. Split will be rounded if the batch represents an odd (not even) quantity of goods
since we don't allow individual RTE to be split in half.
	 * @return The new batch.
	 */
	public Batch createCopy(boolean halve) {

		Batch b = new Batch(this.getName()+" halved
copy",this.origin,this.destination,this.acceptableWarehouse,this.quantity,this.volume,this.weight,this.deliveryDate,this.returnDa
te);
		this.setName(getName()+" halved original");
			b.setQuantity((this.getQuantity()+2-1)/2);
			this.setQuantity(this.getQuantity()-b.getQuantity());
			b.setWeight((this.getWeight()+2-1)/2);
			this.setWeight(this.getWeight()-b.getWeight());
			b.setVolume((this.getVolume()+2-1)/2);
			this.setVolume(this.getVolume()-b.getVolume());
			this.setHasHalved(true);
			b.setHasHalved(true);
		return b;
	}

	/**
	 * Sets the warehouse which the Batch will ask for delivery to if it can't get direct delivery.
	 * @param acceptableWarehouse
	 */
	public void setAcceptableWarehouse(Location acceptableWarehouse) {
		this.acceptableWarehouse = acceptableWarehouse;
	}


	/**
	 * @return The acceptable warehouse.
	 */
	public Location getAcceptableWarehouse() {
		return acceptableWarehouse;
	}
```

```java
	/**
	 * Sets whether batch has already split or not.
	 * @param hasHalved
	 */
	public void setHasHalved(boolean hasHalved) {
		this.hasHalved = hasHalved;
	}

	/**
	 * @return Whether the batch has split or not.
	 */
	public boolean isHasHalved() {
		return hasHalved;
	}


	/**
	 * The difference between the Batch's actual delivery date and the batch's original requested delivery date.
	 * @param daysLateOnDelivery
	 */
	public void setDaysLateOnDelivery(int daysLateOnDelivery) {
		this.daysLateOnDelivery = daysLateOnDelivery;
	}

	/**
	 * @return Returns the difference between the Batch's actual delivery date and the batch's original requested delivery
date.
	 */
	public int getDaysLateOnDelivery() {
		return daysLateOnDelivery;
	}

	/**
	 * The difference between the Batch's actual return date and the batch's original requested return date.
	 * @param daysLateOnReturn
	 */
	public void setDaysLateOnReturn(int daysLateOnReturn) {
		this.daysLateOnReturn = daysLateOnReturn;
	}

	/**
	 * @return Returns The difference between the Batch's actual return date and the batch's original requested return date.
	 */
```

```java
	public int getDaysLateOnReturn() {
		return daysLateOnReturn;
	}

	/**
	 * The number of days that have passed since the auction's original delivery date without the batch having secured bids.
	 * @param daysPostponed
	 */
	public void setDaysPostponed(int daysPostponed) {
		this.daysPostponed = daysPostponed;
	}

	/**
	 * @return The number of days that have passed since the auction's original delivery date without the batch having secured
bids.
	 */
	public int getDaysPostponed() {
		return daysPostponed;
	}


	/**
	 * @return Whether any days have passed since the auction's original delivery date without the batch having secured bids.
	 */
	public boolean isAuctionPostponed() {
		return auctionPostponed;
	}


	/**
	 * Set whether any days have passed since the auction's original delivery date without the batch having secured bids.
	 * @param auctionPostponed
	 */
	public void setAuctionPostponed(boolean auctionPostponed) {
		this.auctionPostponed = auctionPostponed;
	}


	/**
	 * @return The name of the batch.
	 */
	public String getName() {
		return name;
	}
```

```java
/**
 * Set the name of the batch.
 * @param name
 */
public void setName(String name) {
	this.name = name;
}

public String toString() {
	return name;
}


/**
 * @return Return the currently acceptable intermediate forward bid (i.e. to Warehouse instead of Customer)
 */
public Bid getProvisionalForwardBid() {
	return provisionalForwardBid;
}


/**
 * Set the currently acceptable intermediate forward bid.
 * @param provisionalForwardBid
 */
public void setProvisionalForwardBid(Bid provisionalForwardBid) {
	this.provisionalForwardBid = provisionalForwardBid;
}


/**
 * @return Return the currently acceptable intermediate reverse bid (i.e. to Warehouse instead of Customer)
 */
public Bid getProvisionalReverseBid() {
	return provisionalReverseBid;
}


/**
 * Set the currently acceptable intermediate reverse bid.
 * @param provisionalReverseBid
 */
```

```java
	public void setProvisionalReverseBid(Bid provisionalReverseBid) {
		this.provisionalReverseBid = provisionalReverseBid;
	}

}
```

D-2: Truck Agent

```java
package agents;

import java.util.ArrayList;

import control.Controller;
import control.ModelVariables;

import edu.uci.ics.jung.graph.UndirectedSparseGraph;

import locations.Location;
import locations.Road;

import bidding.Auction;
import bidding.Bid;

import routes.Leg;
import routes.Route;

/**
 * @author Rebecca
 *
 */
public class Truck {

	private String name;
	private double volumeCapacity;
	private double weightCapacity;
	private double currentWeight;
	private double currentVolume;
	private ArrayList<Batch> currentLoad;
	private double currentShiftLength;
	private double maxShiftLength;
	private Route[] routes;
```

```java
	private Location origin;
	private Location currentLocation;




	/**
	 * A class to represent a truck agent. Trucks have names, volume capacities, weight capacities, maximum shift lengths and
origins.
	 * @param name
	 * @param volumeCapacity
	 * @param weightCapacity
	 * @param maxShiftLength
	 * @param origin
	 */
	public Truck(String name, double volumeCapacity, double weightCapacity, double maxShiftLength,Location origin) {
		setName(name);
		setVolumeCapacity(volumeCapacity);
		setWeightCapacity(weightCapacity);
		setOrigin(origin);
		setCurrentLocation(origin);
		setCurrentWeight(100);
		setCurrentVolume(100);
		setMaxShiftLength(maxShiftLength);
		setCurrentShiftLength(0);
		routes = new Route[ModelVariables.daysAheadToPlan];
	}

	/**
	 * At day 0, populates the array with empty routes. At days past 0, adds a new route to the end of the array at each day.
	 */

	//LENGTH refers to the length of the array so it will iterate i for the value of the "days ahead to plan"
	public void setUpRoutes() {
		if (Controller.getController().getDay() == 0) {
			for (int i = 0; i < routes.length; i++) {
				routes[i] = new Route(i,this);
			}
		}
		else {
			Route[] tempRoutes = new Route[ModelVariables.daysAheadToPlan];
			for (int i = 1; i < routes.length; i++) {
				tempRoutes[i-1] = routes[i];
```

```
                    tempRoutes[tempRoutes.length-1] = new Route(Controller.getController().getDay()+tempRoutes.length-
1,this);
                    //at each iteration it will get the day of the simulation and put the temp route in.
                }
                routes = tempRoutes;
            }
        }


        /**
         * Set up routes to reflect new routes proposed in an accepted bid.
         * @param b
         */

        public void bidAccepted(Bid b) {
            if (b.isForward()) {
                routes[b.getForwardDate()-Controller.getController().getDay()] = b.getForwardRoute();
                routes[b.getForwardDate()-Controller.getController().getDay()].calculateNewDetailsForAllLegs();
            }
            if (b.isReverse()) {
                routes[b.getReverseDate()-Controller.getController().getDay()] = b.getReverseRoute();
                routes[b.getReverseDate()-Controller.getController().getDay()].calculateNewDetailsForAllLegs();
            }
            System.out.println("Bid accepted for Batch agent "+b.getBatch()+" with weight " +b.getBatch().getWeight()+" bid by
Truck "+this.toString());
            System.out.println("Delivery date: "+b.getForwardDate()+" :: Return Date: "+b.getReverseDate());


        }

        /**
         * Bid in an auction. Each (else) if statement deals with specific parameters, including whether the bid is for
forward/reverse delivery, whether the bid is open/closed
         * and whether the bid is for indirect delivery. Provisional forward/reverse routes are checked to stop a truck offering
indirect intermediate delivery on both legs on a single day;
         * in this case a direct bid makes more sense.
         * @param auction
         * @param b
         * @param forForward
         * @param forReverse
         * @param openDelivery
         * @param openReturn
         * @param intermediateToWarehouse
         * @param intermediateToDestination
```

```
         * @return
         */
        public Bid bid(Auction auction, Batch b, boolean forForward, boolean forReverse, boolean openDelivery, boolean openReturn,
boolean intermediateToWarehouse,
                        boolean intermediateToDestination) {
                Route provisionalForwardRoute = null;
                Route provisionalReverseRoute = null;
                if (b.getProvisionalForwardBid() != null && b.getProvisionalForwardBid().getBidder().equals(this)) {
                        provisionalForwardRoute = b.getProvisionalForwardBid().getForwardRoute();
                }
                if (b.getProvisionalReverseBid() != null && b.getProvisionalReverseBid().getBidder().equals(this)) {
                        provisionalReverseRoute = b.getProvisionalReverseBid().getReverseRoute();
                }
                boolean forward = false;
                int forwardDate = -1;
                boolean reverse = false;
                int reverseDate = -1;
                Route forwardRoute = null;
                Route reverseRoute = null;

                // If bidding for direct forward closed delivery
                if (!openDelivery && forForward && !intermediateToWarehouse && !intermediateToDestination ) {
                        Route r = routes[b.getDeliveryDate()-Controller.getController().getDay()];
                        if (provisionalForwardRoute != null) {
                                forward = false;
                                forwardDate = -1;
                        }
                        else {
                        forwardRoute = calculateBestRouteWithNewLeg(r, b.getOrigin(), b.getDestination(),b,true);
                        if (forwardRoute != null) {
                                forward = true;
                                forwardDate = b.getDeliveryDate();
                        }
                        }
                }

                // If bidding for indirect forward closed delivery to warehouse
                else if (!openDelivery && forForward && intermediateToWarehouse && !intermediateToDestination ) {
                        Route r = routes[b.getDeliveryDate()-Controller.getController().getDay()];


                        if (provisionalForwardRoute != null) {
                                forward = false;
                                forwardDate = -1;
```

```java
				}
				else {

					if (b.getDaysPostponed() > 0) {

					forwardRoute = calculateBestRouteWithNewLeg(r, b.getOrigin(),b.getAcceptableWarehouse(),b,true);
					if (forwardRoute != null) {
						forward = true;
						forwardDate = b.getDeliveryDate();
					}
					}


				}
			}
			// If bidding for indirect forward closed delivery to customer
			else if (!openDelivery && forForward && !intermediateToWarehouse && intermediateToDestination) {
				Route r = routes[b.getDeliveryDate()-Controller.getController().getDay()];
				if (provisionalForwardRoute != null) {
					forward = false;
					forwardDate = -1;
				}
				else {
					if (b.getDaysPostponed() > 0) {

						forwardRoute = calculateBestRouteWithNewLeg(r, 
b.getAcceptableWarehouse(),b.getDestination(),b,true);
					if (forwardRoute != null) {
						forward = true;
						forwardDate = b.getDeliveryDate();
					}

					}

				}
			}
/*			// If bidding for indirect forward open delivery to warehouse
			else if (openDelivery && forForward && intermediateToWarehouse && !intermediateToDestination) {
				for (int i = b.getDeliveryDate()-Controller.getController().getDay(); i < routes.length-
ModelVariables.batchProcessingDays; i++) {
					Route r = routes[i];
					if (provisionalForwardRoute != null && provisionalForwardRoute.getDay() != routes[i].getDay()) {
						forwardRoute = calculateBestRouteWithNewLeg(r, b.getOrigin(),b.getAcceptableWarehouse(),b,true);
						if (forwardRoute != null) {
```

```
                                        forward = true;
                                        forwardDate = r.getDay();
                                }
                        }
                }
        }
        // If bidding for indirect forward open delivery to customer
        else if (openDelivery && forForward && !intermediateToWarehouse && intermediateToDestination) {
                for (int i = b.getDeliveryDate()-Controller.getController().getDay(); i < routes.length-
ModelVariables.batchProcessingDays; i++) {
                        Route r = routes[i];
                        if (provisionalForwardRoute != null && provisionalForwardRoute.getDay() != routes[i].getDay()) {
                                forwardRoute = calculateBestRouteWithNewLeg(r, b.getAcceptableWarehouse(), b.getDestination(),
b,true);
                                if (forwardRoute != null) {
                                        forward = true;
                                        forwardDate = r.getDay();
                                        break;
                                }
                        }
                }
        }*/
        // If bidding for direct reverse closed delivery
        if (!openReturn && forReverse && !intermediateToWarehouse && !intermediateToDestination) {
                Route r = routes[(b.getReturnDate()-(Controller.getController().getDay()))];
                if (provisionalReverseRoute != null) {
                        reverse = false;
                        reverseDate = -1;
                }
                else {
                        if (b.getDaysPostponed() > 0) {

                        reverseRoute = calculateBestRouteWithNewLeg(r, b.getDestination(),b.getOrigin(),b,false);
                if (reverseRoute != null) {
                        reverse = true;
                        reverseDate = b.getReturnDate();
                }

                        }

                }
        }
        // If bidding for indirect reverse closed delivery to warehouse
        else if (!openReturn && forReverse && intermediateToWarehouse && !intermediateToDestination ){
```

```
                    Route r = routes[(b.getReturnDate()-(Controller.getController().getDay()))];
                    if (provisionalReverseRoute != null) {
                        reverse = false;
                        reverseDate = -1;
                    }
                    else {

                        if (b.getDaysPostponed() > 0) {


                        reverseRoute = calculateBestRouteWithNewLeg(r,
b.getDestination(),b.getAcceptableWarehouse(),b,false);
                    if (reverseRoute != null) {
                        reverse = true;
                        reverseDate = b.getReturnDate();
                    }

                        }



                    }
            }
            // If bidding for indirect reverse closed delivery to distribution centre
            else if (!openReturn && forReverse && !intermediateToWarehouse && intermediateToDestination){
                Route r = routes[b.getReturnDate()-Controller.getController().getDay()];
                if (provisionalReverseRoute != null) {
                    reverse = false;
                    reverseDate = -1;
                }
                else {

                    if (b.getDaysPostponed() > 0) {


                        reverseRoute = calculateBestRouteWithNewLeg(r,
b.getAcceptableWarehouse(),b.getOrigin(),b,false);
                if (reverseRoute != null) {
                    reverse = true;
                    reverseDate = b.getReturnDate();
                }


                    }
```

```
                }
            }
/*          // If bidding for indirect reverse open delivery to warehouse
            else if (openReturn && forReverse && intermediateToWarehouse && !intermediateToDestination){
                for (int i = b.getReturnDate()-Controller.getController().getDay(); i < routes.length; i++) {
                    Route r = routes[i];
                    if (provisionalReverseRoute != null && provisionalReverseRoute.getDay() != routes[i].getDay()) {
                        reverseRoute = calculateBestRouteWithNewLeg(r, b.getDestination(), b.getAcceptableWarehouse(),
b,false);
                        if (reverseRoute != null) {
                            reverse = true;
                            reverseDate = r.getDay();
                        }
                    }
                }
            }
            // If bidding for indirect reverse open delivery to distribution centre
            else if (openReturn && forReverse && !intermediateToWarehouse && intermediateToDestination){
                for (int i = b.getReturnDate()-Controller.getController().getDay(); i < routes.length; i++) {
                    Route r = routes[i];
                    if (provisionalReverseRoute != null && provisionalReverseRoute.getDay() != routes[i].getDay()) {
                        reverseRoute = calculateBestRouteWithNewLeg(r,  b.getAcceptableWarehouse(),b.getDestination(),
b,false);
                        if (reverseRoute != null) {
                            reverse = true;
                            reverseDate = r.getDay();
                        }
                    }
                }
            }*/
            if (!forward && !reverse) {
                return null;
            }
            return new Bid(this, b, forward,reverse,forwardDate,reverseDate,forwardRoute,reverseRoute);
        }

        /**
         * Finds the best place in a given route to add a new leg. Returns the new route with the shortest distance.
         * @param route
         * @param origin
         * @param destination
         * @param b
```

```java
 * @param forward
 * @return
 */
public Route calculateBestRouteWithNewLeg(Route route, Location origin, Location destination, Batch b, boolean forward) {
	double shortestLength = getMaxShiftLength()+1;
	Route bestSoFar = null;
	for (int i = 0; i <= route.getLegs().size(); i++) {
		for (int i2 = i+1; i2 <= route.getLegs().size()+1; i2++) {
		Route candidate = new Route(route.getDay(), this);
		for (Leg l : route.getLegs()) {
		candidate.addLeg(l);
		}
		Leg pickup = new Leg(origin, b, true);
		candidate.addLegAtIndex(i, pickup);

		Leg dropoff = new Leg(destination, b, false);
		candidate.addLegAtIndex(i2, dropoff);

		if (candidate.getLegs().get(candidate.getLegs().size()-1).getLocation() != getOrigin()) {
			Leg toOrigin = new Leg(getOrigin(), null, false);
			candidate.addLeg(toOrigin);

		}

		if (candidateLength < shortestLength && candidate.allLegsValid() && candidate.getLength() <
		getMaxShiftLength()) {
			candidate.allLegsValid();
			shortestLength = candidateLength;
			bestSoFar = candidate;
		}

		}


	}
	return bestSoFar;
}





/**
```

```java
 * Sets the truck's volume capacity.
 * @param volumeCapacity
 */
public void setVolumeCapacity(double volumeCapacity) {
       this.volumeCapacity = volumeCapacity;
}

/**
 * @return The truck's volume capacity.
 */
public double getVolumeCapacity() {
       return volumeCapacity;
}

/**
 * Sets the truck's weight capacity.
 * @param weightCapacity
 */
public void setWeightCapacity(double weightCapacity) {
       this.weightCapacity = weightCapacity;
}

/**
 * @return The truck's weight capacity.
 */
public double getWeightCapacity() {
       return weightCapacity;
}

/**
 * Set's the truck's current weight (currently not used, useful for a more complex scenario).
 * @param currentWeight
 */
public void setCurrentWeight(double currentWeight) {
       this.currentWeight = currentWeight;
}

/**
 * @return The truck's current weight.
 */
public double getCurrentWeight() {
       return currentWeight;
}
```

```java
/**
 * Sets the truck's current volume (currently not used, useful for a more complex scenario).
 * @param currentVolume
 */
public void setCurrentVolume(double currentVolume) {
	this.currentVolume = currentVolume;
}

/**
 * @return The truck's current volume.
 */
public double getCurrentVolume() {
	return currentVolume;
}


/**
 * Sets the truck's current shift length.
 * @param currentShiftLength
 */
public void setCurrentShiftLength(double currentShiftLength) {
	this.currentShiftLength = currentShiftLength;
}

/**
 * @return The truck's current shift length.
 */
public double getCurrentShiftLength() {
	return currentShiftLength;
}

/**
 * Sets the truck's maximum shift length.
 * @param maxShiftLength
 */
public void setMaxShiftLength(double maxShiftLength) {
	this.maxShiftLength = maxShiftLength;
}

/**
 * @return The truck's maximum shift length.
 */
public double getMaxShiftLength() {
	return maxShiftLength;
```

```java
}

/**
 * @param day
 * @return The route for a given day.
 */
public Route getRoute(int day) {
	if (day < routes.length) {
		return routes[day];
	}
	return null;
}

/**
 * Sets the truck's origin.
 * @param origin
 */
public void setOrigin(Location origin) {
	this.origin = origin;
}

/**
 * @return The truck's origin.
 */
public Location getOrigin() {
	return origin;
}

/**
 * Sets the truck's current location (currently not used, useful for more complex scenarios).
 * @param currentLocation
 */
public void setCurrentLocation(Location currentLocation) {
	this.currentLocation = currentLocation;
}

/**
 * @return The truck's current location.
 */
public Location getCurrentLocation() {
	return currentLocation;
}

/**
```

```java
 * Print out the truck's routes for today.
 */
public void printRouteForToday() {
	Route r =  routes[0];
	for (Leg l : r.getLegs()) {
		System.out.println(l.getLocation());

	}
	System.out.println(r.getLength());
}

/**
 * Print all of the truck's planned routes.
 */
public void printAllRoutes() {
	for (int i = 0; i < routes.length; i++) {
	Route r =  routes[i];
	System.out.println("Planned route for day "+(i+Controller.getController().getDay()));
	if (!r.getLegs().isEmpty()) {
	for (Leg l : r.getLegs()) {
		String loc = "Location: "+l.getLocation()+" ";
		String action = "No Action";
		if (l.isLoad() && l.getBatch() != null) {
			action = "Loading "+l.getBatch();
		}
		else if (!l.isLoad() && l.getBatch() != null) {
			action = "Unloading "+l.getBatch();
		}
		System.out.println(loc+action);
	}
	System.out.println(r.getLength());
	}
	else {
		System.out.println("None");
	}


	}

	}

/**
 * @return The truck's name.
 */
```

```
	public String getName() {
		return name;
	}

	/**
	 * Set the truck's name.
	 * @param name
	 */
	public void setName(String name) {
		this.name = name;
	}

	public String toString() {
		return name;
	}
}
```

D-3: Leg

```
package routes;

import java.util.ArrayList;
import java.util.LinkedList;

import control.Routing;

import agents.Batch;
import agents.Truck;

import locations.Location;

/**
 * A class to model one leg of a particular route. A leg involves either travel between two locations with no loading/unloading
(i.e. when returning "home" empty),
 * travel between two locations with loading/unloading (normal behaviour) or loading/unloading at a single location (i.e. when
loading multiple batches at a single location).
 *
 *
 */
public class Leg {
	private Location location;
```

```java
private double weightAtStart;
private double weightAtEnd;
private double volumeAtStart;
private double volumeAtEnd;
private LinkedList<Batch> loadAtStart;
private LinkedList<Batch> loadAtEnd;
private Batch batch;
private boolean load;

public Leg(Location location, Batch batch, boolean load) {
	this.setLocation(location);
	this.setLoad(load);
	this.setBatch(batch);
	this.loadAtStart = new LinkedList<Batch>();
	this.loadAtEnd = new LinkedList<Batch>();
	setVolumeAndWeightAtStart();
	setVolumeAndWeightAtEnd();
}

/**
 * Get the weight of the truck's batches at the end of the leg
 */
public double getWeightAtEnd() {
	return weightAtEnd;
}

/**
 * Set the weight of the truck's batches at the end of the leg
 * @param weightAtEnd
 */
public void setWeightAtEnd(double weightAtEnd) {
	this.weightAtEnd = weightAtEnd;
}

/**
 * Calculate new details for the batch.
 */
public void calculateNewDetails() {

	loadAtEnd.clear();
		for (Batch b : getLoadAtStart()) {
			loadAtEnd.add(b);
		}
		if (load && batch != null) {
```

```
                loadAtEnd.add(batch);
            }
            if (!load && batch != null) {
                loadAtEnd.remove(batch);
            }
            setVolumeAndWeightAtStart();
            setVolumeAndWeightAtEnd();
    }

    /**
     * Update the values for volume and weight at the start of the leg
     */
    public void setVolumeAndWeightAtStart() {
        for (Batch b : loadAtStart) {
            volumeAtStart = 0;
            weightAtStart = 0;
            volumeAtStart+=b.getVolume();
            weightAtStart+=b.getWeight();
        }
    }

    /**
     * Update the values for volume and weight at the end of the leg
     */
    public void setVolumeAndWeightAtEnd() {
        for (Batch b : loadAtEnd) {
            volumeAtEnd = 0;
            weightAtEnd = 0;
            volumeAtEnd+=b.getVolume();
            weightAtEnd+=b.getWeight();
        }
    }

    /**
     * @return The volume at the start of the leg
     */
    public double getVolumeAtStart() {
        double volume = 0.0;
        for (Batch b : loadAtStart) {
            volume += b.getVolume();
        }
        return volume;
    }
```

```java
/**
 * @return The volume at the end of the leg
 */
public double getVolumeAtEnd() {
	double volume = 0.0;
	for (Batch b : loadAtEnd) {
		volume += b.getVolume();
	}
	return volume;
}

/**
 * @return The weight at the start of the leg
 */
public double getWeightAtStart() {
	double weight = 0.0;
	for (Batch b: loadAtStart) {
		weight += b.getWeight();
	}
	return weight;
}


/**
 * @return The batches being carried at the start of the leg
 */
public LinkedList<Batch> getLoadAtStart() {
	return loadAtStart;
}

/**
 * Set the batches being carried at the start of the leg
 * @param loadAtStart
 */
public void setLoadAtStart(LinkedList<Batch> loadAtStart) {
	this.loadAtStart = loadAtStart;
}


/**
 * @return The batches being carried at the end of the leg, after loading
 * or unloading have been performed
 */
public LinkedList<Batch> getLoadAtEnd() {
```

```java
		return loadAtEnd;
	}

	/**
	 * Set the load at the end of the leg.
	 * @param loadAtEnd
	 */
	public void setLoadAtEnd(LinkedList<Batch> loadAtEnd) {
		this.loadAtEnd = loadAtEnd;
	}

	/**
	 * Set the destination of the leg
	 * @param location
	 */
	public void setLocation(Location location) {
		this.location = location;
	}

	/**
	 * @return The destination of the leg
	 */
	public Location getLocation() {
		return location;
	}

	/**
	 * Set whether the leg is for loading or unloading
	 * @param load
	 */
	public void setLoad(boolean load) {
		this.load = load;
	}

	/**
	 * @return Whether the leg is for loading or unloading
	 */
	public boolean isLoad() {
		return load;
	}


	/**
	 * @return The batch being loaded/unloaded in this leg
```

```
	 */
	public Batch getBatch() {
		return batch;
	}


	/**
	 * Set the batch being loaded/unloaded in this leg
	 * @param batch
	 */
	public void setBatch(Batch batch) {
		this.batch = batch;
	}
}
```

D-4: Route

```
package routes;

import java.text.DecimalFormat;
import java.util.ArrayList;

import agents.Batch;
import agents.Truck;

import control.Controller;
import control.Routing;
import locations.Location;



/**
 * A class that models a series of legs.
 *
 */
public class Route {

	private int day;
	private ArrayList<Leg> legs;
	private Truck owner;
```

```java
public Route(int day, Truck owner) {
	this.setDay(day);
	this.setOwner(owner);
	legs = new ArrayList<Leg>();
}

/**
 * Add a new leg to the route at a given index.
 * @param index
 * @param l
 */
public void addLegAtIndex(int index, Leg leg) {
	legs.add(index, leg);
}

/**
 * Update the details of all legs.
 */
public void calculateNewDetailsForAllLegs() {
	for (int i = 0; i <legs.size(); i++) {

			legs.get(i).getLoadAtStart().clear();
			legs.get(i).getLoadAtEnd().clear();

			if (i > 0) {
				for (Batch b : legs.get(i-1).getLoadAtEnd()) {
					legs.get(i).getLoadAtStart().add(b);
				}
			}
			legs.get(i).calculateNewDetails();
	}
}

/**
 * Set the day of the route.
 * @param day
 */
public void setDay(int day) {
	this.day = day;
}

/**
 * Get the day of the route
```

```java
 * @return
 */
public int getDay() {
       return day;
}


/**
 * @return Owner of route
 */
public Truck getOwner() {
       return owner;
}

/**
 * Set the owner of the route.
 * @param owner
 */
public void setOwner(Truck owner) {
       this.owner = owner;
}

/**
 * Add a leg to the end of the route
 * @param leg
 */
public void addLeg(Leg leg) {
    legs.add(leg);
}

/**
 * @return All of the legs modelled by this route
 */
public ArrayList<Leg> getLegs() {
       return legs;
}

/**
 * @return The length of this route. Each leg where loading/unloading occurs adds 1 hour to the route, plus
 * the time taken to travel between locations. For legs where no action is performed, only the travel time is
 * added.
 */
public double getLength() {
       double distanceCounter = 0.0;
```

```
		for (int i = 0; i < getLegs().size(); i++) {
			if (getLegs().get(i).getBatch() != null) {
			distanceCounter+=1.0;
			}
			if (i == 0 && getLegs().get(i).getLocation() != getOwner().getOrigin()) {
				distanceCounter+= Routing.getShortestPath(getOwner().getOrigin(), getLegs().get(i).getLocation());

			}
			if (i > 0){
				distanceCounter+= Routing.getShortestPath(getLegs().get(i-1).getLocation(),
getLegs().get(i).getLocation());
			}
		}
		DecimalFormat df = new DecimalFormat("0.00");


		return Double.valueOf(df.format(distanceCounter));
	}


	/**
	 * @return Whether, given the list of routes, whether the weight and volume remain within
	 * the truck's weight and volume capacities at all times.
	 */

	public boolean allLegsValid() {
		calculateNewDetailsForAllLegs();

		for (Leg l : legs) {
			double r = getLength ();
			Batch b = l.getBatch();

			l.setVolumeAndWeightAtStart();
			l.setVolumeAndWeightAtEnd();
			if (l.getWeightAtStart() > getOwner().getWeightCapacity()) {
				return false;
			}
			if (l.getVolumeAtStart() > getOwner().getVolumeCapacity()) {
				return false;
			}
			if (l.getWeightAtEnd() > getOwner().getWeightCapacity()) {
				return false;
			}
			if (l.getVolumeAtEnd() > getOwner().getVolumeCapacity()) {
```

```
                    return false;
                }

                if ((b != null) && (b.getDaysPostponed()== 0)){
                    return false;
                }
            }
            return true;
        }

        public String toString() {
            String route = "";
            if (!getLegs().isEmpty()) {
            for (Leg l : getLegs()) {
                route = route + l.getLocation()+" ";
                String action = "No Action // ";
                if (l.isLoad() && l.getBatch() != null) {
                    route = route + "Loading "+l.getBatch()+" // ";
                }
                else if (!l.isLoad() && l.getBatch() != null) {
                    route = route + "Unloading "+l.getBatch()+" // ";
                }
                else {
                    route = route + action;
                }
            }

            route = "AuctionDay " + Controller.getController().getDay() + owner + route + "Length: "+
                getLength() + " Cost: £"  + getLength() * 80;

            System.out.println ("Route " + route);
            }


            else {
                route = "None";

            }
            return route;

        }
}
```

D-5: Auction

```
package bidding;

import gui.GUI;

import java.util.ArrayList;
import java.util.Collections;
import java.util.HashMap;
import java.util.LinkedHashMap;
import java.util.TreeMap;

import output.AveragesStatsWriter;
import output.RoutesWriter;
import control.ModelVariables;
import control.Controller;

import random.RandomGenerator;
import routes.Route;

import agents.Batch;
import agents.Truck;

/**
 * A class to represent an auction for a particular batch. Holds an ArrayList which keeps track of the bids currently placed.
 *
 */
public class Auction {

	private Batch batch;
	private ArrayList<Bid> bids;

	public Auction(Batch batch) {
		setBatch(batch);
		bids = new ArrayList<Bid>();

}
	/**
	 * @return The batch that this auction corresponds to
	 */
	public Batch getBatch() {
		return this.batch;
	}
```

```
	/**
	 * Set the batch that this auction corresponds to
	 * @param batch
	 */
	public void setBatch(Batch batch) {
		this.batch = batch;
	}

	/**
	 * @return A randomly sorted list of all the bids received
	 */
	public ArrayList<Bid> getBids() {
		Collections.shuffle(bids,RandomGenerator.getRandom());
		return bids;
	}

	/**
	 * Accepts a bid. If the bidder has already made a bid for the same combination of forward and reverse then the bidder's 
original
	 * bid is removed and the new one is entered.
	 * @param b
	 */
	public void bid(Bid b) {
		if (b != null) {
		ArrayList<Bid> bidsToRemove = new ArrayList<Bid>();
		for (Bid bid : bids) {
			if (bid.getBidder().equals(b.getBidder()) && bid.isForward() == b.isForward() && bid.isReverse() == 
b.isReverse()) {
				bidsToRemove.add(bid);
			}
		}
		bids.removeAll(bidsToRemove);
		bids.add(b);
		}
	}


	/**
	 * Seek combination forward and reverse closed direct bids from all trucks.
	 * @param trucks
	 * @return
	 */
	public Bid combinationBids(ArrayList<Truck> trucks) {
```

```
            Bid acceptableBid = null;
            for (Truck t : trucks) {
                  bid(t.bid(this, getBatch(), true, true, false, false, false, false));
            }
            Bid comboBid = getBatch().selectBestClosedCombinationOffer(this);
            acceptableBid = comboBid;
            if (acceptableBid != null) {
            System.out.println("combo accepted");
            }
            return acceptableBid;
      }


      /**
       * Seek forward bids from all trucks according to the parameters supplied. IntermediateToWarehouse specifies that the
truck is
       * seeking delivery from the distribution centre to the warehouse; intermediateToDestination that it is seeking delivery
from
       * warehouse to customer
       * @param trucks
       * @param open
       * @param intermediateToWarehouse
       * @param intermediateToDestination
       * @return
       */
      public Bid forwardBids(ArrayList<Truck> trucks, boolean open, boolean intermediateToWarehouse, boolean
intermediateToDestination) {
            Bid acceptableBid = null;
            for (Truck t : trucks) {
                  if (!open) {
                        bid(t.bid(this, getBatch(), true, false, false, false, intermediateToWarehouse,
intermediateToDestination));
                        }
                        else {
                              bid(t.bid(this, getBatch(), true, false, true, false, intermediateToWarehouse,
intermediateToDestination));

                        }
                  if (!open) {
                        Bid forwardBid = getBatch().selectBestClosedForwardOffer(this);
                        acceptableBid = forwardBid;
                  }
                  else {
                        Bid forwardBid = getBatch().selectBestOpenForwardOffer(this);
```

```
                    acceptableBid = forwardBid;
                }
            }
            getBids().clear();
            return acceptableBid;
        }

        /**
         * Seek reverse bids from all trucks according to the parameters supplied. IntermediateToWarehouse specifies that the
truck is
         * seeking delivery from the customer to the warehouse; intermediateToDestination that it is seeking delivery from
         * warehouse to the distribution centre.
         * @param trucks
         * @param open
         * @param intermediateToWarehouse
         * @param intermediateToDestination
         * @return
         */
        public Bid reverseBids(ArrayList<Truck> trucks, boolean open, boolean intermediateToWarehouse, boolean
intermediateToDestination) {
            Bid acceptableBid = null;
            for (Truck t : trucks) {
                if (!open) {
                    bid(t.bid(this, getBatch(), false, true, false, false, intermediateToWarehouse,
intermediateToDestination));

                    }
                    else {
                        bid(t.bid(this, getBatch(), false, true, false, true, intermediateToWarehouse,
intermediateToDestination));

                    }
                if (!open) {
                    Bid reverseBid = getBatch().selectBestClosedReverseOffer(this);
                    acceptableBid = reverseBid;
                }
                else {
                    Bid reverseBid = getBatch().selectBestOpenReverseOffer(this);
                    acceptableBid = reverseBid;
                }
            }
            getBids().clear();

            return acceptableBid;
```

```
	}
}
```

D-6: Bid

```
package bidding;

import routes.Route;
import agents.Batch;
import agents.Truck;
import locations.Location;

public class Bid {
	private Truck bidder;
	private Batch batch;
	private boolean forward;
	private boolean reverse;
	private int forwardDate;
	private int reverseDate;
	private Route forwardRoute;
	private Route reverseRoute;

	/**
	 * A class to represent a bid by a given bidder. Encodes the delivery/return parameters that the bid represents.
	 * @param bidder
	 * @param batch
	 * @param forward
	 * @param reverse
	 * @param forwardDate
	 * @param reverseDate
	 * @param forwardRoute
	 * @param reverseRoute
	 */
	public Bid(Truck bidder, Batch batch, boolean forward, boolean reverse, int forwardDate, int reverseDate, Route
forwardRoute, Route reverseRoute) {
		this.setBidder(bidder);
		this.setBatch(batch);
		this.setForward(forward);
		this.setReverse(reverse);
		this.setForwardDate(forwardDate);
```

```
            this.setReverseDate(reverseDate);
            this.setForwardRoute(forwardRoute);
            this.setReverseRoute(reverseRoute);
      }

      public String toString() {
            return "Bidder: "+bidder+" Forward: "+forward+" Reverse: "+reverse+" Forward Date: "+forwardDate+" Reverse Date: 
"+reverseDate;
      }

      /**
       * Set whether the bid is for forward.
       * @param forward
       */
      public void setForward(boolean forward) {
            this.forward = forward;
      }

      /**
       * @return Whether the bid is for forward.
       */
      public boolean isForward() {
            return forward;
      }

      /**
       * Set whether the bid is for reverse.
       * @param reverse
       */
      public void setReverse(boolean reverse) {
            this.reverse = reverse;
      }

      /**
       * @return Whether the bid is for reverse
       */
      public boolean isReverse() {
            return reverse;
      }

      /**
       * Set the date for which the bid offers forward delivery
       * @param forwardDate
       */
```

```java
public void setForwardDate(int forwardDate) {
	this.forwardDate = forwardDate;
}

/**
 * @return The date for which the bid offers forward delivery
 */
public int getForwardDate() {
	return forwardDate;
}

/**
 * Set the date for which the bid offers reverse delivery
 * @param reverseDate
 */
public void setReverseDate(int reverseDate) {
	this.reverseDate = reverseDate;
}

/**
 * @return The date for which the bid offers reverse delivery
 */
public int getReverseDate() {
	return reverseDate;
}

/**
 * Set the owner of this bid
 * @param bidder
 */
public void setBidder(Truck bidder) {
	this.bidder = bidder;
}

/**
 * @return The owner of this bid
 */
public Truck getBidder() {
	return bidder;
}

/**
 * Set the proposed forward route for this bid
 * @param forwardRoute
```

```java
	 */
	public void setForwardRoute(Route forwardRoute) {
		this.forwardRoute = forwardRoute;
	}

	/**
	 * @return The proposed forward route for this bid
	 */
	public Route getForwardRoute() {
		return forwardRoute;
	}

	/**
	 * Set the proposed reverse route for this bid
	 * @param reverseRoute
	 */
	public void setReverseRoute(Route reverseRoute) {
		this.reverseRoute = reverseRoute;
	}

	/**
	 * @return The proposed reverse route for this bid
	 */
	public Route getReverseRoute() {
		return reverseRoute;
	}

	/**
	 * @return The batch that this bid is for
	 */
	public Batch getBatch() {
		return batch;
	}

	/**
	 * Set the batch that this bid is for
	 * @param batch
	 */
	public void setBatch(Batch batch) {
		this.batch = batch;
	}

}
```

D-7: Customer

```
package locations;

/**
 * Represents a Customer location
 *
 */
public class Customer extends Location {

	public Customer(String name) {
		super(name);
	}

}
```

D-8: Warehouse

```
package locations;

/**
 * Represents a warehouse.
 *
 */
public class Warehouse extends LogisticsFacility {

	public Warehouse(String name) {
		super(name);
	}
}
```

D-9: Location

```
package locations;

import java.util.ArrayList;
import java.util.HashMap;

import agents.Batch;

/**
```

```
 * Abstract superclass for all location objects
 *
 */
public abstract class Location {

	private String name;
	private ArrayList<Batch> batches;

	public Location(String name) {
		this.name = name;
		batches = new ArrayList<Batch>();
	}

	public ArrayList<Batch> getBatches() {
		return batches;
	}

	public void setBatches(ArrayList<Batch> batches) {
		this.batches = batches;
	}

	public void addBatch(Batch b) {
		batches.add(b);
	}

	public void removeBatch(Batch b){
		if (batches.contains(b)) {
			batches.remove(b);
		}
	}

	private HashMap<Location,Integer> connectedLocations;

	public HashMap<Location,Integer> getConnectedLocations() {
		return connectedLocations;
	}

	public void addConnectedLocation(Location l, int distance) {
		if (!connectedLocations.containsKey(l)) {
			connectedLocations.put(l, distance);
		}
	}

	public String toString() {
```

```
		return name;

	}



}
```

D-10: Logisitrcs Facility

```
package locations;

/**
 * Abstract superclass for all distribution centre and warehouse locations.
 *
 */
public abstract class LogisticsFacility extends Location {

	public LogisticsFacility(String name) {
		super(name);
	}

}
```

D-11: Distribution Centre

```
package locations;

/**
 * Represents a distribution centre location
 *
 */
public class DistributionCentre extends LogisticsFacility {

	public DistributionCentre(String name) {
		super(name);
	}
```

```
}
```

D-12: Road

```
package locations;

/**
 * Represents a road. Roads connect two locations, and have an attached value that specifies the time it takes to get from one
location to the other.
 *
 */
public class Road {

	private double length;

	public Road(double length) {
		this.setLength(length);
	}

	public void setLength(double length) {
		this.length = length;
	}

	public double getLength() {
		return length;
	}
```

D-13: Controller

```
package control;
import java.awt.Dimension;
import java.math.BigDecimal;
import java.math.RoundingMode;
import java.text.DecimalFormat;
import java.util.ArrayList;
import java.util.Random;

import edu.uci.ics.jung.algorithms.layout.CircleLayout;
import edu.uci.ics.jung.algorithms.layout.Layout;
import edu.uci.ics.jung.algorithms.shortestpath.DijkstraShortestPath;
import edu.uci.ics.jung.graph.*;
import edu.uci.ics.jung.visualization.BasicVisualizationServer;
```

```java
import edu.uci.ics.jung.visualization.decorators.ToStringLabeller;
import gui.GUI;

import java.util.Date;

import javax.swing.JFrame;

import org.apache.commons.collections15.Transformer;

import output.AveragesStatsWriter;
import output.RoutesWriter;

import random.RandomGenerator;
import routes.Leg;

import bidding.Auction;
import bidding.Bid;

import locations.Customer;
import locations.DistributionCentre;
import locations.Location;
import locations.Road;
import locations.Warehouse;

import agents.Batch;
import agents.Truck;


/**
 * A class to control the simulation. Responsible for setting up the network, agents and locations and running the simulation.
Also keeps track of
 * the required statistics for the output.
 *
 */
public class Controller {

	private static Controller controller = null;
	private ArrayList<Batch> batches;
	private ArrayList<Batch> batchesForTomorrow;
	private ArrayList<Batch> splitBatches;
	private ArrayList<Truck> trucks;
	private ArrayList<DistributionCentre> distributionCentres;
	private ArrayList<Warehouse> warehouses;
	private ArrayList<Customer> customers;
```

```java
private int currentTime;
private int day;
private int numberOfBatchesIntroducedToday = 0;
private int totalNumberOfAuctions = 0;
private int numberOfBatchesWithSuccessfulBids = 0;
private int numberOfBatchesWithoutSuccessfulBids = 0;
private int numberOfBatchesOnTimeForward = 0;
private int numberOfBatchesOnTimeReverse = 0;
private int numberOfBatchesLateForward = 0;
private int numberOfBatchesLateReverse = 0;
private int numberOfDaysLateOnForward = 0;
private int numberOfDaysLateOnReverse = 0;
private UndirectedSparseGraph<Location,Road> network;
private boolean usingGUI = false;

public Controller() {
	batches = new ArrayList<Batch>();
	batchesForTomorrow = new ArrayList<Batch>();
	splitBatches = new ArrayList<Batch>();
	trucks = new ArrayList<Truck>();
	distributionCentres = new ArrayList<DistributionCentre>();
	warehouses = new ArrayList<Warehouse>();
	customers = new ArrayList<Customer>();
	day = 0;
	RandomGenerator.initialiseWithSeed(ModelVariables.seed);


}

public static void main(String args[]) {
	Controller c = Controller.getController();
	c.createNetwork();
	c.start();
}

/**
 * Makes sure that there's only ever one instance of the controller. Use Controller.getController() to make
 * controller objects rather than Controller c = new Controller();
 * @return This Controller.
 */
public static Controller getController()
{
	if(controller==null){
		controller = new Controller();
```

```java
        }
        return controller;
    }

    /**
     * Sets up the trucks and batches and starts the simulation for the number of days specified in ModelVariables.
     */
    public void start() {
        createNetwork();
        setHasRun(true);
        setUpTrucks();
        setUpBatches(true);
        simulate(ModelVariables.days);
    }

    /**
     * Restarts the simulation. This is called from GUI, rather than start(), to allow multiple runs to be conducted.
     */
    public void restart() {


        trucks.clear();
        batches.clear();
        batchesForTomorrow.clear();
        splitBatches.clear();
        resetDeliveryValues();
        day = 0;

        setUpTrucks();
        setUpBatches(true);
        simulate(ModelVariables.days);
    }

    /**
     * Runs the simulation for the specified number of days.
     * @param numDays
     */
    public void simulate(int numDays) {
        // Set up stats writers
        AveragesStatsWriter avsWriter = new AveragesStatsWriter(GUI.getGUI().getResultsDirectory());
        RoutesWriter rtsWriter = new RoutesWriter(GUI.getGUI().getResultsDirectory());
        // Model days in hours to allow extensibility and split batches to be auctioned in the same day
        int hour = 0;
        // Set up routes of trucks
```

```java
for (Truck t : trucks) {
        t.setUpRoutes();
}
while (day <= numDays) {

        simulateOneStep();
        hour++;
        // Add all split batches into batches ArrayList and then clear splitBatches ArrayList
        batches.addAll(splitBatches);
        splitBatches.clear();
        // End of the day
        if (hour >= ModelVariables.maxTruckShiftLimit) {
                // Write output files and clear variables
                calculateDeliveryValues();
                avsWriter.writeToFile();
                rtsWriter.writeToFile();
                numberOfBatchesIntroducedToday = 0;
                totalNumberOfAuctions = 0;
                numberOfBatchesWithSuccessfulBids = 0;
                numberOfBatchesWithoutSuccessfulBids = 0;
                //Reset the current shift lengths of all trucks
                for (Truck t : trucks) {
                        t.setCurrentShiftLength(0);
                }
                day++;
                hour = 0;
                // Set up routes of trucks
                // and current locations of batches.
                // To see locations after batches at START of current day, move
                // call to setUpBatchLocations BEFORE call to setUpRoutes
                for (Truck t : trucks) {
                        t.setUpRoutes();
                        setUpBatchLocations(t);

                }
                for (DistributionCentre dc : distributionCentres) {
                        for (Batch b : dc.getBatches()) {
                        System.out.println(dc+" has batch "+b );
                        }
                }
                for (Warehouse w : warehouses) {
                        for (Batch b : w.getBatches()) {
                                System.out.println(w+" has batch "+b);
                        }
```

```java
				}
				for (Customer c : customers) {
					for (Batch b : c.getBatches()) {
						System.out.println(c+" has batch "+b);
					}
				}
				// Add batches that received no bids yesterday into the head of the queue today and clear ArrayList
				batches.addAll(batchesForTomorrow);
				batchesForTomorrow.clear();
				// Add new batches for today
				setUpBatches(false);
			}

		}
		// If simulation finishes and GUI is in use, create the "Finished!" window
		if (usingGUI) {
			GUI.createFInishedFrame();
		}
	}

	public void setUpBatchLocations(Truck t) {
		for (Leg l : t.getRoute(0).getLegs()) {
			if (l.getBatch() != null && l.isLoad()) {
				l.getLocation().addBatch(l.getBatch());
			}
			else if (l.getBatch() != null && !l.isLoad()) {
				l.getLocation().removeBatch(l.getBatch());
			}
		}
	}

	/**
	 * @return Whether the GUI is being used or not
	 */
	public boolean usingGUI() {
		return usingGUI;
	}

	/**
	 * Set whether the GUI is being used or not
	 * @param usingGUI
	 */
	public void setUsingGUI(boolean usingGUI) {
		this.usingGUI = usingGUI;
```

```
}

/**
 * Performs one step of the simulation
 */
public void simulateOneStep() {
	// Increase each truck's current shift length (does nothing at the moment but useful for extension)
	for (Truck t : trucks) {
		t.setCurrentShiftLength(t.getCurrentShiftLength()+1);
	}
	ArrayList<Batch> batchesToRemove = new ArrayList<Batch>();
	// Run auctions for all batches and update metrics according to their status
	for (Batch b : batches) {
		if (!b.isAuctionPostponed() && !b.isHasHalved()) {
		numberOfBatchesIntroducedToday++;
		}
		totalNumberOfAuctions++;
		// This runs the auction. If the auction returns false, update metrics and add the batch
		// into the batches for tomorrow ArrayList
		if (!b.runAuction(trucks)) {
			numberOfBatchesWithoutSuccessfulBids++;

			//add 1 to batch;______{}{}{}PL:@KLOPIPPUJ HhERE print

			addBatchForTomorrow(b);
			batchesToRemove.add(b);
		}
		// Auction succeeded.
		//
		else {
			numberOfBatchesWithSuccessfulBids++;
			batchesToRemove.add(b);
		}
	}
	// Remove all batches that had auctions in this step from the queue.
	batches.removeAll(batchesToRemove);

}

/**
 * Calculate delivery metrics for the output files.
 */
public void calculateDeliveryValues() {
	resetDeliveryValues();
```

```java
		for (Truck t : trucks) {
			for (Leg l : t.getRoute(0).getLegs()) {
				if (!l.isLoad() && l.getBatch() != null) {
						Batch b = l.getBatch();
						if (b.getDeliveryDate() == day) {
							if (b.getDaysLateOnDelivery() == 0) {
							numberOfBatchesOnTimeForward++;
							}
							else {
								numberOfBatchesLateForward++;
								numberOfDaysLateOnForward += b.getDaysLateOnDelivery();
							}
						}
						else if (b.getReturnDate() == day) {
							if (b.getDaysLateOnReturn() == 0) {
								numberOfBatchesOnTimeReverse++;
							}
							else {
								numberOfBatchesLateReverse++;
								numberOfDaysLateOnReverse += b.getDaysLateOnReturn();

							}
						}

				}
			}
		}
	}

	/**
	 * Reset delivery metrics for the output files.
	 */
	public void resetDeliveryValues() {
		numberOfBatchesOnTimeForward = 0;
		numberOfBatchesOnTimeReverse = 0;
		numberOfBatchesLateForward = 0;
		numberOfBatchesLateReverse = 0;
		numberOfDaysLateOnForward = 0;
		numberOfDaysLateOnReverse = 0;
	}

	/**
	 * @return The average total volume utilisation for all trucks.
	 */
```

```java
	public BigDecimal getAverageVolumeUtilisation() {
		BigDecimal avUtil = new BigDecimal("0.0");
		for (Truck t : trucks) {
			BigDecimal truckUtil = new BigDecimal("0.0");
			t.getRoute(0).calculateNewDetailsForAllLegs();
			// Calculate details for one truck, don't count last leg because truck will always be empty at end
			for (int i = 0; i < t.getRoute(0).getLegs().size()-1; i++) {
				Leg l = t.getRoute(0).getLegs().get(i);
				BigDecimal util = new BigDecimal(Double.toString(l.getVolumeAtEnd())).divide(new
BigDecimal(Double.toString(t.getVolumeCapacity())),3,RoundingMode.HALF_UP);
				truckUtil = truckUtil.add(util);
			}
			if (t.getRoute(0).getLegs().size() > 0) {
			truckUtil = truckUtil.divide(new BigDecimal(Integer.toString(t.getRoute(0).getLegs().size()-
1)),3,RoundingMode.HALF_UP);
			avUtil = avUtil.add(truckUtil);
			}
		}
		// Average for all trucks
		return avUtil.divide(new BigDecimal(Integer.toString(trucks.size())),3,RoundingMode.HALF_UP);
	}

	/**
	 * @return The average total weight utilisation for all trucks.
	 */
	public BigDecimal getAverageWeightUtilisation() {
		BigDecimal avUtil = new BigDecimal("0.0");
		for (Truck t : trucks) {
			BigDecimal truckUtil = new BigDecimal("0.0");
			// Calculate details for one truck, don't count last leg because truck will always be empty at end
			for (int i = 0; i < t.getRoute(0).getLegs().size()-1; i++) {
				Leg l = t.getRoute(0).getLegs().get(i);
				BigDecimal util = new BigDecimal(Double.toString(l.getWeightAtEnd())).divide(new
BigDecimal(Double.toString(t.getWeightCapacity())),3,RoundingMode.HALF_UP);
				truckUtil = truckUtil.add(util);
			}
			if (t.getRoute(0).getLegs().size() > 0) {
			truckUtil = truckUtil.divide(new BigDecimal(Integer.toString(t.getRoute(0).getLegs().size()-
1)),3,RoundingMode.HALF_UP);
			avUtil = avUtil.add(truckUtil);
			}
		}
		// Average for all trucks
		return avUtil.divide(new BigDecimal(Integer.toString(trucks.size())),3,RoundingMode.HALF_UP);
```

```java
	}

	/**
	 * @return The average total time utilisation for all trucks.
	 */
	public BigDecimal getAverageTimeUtilisation() {
		BigDecimal avUtil = new BigDecimal("0.0");
		for (Truck t : trucks) {
			if (t.getRoute(0).getLength() > 0) {
				//Calculate details for one truck
				BigDecimal util = new BigDecimal(Double.toString(t.getRoute(0).getLength())).divide(new
BigDecimal(Double.toString(t.getMaxShiftLength())),3,RoundingMode.HALF_UP);
			avUtil = avUtil.add(util);
			}
		}
		// Average for all trucks
		avUtil = avUtil.divide(new BigDecimal(Integer.toString(trucks.size())),3,RoundingMode.HALF_UP);
		return avUtil;
	}


	/**
	 * @return An array of strings containing the current day, name and route of each truck
	 */
	public String[] getTruckRoutes() {
		String[] routes = new String[trucks.size()*3];
		int truckIndex = 0;
		for (int i = 0; i < routes.length; i+=3) {
			Truck t = trucks.get(truckIndex);
			String name = t.getName();
			String route = t.getRoute(0).toString();
			routes[i] = Integer.toString(getDay());
			routes[i+1] = name;
			routes[i+2] = route;
			truckIndex++;
		}
		return routes;
	}

	/**
	 * @return Get stats for averages output
	 */
	public String[] getAveragesStats() {
		String[] results = new String[14];
```

```
        results[0] = Integer.toString(getDay());
        results[1] = getAverageVolumeUtilisation().toPlainString();
        results[2] = getAverageWeightUtilisation().toPlainString();
        results[3] = getAverageTimeUtilisation().toPlainString();
        results[4] = Integer.toString(numberOfBatchesIntroducedToday);
        results[5] = Integer.toString(totalNumberOfAuctions);
        results[6] = Integer.toString(numberOfBatchesWithSuccessfulBids);
        results[7] = Integer.toString(numberOfBatchesWithoutSuccessfulBids);
        results[8] = Integer.toString(numberOfBatchesOnTimeForward);
        results[9] = Integer.toString(numberOfBatchesOnTimeReverse);
        results[10] = Integer.toString(numberOfBatchesLateForward);
        results[11] = Integer.toString(numberOfBatchesLateReverse);
        results[12] = Integer.toString(numberOfDaysLateOnForward);
        results[13] = Integer.toString(numberOfDaysLateOnReverse);
        return results;
    }

    /**
     * Set up all the trucks in the simulation. Trucks are given a randomly chosen origin from the set of distribution
centres, and
     * randomly chosen volume, weight and shift capacities between the values specified in ModelVariables
     */
    public void setUpTrucks() {
        DecimalFormat df = new DecimalFormat("0.00");
        Random random = RandomGenerator.getRandom();
        ArrayList<Location> possibleOrigins = new ArrayList<Location>();
        possibleOrigins.addAll(distributionCentres);
        for (int i = 1; i <= ModelVariables.numberOfTrucks; i++) {
            double volumeCapacity = Double.valueOf(df.format(random.nextDouble()*(ModelVariables.maxTruckVolumeCapacity -
ModelVariables.minTruckVolumeCapacity) + ModelVariables.minTruckVolumeCapacity));
            double weightCapacity = Double.valueOf(df.format(random.nextDouble()*(ModelVariables.maxTruckWeightCapacity -
ModelVariables.minTruckWeightCapacity) + ModelVariables.minTruckWeightCapacity));
            double maxShiftLength = Double.valueOf(df.format(random.nextDouble()*(ModelVariables.maxTruckShiftLimit -
ModelVariables.minTruckShiftLimit) + ModelVariables.minTruckShiftLimit));
            int originIndex = random.nextInt(possibleOrigins.size());
            Truck t = new Truck("T"+i,volumeCapacity,weightCapacity,maxShiftLength,possibleOrigins.get(originIndex));
            trucks.add(t);
        }
    }

    /**
     * Creates a number of new batches with randomly chosen origins and destinations. Quantities, volumes and weights are
randomly set between the values specified in ModelVariables.
```

```
         * Delivery dates are randomly set between the current day and the maximum planning horizon minus the time it takes
customers to process batches. Return dates are set equal
         * to the delivery date + the number of days it takes customers to process batches
         * @param initial Whether the batches created should be of a number equal to the number of batches for initialisation in
ModelVariables
         * or the number of batches for each step in ModelVariables
         */
        public void setUpBatches(boolean initial) {
                DecimalFormat df = new DecimalFormat("0.00");
                Random random = RandomGenerator.getRandom();
                if (initial) {
                        for (int i = 0; i < ModelVariables.numberOfBatchesAtInitialisation; i++) {
                                Location origin = distributionCentres.get(random.nextInt(distributionCentres.size()));
                                Location destination = customers.get(random.nextInt(customers.size()));
                                double distanceToNearestWarehouse = 999999999999.0;
                                Location acceptableWarehouse = null;
                                for (Warehouse w : warehouses) {
                                        double distance = Routing.getShortestPath(w, destination);
                                        if (distance < distanceToNearestWarehouse) {
                                                acceptableWarehouse = w;
                                                distanceToNearestWarehouse = distance;
                                        }

                                }
                                int quantity = random.nextInt(ModelVariables.maxBatchQuantity-
ModelVariables.minBatchQuantity)+ModelVariables.minBatchQuantity;
                                double volume = Double.valueOf(df.format(random.nextDouble()*(ModelVariables.maxBatchVolume-
ModelVariables.minBatchVolume)+ModelVariables.minBatchVolume));
                                double weight = Double.valueOf(df.format(random.nextDouble()*(ModelVariables.maxBatchWeight-
ModelVariables.minBatchWeight)+ModelVariables.minBatchWeight));
                                int deliveryDate = random.nextInt((ModelVariables.daysAheadToPlan-
ModelVariables.batchProcessingDays));
                                int returnDate  = deliveryDate+ModelVariables.batchProcessingDays;
                                Batch b = new Batch("B0"+"("+i+")",origin,destination,acceptableWarehouse,quantity,volume,weight,
deliveryDate, returnDate);
                                batches.add(b);

                        }
                }

                // For non-0 steps
                else {
                        for (int i = 0; i < ModelVariables.numberOfBatchesPerDay; i++) {
                                Location origin = distributionCentres.get(random.nextInt(distributionCentres.size()));
```

```
                        Location destination = customers.get(random.nextInt(customers.size()));
                        double distanceToNearestWarehouse = 999999999999.0;
                        Location acceptableWarehouse = null;
                        for (Warehouse w : warehouses) {
                              double distance = Routing.getShortestPath(w, destination);
                              if (distance < distanceToNearestWarehouse) {
                                    acceptableWarehouse = w;
                                    distanceToNearestWarehouse = distance;
                              }
                        }
                        int quantity = random.nextInt(ModelVariables.maxBatchQuantity-
ModelVariables.minBatchQuantity)+ModelVariables.minBatchQuantity;
                        double volume = Double.valueOf(df.format(random.nextDouble()*(ModelVariables.maxBatchVolume-
ModelVariables.minBatchVolume)+ModelVariables.minBatchVolume));
                        double weight = Double.valueOf(df.format(random.nextDouble()*(ModelVariables.maxBatchWeight-
ModelVariables.minBatchWeight)+ModelVariables.minBatchWeight));
                        int deliveryDate = random.nextInt(ModelVariables.daysAheadToPlan-
ModelVariables.batchProcessingDays)+getDay();
                        int returnDate  = deliveryDate+ModelVariables.batchProcessingDays;

                        Batch b = new
Batch("B"+getDay()+"("+i+")",origin,destination,acceptableWarehouse,quantity,volume,weight, deliveryDate, returnDate);

                        batches.add(b);


                        System.out.println("B"+"("+i+")" + " Origin" + origin + " Destination" + destination + " Acceptable
Warehouse" +
                                    acceptableWarehouse + " Quantity" + quantity + " Volume" + volume + " Weight" + weight +
" Delivery Date" +
                                                deliveryDate + " Return Date" + returnDate);

                        System.out.println (" numberOfBatchesOnTimeForward " + numberOfBatchesOnTimeForward + "
numberOfBatchesOnTimeReverse " +
                                    numberOfBatchesOnTimeReverse + " numberOfBatchesLateForward " +
numberOfBatchesLateForward +
                                    " numberOfBatchesLateReverse " + numberOfBatchesLateReverse + " numberOfDaysLateOnForward
" +
                                    numberOfDaysLateOnForward + " numberOfDaysLateOnReverse " + numberOfDaysLateOnReverse);


                  }
            }
```

```java
	}


	/**
	 * Creates the network with the number of distribution centres, warehouses and customers specified in ModelVariables and
adds roads between them.
	 * A number of roads corresponding to the roadProbability in ModelVariables are first added randomly between locations.
After that, we perform
	 * cleanup to ensure that all locations have some valid route to every other location.
	 */
	public void createNetwork() {
		distributionCentres.clear();
		warehouses.clear();
		customers.clear();
		Random random = RandomGenerator.getRandom();
		network = new UndirectedSparseGraph<Location,Road>();
		for (int i = 0; i < ModelVariables.numDistCentres; i++) {
			DistributionCentre d = new DistributionCentre("DC"+(i+1));
			distributionCentres.add(d);
			network.addVertex(d);		}
		for (int i = 0; i < ModelVariables.numWarehouses; i++) {
			Warehouse w = new Warehouse("W"+(i+1));
			warehouses.add(w);
			network.addVertex(w);
		}
		for (int i = 0; i < ModelVariables.numCustomers; i++) {
			Customer c = new Customer("C"+(i+1));
			customers.add(c);
			network.addVertex(c);
		}
		// Add roads randomly between distribution centres and warehouses
		DecimalFormat df = new DecimalFormat("0.00");
		for (DistributionCentre d : distributionCentres) {
			for (Warehouse w : warehouses) {
				if (random.nextDouble() < ModelVariables.roadProbability) {
				double length = Double.valueOf(df.format(random.nextDouble()*(ModelVariables.maxRoadTravelTime-
ModelVariables.minRoadTravelTime)+ModelVariables.minRoadTravelTime));
				Road r = new Road(length);
				network.addEdge(r, d, w);
				System.out.println (" Road Length " + r.getLength() + d + w);
				}
			}
		}
		// Add roads randomly between warehouses and customers
```

```
		for (Warehouse w : warehouses) {
			for (Customer c : customers) {
				if (random.nextDouble() < ModelVariables.roadProbability) {
					double length = Double.valueOf(df.format(random.nextDouble()*(ModelVariables.maxRoadTravelTime-
ModelVariables.minRoadTravelTime)+ModelVariables.minRoadTravelTime));
				Road r = new Road(length);
				network.addEdge(r, w, c);
				System.out.println (" Road Length " + r.getLength() + w + c);
				}
			}
		}
		// Add roads from distribution centres to other distribution centres and warehouses
		// to ensure the network is fully connected
		for (DistributionCentre d : distributionCentres) {
			int distCentreNeighbours = 0;
			int warehouseNeighbours = 0;
			if (network.getNeighborCount(d) > 0) {
			for (Location l : network.getNeighbors(d)) {
				if (l instanceof DistributionCentre) {
					distCentreNeighbours++;
				}
				else if (l instanceof Warehouse) {
					warehouseNeighbours++;
				}
			}
			}
			// If distribution centre is not connected to any other distribution centres, add road to one of them
			if (distCentreNeighbours == 0 && distributionCentres.size() > 1) {
				DistributionCentre d2 = distributionCentres.get(random.nextInt(distributionCentres.size()));
				// Make sure that the distributionCentre isnt being connected to itself
				while (d2 == d) {
					d2 = distributionCentres.get(random.nextInt(distributionCentres.size()));
				}
				double length = Double.valueOf(df.format(random.nextDouble()*(ModelVariables.maxRoadTravelTime-
ModelVariables.minRoadTravelTime)+ModelVariables.minRoadTravelTime));
				Road r = new Road(length);
				network.addEdge(r, d, d2);
				System.out.println (" Road Length " + r.getLength() + d + d2);
			}
			// If distribution centre is not connected to any warehouses, add road to one of them
			if (warehouseNeighbours == 0 && warehouses.size()>0) {
				Warehouse w = warehouses.get(random.nextInt(warehouses.size()));
				double length = Double.valueOf(df.format(random.nextDouble()*(ModelVariables.maxRoadTravelTime-
ModelVariables.minRoadTravelTime)+ModelVariables.minRoadTravelTime));
```

```
                    Road r = new Road(length);
                    network.addEdge(r, d, w);
                    System.out.println (" Road Length " + r.getLength() + d + w);
                }
            }
            //  If customer is not connected to warehouse, add road to one of them
            for (Customer c : customers) {
                if (network.getNeighborCount(c) == 0 && warehouses.size() > 0) {
                    Warehouse w = warehouses.get(random.nextInt(warehouses.size()));
                    double length = Double.valueOf(df.format(random.nextDouble()*(ModelVariables.maxRoadTravelTime-
ModelVariables.minRoadTravelTime)+ModelVariables.minRoadTravelTime));
                    Road r = new Road(length);
                    network.addEdge(r, w, c);
                    System.out.println (" Road Length " + r.getLength() + w + c);
                }
            }
            for (Warehouse w : warehouses) {
                int distCentreNeighbours = 0;
                int custNeighbours = 0;
                if (network.getNeighborCount(w) > 0) {
                for (Location l : network.getNeighbors(w)) {
                    if (l instanceof DistributionCentre) {
                        distCentreNeighbours++;
                    }
                    else if (l instanceof Customer) {
                        custNeighbours++;
                    }
                }
                }
                // If warehouse is not connected to a distribution centre, add road to one of them
                if (distCentreNeighbours == 0 && distributionCentres.size() > 0) {
                    DistributionCentre d = distributionCentres.get(random.nextInt(distributionCentres.size()));
                    double length = Double.valueOf(df.format(random.nextDouble()*(ModelVariables.maxRoadTravelTime-
ModelVariables.minRoadTravelTime)+ModelVariables.minRoadTravelTime));
                    Road r = new Road(length);
                    network.addEdge(r, d, w);
                    System.out.println (" Road Length " + r.getLength() + d + w);
                }
                // If warehouse is not connected to a customer, add a road to one of them
                if (custNeighbours == 0 && customers.size() > 0) {
                    Customer c = customers.get(random.nextInt(customers.size()));
                    double length = Double.valueOf(df.format(random.nextDouble()*(ModelVariables.maxRoadTravelTime-
ModelVariables.minRoadTravelTime)+ModelVariables.minRoadTravelTime));
                    Road r = new Road(length);
```

```java
                    network.addEdge(r, w, c);
                    System.out.println (" Road Length " + r.getLength() + " " + w + "" + c);


                }

                // System.out.println (network);
        }


    }


    /**
     * Shows the network. See JUNG (available online) to see how to expand this.
     */
    public void showNetwork() {
        // The Layout<V, E> is parameterized by the vertex and edge types
        Layout<Integer, String> layout = new CircleLayout(this.network);
        layout.setSize(new Dimension(300,300)); // sets the initial size of the space
         // The BasicVisualizationServer<V,E> is parameterized by the edge types
         BasicVisualizationServer<Integer,String> vv =
                  new BasicVisualizationServer<Integer,String>(layout);
         vv.setPreferredSize(new Dimension(350,350)); //Sets the viewing area size
         vv.getRenderContext().setVertexLabelTransformer(new ToStringLabeller());
         JFrame frame = new JFrame("Network");
         frame.getContentPane().add(vv);
         frame.pack();
         frame.setVisible(true);


    }


    /**
     * @return The network
     */
    public UndirectedSparseGraph<Location,Road> getNetwork() {
            return network;
    }

    /**
     * @return The current day of the simulation
     */
```

```java
public int getDay() {
        return day;
}

/**
 * Set the current day of the simulation
 * @param day
 */
public void setDay(int day) {
        this.day = day;
}

/**
 * @return The ArrayList of trucks
 */
public ArrayList<Truck> getTrucks() {
        return trucks;
}

/**
 * Add a batch that has split in half into the splitBatches ArrayList
 * @param b
 */
public void addSplitBatch(Batch b) {
        if (!splitBatches.contains(b)) {
                splitBatches.add(b);
        }
}

/**
 * Add a batch that had a failed auction into the batches for tomorrow
 * @param b
 */
public void addBatchForTomorrow(Batch b) {
        if (b.getDeliveryDate() <= getDay()) {
                b.setDeliveryDate(b.getDeliveryDate()+1);
                b.setDaysPostponed(b.getDaysPostponed()+1);
                b.setReturnDate(b.getReturnDate()+1);

        }
        b.setAuctionPostponed(true);
        batchesForTomorrow.add(b);
}
```

```
}
```

D-14: Routing

```
package control;

import java.text.DecimalFormat;
import java.util.List;

import locations.Location;
import locations.Road;

import org.apache.commons.collections15.Transformer;

import edu.uci.ics.jung.algorithms.shortestpath.DijkstraShortestPath;

/**
 * A class to calculate the shortest path between a pair of locations
 *
 */
public class Routing {

	public static Transformer<Road,Double> getTransformer() {
		Transformer<Road, Double> wtTransformer = new Transformer<Road,Double>() {
		 public Double transform(Road road) {
		     return road.getLength();
		 }
	    };
	    return wtTransformer;
	}

	/**
	 * Calculate the shortest distance between two locations
	 * @param start
	 * @param finish
	 * @return
	 */
	public static double getShortestPath(Location start, Location finish) {
	DecimalFormat df = new DecimalFormat("0.00");
```

```java
	DijkstraShortestPath<Location,Road> alg = new DijkstraShortestPath<Location,Road>(Controller.getController().getNetwork(),
getTransformer());
	double distance = Double.valueOf(df.format(alg.getDistance(start, finish)));
	return distance;
	}

	/**
	 * Calculate the shortest route between two locations
	 * @param start
	 * @param finish
	 * @return
	 */
	public static List<Road> getPath(Location start, Location finish) {
		DijkstraShortestPath<Location,Road> alg = new
DijkstraShortestPath<Location,Road>(Controller.getController().getNetwork(), getTransformer());

		return alg.getPath(start, finish);
		}

}
```

D-15: Model Variables

```java
package control;

/**
 * A class to hold the parameters used in the simulation.
 *
 */
public class ModelVariables {

	public static int days = 15;
	public static int seed = 123;
	public static int numDistCentres = 1;
	public static int numWarehouses = 2;
	public static int numCustomers = 5;
	public static double roadProbability = 0.1;
	public static int numberOfTrucks = 5; //15;
	public static double minRoadTravelTime = 0.1;
	public static double maxRoadTravelTime = 1.0;
	public static double minTruckVolumeCapacity = 500.0;
	public static double maxTruckVolumeCapacity = 500.0; //1200.0;
	public static double minTruckWeightCapacity = 300.0;
```

```
	public static double maxTruckWeightCapacity = 300.0;
	public static double minTruckShiftLimit = 9.0;
	public static double maxTruckShiftLimit = 9.0;
	public static int daysAheadToPlan = 10;
	public static int numberOfBatchesAtInitialisation = 5;
	public static int numberOfBatchesPerDay = 15; //10;
	public static int minBatchQuantity = 20;
	public static int maxBatchQuantity = 60;
	public static double minBatchWeight = 15.0;
	public static double maxBatchWeight = 45.0;
	public static double minBatchVolume = 20.0; //200.0;
	public static double maxBatchVolume = 50.0;
	public static int batchProcessingDays = 2;


		}
```

D-16: Average Stats Writer

```
package output;

import java.io.BufferedWriter;
import java.io.FileWriter;
import java.util.Calendar;

import random.RandomGenerator;

import control.Controller;

public class AveragesStatsWriter
{
	private String filename;
	private BufferedWriter out;
	private String date;

	/**
	 * Writes the averages stats at each step.
	 * @param filename The file to write to.
	 */
	public AveragesStatsWriter(String filename)
	{
		this.filename = filename;
```

```java
        try {
                date = DateFormatter.getFormat().format(Calendar.getInstance().getTime())+"average results.csv";
                out = new BufferedWriter(new FileWriter(filename+date, true));
                out.write("Day");
                out.write(",");
                out.write("Average Truck Volume Utilisation");
                out.write(",");
                out.write("Average Truck Weight Utilisation");
                out.write(",");
                out.write("Average Truck Time Utilisation");
                out.write(",");
                out.write("Number of New Batches Introduced");
                out.write(",");
                out.write("Total Number of Auctions");
                out.write(",");
                out.write("Number of Batches With Successful Bids");
                out.write(",");
                out.write("Number of Batches Without a Successful Bid");
                out.write(",");
                out.write("Number of Batches Delivered On Time (Forward)");
                out.write(",");
                out.write("Number of Batches Delivered On Time (Reverse)");
                out.write(",");
                out.write("Number of Batches Delivered Late (Forward)");
                out.write(",");
                out.write("Number of Batches Delivered Late (Reverse)");
                out.write(",");
                out.write("Total Days Late on Delivered Items");
                out.write(",");
                out.write("Total Days Late on Returned Items");
                out.write(",");
                out.write("Seed");
                out.flush();
                out.newLine();
                out.close();
        }
        catch(Exception e)
        {
                e.printStackTrace();
        }
}
```

```java
public void writeToFile()
{
	String[] results = Controller.getController().getAveragesStats();
	try {
		out = new BufferedWriter(new FileWriter(filename+date, true));
		out.write(results[0]);
		out.write(',');
		out.write(results[1]);
		out.write(',');
		out.write(results[2]);
		out.write(',');
		out.write(results[3]);
		out.write(',');
		out.write(results[4]);
		out.write(',');
		out.write(results[5]);
		out.write(',');
		out.write(results[6]);
		out.write(',');
		out.write(results[7]);
		out.write(',');
		out.write(results[8]);
		out.write(',');
		out.write(results[9]);
		out.write(',');
		out.write(results[10]);
		out.write(',');
		out.write(results[11]);
		out.write(',');
		out.write(results[12]);
		out.write(',');
		out.write(results[13]);
		out.write(',');
		out.write(Integer.toString(RandomGenerator.getSeed()));
		out.write(',');
		out.newLine();
		out.close();
	}
	catch(Exception e)
	{
		e.printStackTrace();
	}
}
```

```
}
```

D-17: Date Formatter

```
package output;

import java.text.SimpleDateFormat;

/**
 * A class to format the dates attached to each output file.
 *
 */
public class DateFormatter {

	public static SimpleDateFormat getFormat() {
		SimpleDateFormat sdf = new SimpleDateFormat("ddMM HHmmss");
		return sdf;
	}
}
```

D-18: Routes Writer

```
package output;

import java.io.BufferedWriter;
import java.io.FileWriter;
import java.util.Calendar;

import random.RandomGenerator;

import control.Controller;

public class RoutesWriter
{
	private String filename;
	private BufferedWriter out;
	private String date;

	/**
	 * A class to write the routes of each truck at each day to an output file.
	 */
	public RoutesWriter(String filename)
```

```
{
        this.filename = filename;
        try {
                date = DateFormatter.getFormat().format(Calendar.getInstance().getTime())+"truck routes.csv";
                out = new BufferedWriter(new FileWriter(filename+date, true));
                out.write("Day");
                out.write(",");
                out.write("Truck Name");
                out.write(",");
                out.write("Truck Route");
                out.write(",");
                out.flush();
                out.newLine();
                out.close();
        }
        catch(Exception e)
        {
                e.printStackTrace();
        }
}


public void writeToFile()
{
        String[] results = Controller.getController().getTruckRoutes();
        try {
                out = new BufferedWriter(new FileWriter(filename+date, true));
                for (int i = 0; i < results.length; i+=3) {
                        out.write(results[i]);
                        out.write(',');
                        out.write(results[i+1]);
                        out.write(',');
                        out.write(results[i+2]);
                        out.write(',');
                        out.newLine();
                }
                out.close();
        }
        catch(Exception e)
        {
                e.printStackTrace();
        }
}
```

```
}
```

D-19: GUI

```
package gui;

import java.awt.EventQueue;

import javax.swing.JFrame;
import java.awt.BorderLayout;
import java.awt.FlowLayout;
import java.awt.GridBagLayout;
import java.awt.GridLayout;
import javax.swing.JComboBox;
import javax.swing.JPanel;
import java.awt.GridBagConstraints;
import javax.swing.DefaultComboBoxModel;
import javax.swing.JFileChooser;
import javax.swing.JScrollPane;
import javax.swing.JTabbedPane;
import javax.swing.JButton;
import javax.swing.UIManager;
import javax.swing.UnsupportedLookAndFeelException;

import java.awt.Insets;
import javax.swing.JToolBar;
import javax.swing.JMenuBar;
import java.awt.ScrollPane;
import java.awt.Label;
import javax.swing.AbstractAction;
import java.awt.event.ActionEvent;
import javax.swing.Action;
import java.awt.event.ActionListener;
import javax.swing.JScrollBar;
import javax.swing.JSeparator;
import java.awt.Button;
import java.awt.Dimension;
import javax.swing.JLabel;
import javax.swing.JTextField;

import control.Controller;
```

```java
import control.ModelVariables;

import javax.swing.SwingConstants;
import javax.swing.JTextArea;
import javax.swing.border.LineBorder;
import java.awt.Color;
import javax.swing.JMenuItem;
import javax.swing.JMenu;
import java.awt.Toolkit;

/**
 * A class to allow user input of the simulation values and running of the simulation.
 *
 */
public class GUI {

	private static GUI gui = null;
	private JFrame frmSimulator;
	private JTextField numberOfTrucksField;
	private JTextField minTruckWeightField;
	private JTextField maxTruckWeightField;
	private JTextField minTruckVolumeField;
	private JTextField maxTruckShiftField;
	private JTextField minTruckShiftField;
	private JTextField initialBatchesField;
	private JTextField batchesPerDayField;
	private JTextField minBatchWeightField;
	private JTextField maxBatchWeightField;
	private JTextField minBatchVolumeField;
	private JTextField maxBatchVolumeField;
	private JTextField numberOfDaysField;
	private JTextField minBatchQuantField;
	private JTextField maxBatchQuantField;
	private JTextField maxTruckVolumeField;
	private String resultsDirectory = "";
	private JTextField seedField;
	private JTextField distCentresField;
	private JTextField warehousesField;
	private JTextField customersField;
	private JTextField horizonField;
	private JTextField batchProcessingField;
	private JButton runButton;
	private JButton createNetworkButton;
	private JMenuItem mntmViewNetwork;
```

```
    private JTextField extraRoadsField;
    private static JFrame runningFrame;
    private static JFrame finishedFrame;
    private JTextField minRoadTravField;
    private JTextField maxRoadTravField;

    /**
     * Launch the application.
     */
    public static void main(String[] args) {
            EventQueue.invokeLater(new Runnable() {
                    public void run() {
                            try {
                                    GUI window = GUI.getGUI();
                                    window.frmSimulator.setVisible(true);
                            } catch (Exception e) {
                                    e.printStackTrace();
                            }
                    }
            });
    }


    /**
     * Create the application.
     */
    public GUI() {
        try {
         // Set cross-platform Java L&F (also called "Metal")
     UIManager.setLookAndFeel(
         UIManager.getSystemLookAndFeelClassName());
 }
        catch (UnsupportedLookAndFeelException e) {
           // handle exception
         }
         catch (ClassNotFoundException e) {
           // handle exception
         }
         catch (InstantiationException e) {
           // handle exception
         }
         catch (IllegalAccessException e) {
           // handle exception
         }
```

```
        initialize();
    }

    public static GUI getGUI() {
        if(gui==null){
            gui = new GUI();
        }
        return gui;
    }

    public static boolean guiInUse() {
        if (gui == null) {
            return false;
        }
        else return true;
    }

    /**
     * Initialize the contents of the frame.
     */
    private void initialize() {
        frmSimulator = new JFrame();
        frmSimulator.setTitle("Simulator");
        frmSimulator.setMinimumSize(new Dimension(800, 400));
        frmSimulator.setPreferredSize(new Dimension(500, 500));
        frmSimulator.setBounds(100, 100, 700, 400);
        frmSimulator.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frmSimulator.getContentPane().setLayout(new BorderLayout(0, 0));

        JPanel panel_1 = new JPanel();
        frmSimulator.getContentPane().add(panel_1, BorderLayout.SOUTH);
        panel_1.setLayout(new GridLayout(1, 0, 0, 0));

        runButton = new JButton("Run Simulation");
        runButton.setEnabled(false);
        runButton.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent arg0) {
                Thread t = new Thread(new Runnable() {
                    public void run() {
                        parseValues();
                        runningFrame = new JFrame("Running");
                        runningFrame.setPreferredSize(new Dimension(150, 85));
                        runningFrame.setBounds(250, 250, 671, 235);
                        runningFrame.getContentPane().setLayout(new BorderLayout());
```

```
                        JLabel runningLabel = new JLabel("Running...");
                        runningLabel.setHorizontalAlignment(SwingConstants.CENTER);
                        JButton okButton = new JButton("OK");
                        okButton.addActionListener(new ActionListener() {
                              public void actionPerformed(ActionEvent arg0) {
                                    runningFrame.setVisible(false);
                              }
                        });
                        runningFrame.getContentPane().add(runningLabel,BorderLayout.CENTER);
                        runningFrame.getContentPane().add(okButton,BorderLayout.SOUTH);
                        runningFrame.pack();
                        runningFrame.setVisible(true);
                        Controller.getController().setUsingGUI(true);
                        Controller.getController().restart();
                        }
            });
            t.start();
            }
});

createNetworkButton = new JButton("Create Network");
createNetworkButton.addActionListener(new ActionListener() {
      public void actionPerformed(ActionEvent arg0) {
            parseValues();
            Controller.getController().createNetwork();
            runButton.setEnabled(true);
            mntmViewNetwork.setEnabled(true);
      }
});
panel_1.add(createNetworkButton);
panel_1.add(runButton);

JPanel panel_2 = new JPanel();
frmSimulator.getContentPane().add(panel_2, BorderLayout.CENTER);
panel_2.setLayout(new GridLayout(1, 2, 0, 0));

JPanel trucksPanel = new JPanel();
trucksPanel.setBorder(new LineBorder(new Color(0, 0, 0)));
panel_2.add(trucksPanel);
trucksPanel.setLayout(new BorderLayout(0, 0));

JPanel trucksFieldsPanel = new JPanel();
trucksPanel.add(trucksFieldsPanel);
trucksFieldsPanel.setLayout(new GridLayout(0,2,0,0));
```

```
            JLabel numberOfTrucksLabel = new JLabel("Number of trucks");
            numberOfTrucksLabel.setToolTipText("The number of trucks in the simulation. Must be a whole number.");
            numberOfTrucksLabel.setHorizontalAlignment(SwingConstants.CENTER);
            trucksFieldsPanel.add(numberOfTrucksLabel);

            numberOfTrucksField = new JTextField(Integer.toString(ModelVariables.numberOfTrucks));
            trucksFieldsPanel.add(numberOfTrucksField);
            numberOfTrucksField.setColumns(10);

            JLabel planningHorizonLabel = new JLabel("Planning horizon (days)");
            planningHorizonLabel.setToolTipText("The amount of days that trucks plan their routes ahead for. Must be a whole
number.");
            planningHorizonLabel.setHorizontalAlignment(SwingConstants.CENTER);
            trucksFieldsPanel.add(planningHorizonLabel);

            horizonField = new JTextField(Integer.toString(ModelVariables.daysAheadToPlan));
            trucksFieldsPanel.add(horizonField);
            horizonField.setColumns(10);

            JLabel minTruckWeightLabel = new JLabel("Minimum truck weight capacity");
            minTruckWeightLabel.setToolTipText("Trucks are created with random weight capacities. This is the minimum possible
weight capacity.");
            minTruckWeightLabel.setHorizontalAlignment(SwingConstants.CENTER);
            trucksFieldsPanel.add(minTruckWeightLabel);

            minTruckWeightField = new JTextField(Double.toString(ModelVariables.minTruckWeightCapacity));
            trucksFieldsPanel.add(minTruckWeightField);
            minTruckWeightField.setColumns(10);

            JLabel maxTruckWeightLabel = new JLabel("Maximum truck weight capacity");
            maxTruckWeightLabel.setToolTipText("Trucks are created with random weight capacities. This is the maximum possible
weight capacity.");
            maxTruckWeightLabel.setHorizontalAlignment(SwingConstants.CENTER);
            trucksFieldsPanel.add(maxTruckWeightLabel);

            maxTruckWeightField = new JTextField(Double.toString(ModelVariables.maxTruckWeightCapacity));
            trucksFieldsPanel.add(maxTruckWeightField);
            maxTruckWeightField.setColumns(10);

            JLabel minTruckVolumeLabel = new JLabel("Minimum truck volume capacity");
            minTruckVolumeLabel.setToolTipText("Trucks are created with random volume capacities. This is the minimum possible
volume capacity.");
            minTruckVolumeLabel.setHorizontalAlignment(SwingConstants.CENTER);
```

```
            trucksFieldsPanel.add(minTruckVolumeLabel);

            minTruckVolumeField = new JTextField(Double.toString(ModelVariables.minTruckVolumeCapacity));
            trucksFieldsPanel.add(minTruckVolumeField);
            minTruckVolumeField.setColumns(10);

            JLabel maxTruckVolumeLabel = new JLabel("Maximum truck volume capacity");
            maxTruckVolumeLabel.setToolTipText("Trucks are created with random volume capacities. This is the maximum possible
volume capacity.");
            maxTruckVolumeLabel.setHorizontalAlignment(SwingConstants.CENTER);
            trucksFieldsPanel.add(maxTruckVolumeLabel);

            maxTruckVolumeField = new JTextField(Double.toString(ModelVariables.maxTruckVolumeCapacity));
            trucksFieldsPanel.add(maxTruckVolumeField);
            maxTruckVolumeField.setColumns(10);

            JLabel minTruckShiftLabel = new JLabel("Minimum truck shift length limit");
            minTruckShiftLabel.setToolTipText("Trucks are created with randomly chosen shift lengths. This is the minimum
possible shift length.");
            minTruckShiftLabel.setHorizontalAlignment(SwingConstants.CENTER);
            trucksFieldsPanel.add(minTruckShiftLabel);

            minTruckShiftField = new JTextField(Double.toString(ModelVariables.minTruckShiftLimit));
            trucksFieldsPanel.add(minTruckShiftField);
            minTruckShiftField.setColumns(10);

            JLabel maxTruckShiftLabel = new JLabel("Maximum truck shift length limit");
            maxTruckShiftLabel.setToolTipText("Trucks are created with randomly chosen shift lengths. This is the maximum
possible shift length.");
            maxTruckShiftLabel.setHorizontalAlignment(SwingConstants.CENTER);
            trucksFieldsPanel.add(maxTruckShiftLabel);

            maxTruckShiftField = new JTextField(Double.toString(ModelVariables.maxTruckShiftLimit));
            trucksFieldsPanel.add(maxTruckShiftField);
            maxTruckShiftField.setColumns(10);

            JLabel lblTrucks = new JLabel("Trucks");
            lblTrucks.setHorizontalAlignment(SwingConstants.CENTER);
            trucksPanel.add(lblTrucks, BorderLayout.NORTH);

            JPanel batchesPanel = new JPanel();
            batchesPanel.setBorder(new LineBorder(new Color(0, 0, 0)));
            panel_2.add(batchesPanel);
            batchesPanel.setLayout(new BorderLayout(0, 0));
```

```java
            JPanel batchesFieldsPanel = new JPanel();
            batchesPanel.add(batchesFieldsPanel, BorderLayout.CENTER);
            batchesFieldsPanel.setLayout(new GridLayout(0, 2, 0, 0));

            JLabel initialBatchesLabel = new JLabel("Initial number of batches");
            initialBatchesLabel.setToolTipText("The number of batches to introduce on the first day of the simulation. Must be a
whole number.");
            initialBatchesLabel.setHorizontalAlignment(SwingConstants.CENTER);
            batchesFieldsPanel.add(initialBatchesLabel);

            initialBatchesField = new JTextField(Integer.toString(ModelVariables.numberOfBatchesAtInitialisation));
            initialBatchesField.setColumns(10);
            batchesFieldsPanel.add(initialBatchesField);

            JLabel batchesPerDayLabel = new JLabel("Number of new batches per day");
            batchesPerDayLabel.setToolTipText("The number of new batches to introduce per day. Must be a whole number.");
            batchesPerDayLabel.setHorizontalAlignment(SwingConstants.CENTER);
            batchesFieldsPanel.add(batchesPerDayLabel);

            batchesPerDayField = new JTextField(Integer.toString(ModelVariables.numberOfBatchesPerDay));
            batchesPerDayField.setColumns(10);
            batchesFieldsPanel.add(batchesPerDayField);

            JLabel batchProcessingLabel = new JLabel("Batch processing time (days)");
            batchProcessingLabel.setToolTipText("The amount of time it takes a customer to process a batch before it is ready
for return.");
            batchProcessingLabel.setHorizontalAlignment(SwingConstants.CENTER);
            batchesFieldsPanel.add(batchProcessingLabel);

            batchProcessingField = new JTextField(Integer.toString(ModelVariables.batchProcessingDays));
            batchesFieldsPanel.add(batchProcessingField);
            batchProcessingField.setColumns(10);

            JLabel minBatchQuantLabel = new JLabel("Minimum batch quantity");
            minBatchQuantLabel.setToolTipText("Batches are created with randomly chosen quantities. This is the minimum possible
quantity.");
            minBatchQuantLabel.setHorizontalAlignment(SwingConstants.CENTER);
            batchesFieldsPanel.add(minBatchQuantLabel);

            minBatchQuantField = new JTextField(Integer.toString(ModelVariables.minBatchQuantity));
            batchesFieldsPanel.add(minBatchQuantField);
            minBatchQuantField.setColumns(10);
```

```java
        JLabel maxBatchQuantLabel = new JLabel("Maximum batch quantity");
        maxBatchQuantLabel.setToolTipText("Batches are created with randomly chosen quantities. This is the maximum possible
quantity.");
        maxBatchQuantLabel.setHorizontalAlignment(SwingConstants.CENTER);
        batchesFieldsPanel.add(maxBatchQuantLabel);

        maxBatchQuantField = new JTextField(Integer.toString(ModelVariables.maxBatchQuantity));
        batchesFieldsPanel.add(maxBatchQuantField);
        maxBatchQuantField.setColumns(10);

        JLabel minBatchWeightLabel = new JLabel("Minimum batch weight");
        minBatchWeightLabel.setToolTipText("Batches are created with randomly chosen weights. This is the minimum possible
weight");
        minBatchWeightLabel.setHorizontalAlignment(SwingConstants.CENTER);
        batchesFieldsPanel.add(minBatchWeightLabel);

        minBatchWeightField = new JTextField(Double.toString(ModelVariables.minBatchWeight));
        minBatchWeightField.setColumns(10);
        batchesFieldsPanel.add(minBatchWeightField);

        JLabel maxBatchWeightLabel = new JLabel("Maximum batch weight");
        maxBatchWeightLabel.setToolTipText("Batches are created with randomly chosen weights. This is the maximum possible
weight");
        maxBatchWeightLabel.setHorizontalAlignment(SwingConstants.CENTER);
        batchesFieldsPanel.add(maxBatchWeightLabel);

        maxBatchWeightField = new JTextField(Double.toString(ModelVariables.maxBatchWeight));
        maxBatchWeightField.setColumns(10);
        batchesFieldsPanel.add(maxBatchWeightField);

        JLabel minBatchVolumeLabel = new JLabel("Minimum batch volume");
        minBatchVolumeLabel.setToolTipText("Batches are created with randomly chosen volumes. This is the minimum possible
volume.");
        minBatchVolumeLabel.setHorizontalAlignment(SwingConstants.CENTER);
        batchesFieldsPanel.add(minBatchVolumeLabel);

        minBatchVolumeField = new JTextField(Double.toString(ModelVariables.minBatchVolume));
        minBatchVolumeField.setColumns(10);
        batchesFieldsPanel.add(minBatchVolumeField);

        JLabel maxBatchVolumeLabel = new JLabel("Maximum batch volume");
        maxBatchVolumeLabel.setToolTipText("Batches are created with randomly chosen volumes. This is the maximum possible
volume");
        maxBatchVolumeLabel.setHorizontalAlignment(SwingConstants.CENTER);
```

```java
            batchesFieldsPanel.add(maxBatchVolumeLabel);

            maxBatchVolumeField = new JTextField(Double.toString(ModelVariables.maxBatchVolume));
            maxBatchVolumeField.setPreferredSize(new Dimension(6, 6));
            maxBatchVolumeField.setColumns(10);
            batchesFieldsPanel.add(maxBatchVolumeField);

            JLabel batchesLabel = new JLabel("Batches");
            batchesLabel.setHorizontalAlignment(SwingConstants.CENTER);
            batchesPanel.add(batchesLabel, BorderLayout.NORTH);

            JPanel simulationPanel = new JPanel();
            simulationPanel.setBorder(new LineBorder(new Color(0, 0, 0)));
            frmSimulator.getContentPane().add(simulationPanel, BorderLayout.NORTH);
            simulationPanel.setLayout(new BorderLayout(0, 0));

            JPanel simulationFieldPanel = new JPanel();
            simulationPanel.add(simulationFieldPanel, BorderLayout.CENTER);
            simulationFieldPanel.setLayout(new GridLayout(0, 2, 0, 0));

            JLabel numberOfDaysLabel = new JLabel("Number of days");
            numberOfDaysLabel.setToolTipText("Number of days to run the simulation for. Must be a whole number.");
            numberOfDaysLabel.setHorizontalAlignment(SwingConstants.CENTER);
            simulationFieldPanel.add(numberOfDaysLabel);

            numberOfDaysField = new JTextField(Integer.toString(ModelVariables.days));
            simulationFieldPanel.add(numberOfDaysField);
            numberOfDaysField.setColumns(10);

            JLabel seedLabel = new JLabel("Random seed");
            seedLabel.setToolTipText("The simulation seed. Simulations with identical parameters and the same seed with produce
identical results. Must be a whole number.");
            seedLabel.setHorizontalAlignment(SwingConstants.CENTER);
            simulationFieldPanel.add(seedLabel);

            seedField = new JTextField(Integer.toString(ModelVariables.seed));
            simulationFieldPanel.add(seedField);
            seedField.setColumns(10);

            JLabel distCentresLabel = new JLabel("Number of distribution centres");
            distCentresLabel.setToolTipText("The number of distribution centres in the simulation. Must be a whole number.");
            distCentresLabel.setHorizontalAlignment(SwingConstants.CENTER);
            simulationFieldPanel.add(distCentresLabel);
```

```
        distCentresField = new JTextField(Integer.toString(ModelVariables.numDistCentres));
        simulationFieldPanel.add(distCentresField);
        distCentresField.setColumns(10);

        JLabel warehousesLabel = new JLabel("Number of warehouses");
        warehousesLabel.setToolTipText("The number of warehouses in the simulation. Must be a whole number.");
        warehousesLabel.setHorizontalAlignment(SwingConstants.CENTER);
        simulationFieldPanel.add(warehousesLabel);

        warehousesField = new JTextField(Integer.toString(ModelVariables.numWarehouses));
        simulationFieldPanel.add(warehousesField);
        warehousesField.setColumns(10);

        JLabel customersLabel = new JLabel("Number of customers");
        customersLabel.setToolTipText("The number of customers in  the simulation. Must be a whole number.");
        customersLabel.setHorizontalAlignment(SwingConstants.CENTER);
        simulationFieldPanel.add(customersLabel);

        customersField = new JTextField(Integer.toString(ModelVariables.numCustomers));
        customersField.setHorizontalAlignment(SwingConstants.LEFT);
        simulationFieldPanel.add(customersField);
        customersField.setColumns(10);

        JLabel extraRoadsLabel = new JLabel("Extra roads");
        extraRoadsLabel.setToolTipText("The network is automatically created to ensure that there is a route from any node
to any other node. Extra roads adds more roads between locations. Must be a value between 0.0 and 1.0. 1.0 means that every
location has a road connecting it to every other location.");
        extraRoadsLabel.setHorizontalAlignment(SwingConstants.CENTER);
        simulationFieldPanel.add(extraRoadsLabel);

        extraRoadsField = new JTextField(Double.toString(ModelVariables.roadProbability));
        simulationFieldPanel.add(extraRoadsField);
        extraRoadsField.setColumns(10);

        JLabel minRoadTravLabel = new JLabel("Minimum road travel time (hours)");
        minRoadTravLabel.setToolTipText("Roads are created with random traversal times. This is the minimum traversal time
for a road.");
        minRoadTravLabel.setHorizontalAlignment(SwingConstants.CENTER);
        simulationFieldPanel.add(minRoadTravLabel);

        minRoadTravField = new JTextField(Double.toString(ModelVariables.minRoadTravelTime));
        simulationFieldPanel.add(minRoadTravField);
        minRoadTravField.setColumns(10);
```

```
            JLabel maxRoadTravLabel = new JLabel("Maximum road travel time (hours)");
            maxRoadTravLabel.setToolTipText("Roads are created with random traversal times. This is the maximum traversal time
for a road.");
            maxRoadTravLabel.setHorizontalAlignment(SwingConstants.CENTER);
            simulationFieldPanel.add(maxRoadTravLabel);

            maxRoadTravField = new JTextField(Double.toString(ModelVariables.maxRoadTravelTime));
            simulationFieldPanel.add(maxRoadTravField);
            maxRoadTravField.setColumns(10);

            JLabel simulationLabel = new JLabel("Simulation");
            simulationLabel.setHorizontalAlignment(SwingConstants.CENTER);
            simulationPanel.add(simulationLabel, BorderLayout.NORTH);

            JMenuBar menuBar = new JMenuBar();
            frmSimulator.setJMenuBar(menuBar);

            JMenu mnFile = new JMenu("File");
            menuBar.add(mnFile);


            JMenuItem mntmQuit = new JMenuItem("Quit");
            mntmQuit.addActionListener(new ActionListener() {
                  public void actionPerformed(ActionEvent arg0) {
                        System.exit(0);
                  }
            });
            mnFile.add(mntmQuit);

            JMenu mnNetwork = new JMenu("Network");
            menuBar.add(mnNetwork);

            mntmViewNetwork = new JMenuItem("View Network");
            mntmViewNetwork.setEnabled(false);
            mntmViewNetwork.addActionListener(new ActionListener() {
                  public void actionPerformed(ActionEvent arg0) {
                        Controller.getController().showNetwork();
                  }
            });
            mnNetwork.add(mntmViewNetwork);
      }


      private void parseValues() {
```

```java
        ModelVariables.days = Integer.parseInt(numberOfDaysField.getText().trim());
        ModelVariables.seed = Integer.parseInt(seedField.getText().trim());
        ModelVariables.numDistCentres = Integer.parseInt(distCentresField.getText().trim());
        ModelVariables.numWarehouses = Integer.parseInt(warehousesField.getText().trim());
        ModelVariables.numCustomers = Integer.parseInt(customersField.getText().trim());
        ModelVariables.numberOfTrucks = Integer.parseInt(numberOfTrucksField.getText().trim());
        ModelVariables.roadProbability = Double.parseDouble(extraRoadsField.getText().trim());
        ModelVariables.minRoadTravelTime = Double.parseDouble(minRoadTravField.getText().trim());
        ModelVariables.maxRoadTravelTime = Double.parseDouble(maxRoadTravField.getText().trim());
        ModelVariables.daysAheadToPlan = Integer.parseInt(horizonField.getText().trim());
        ModelVariables.batchProcessingDays = Integer.parseInt(batchProcessingField.getText().trim());
        ModelVariables.minTruckVolumeCapacity = Double.parseDouble(minTruckVolumeField.getText().trim());
        ModelVariables.maxTruckVolumeCapacity = Double.parseDouble(maxTruckVolumeField.getText().trim());
        ModelVariables.minTruckWeightCapacity = Double.parseDouble(minTruckWeightField.getText().trim());
        ModelVariables.maxTruckWeightCapacity = Double.parseDouble(maxTruckWeightField.getText().trim());
        ModelVariables.minTruckShiftLimit = Double.parseDouble(minTruckShiftField.getText().trim());
        ModelVariables.maxTruckShiftLimit = Double.parseDouble(maxTruckShiftField.getText().trim());
        ModelVariables.numberOfBatchesAtInitialisation = Integer.parseInt(initialBatchesField.getText().trim());
        ModelVariables.numberOfBatchesPerDay = Integer.parseInt(batchesPerDayField.getText().trim());
        ModelVariables.minBatchQuantity = Integer.parseInt(minBatchQuantField.getText().trim());
        ModelVariables.maxBatchQuantity = Integer.parseInt(maxBatchQuantField.getText().trim());
        ModelVariables.minBatchWeight = Double.parseDouble(minBatchWeightField.getText().trim());
        ModelVariables.maxBatchWeight = Double.parseDouble(maxBatchWeightField.getText().trim());
        ModelVariables.minBatchVolume = Double.parseDouble(minBatchVolumeField.getText().trim());
        ModelVariables.maxBatchVolume = Double.parseDouble(maxBatchVolumeField.getText().trim());

    }

    public static void createFInishedFrame() {
        if (runningFrame != null) {
        finishedFrame = new JFrame("Finished");
        finishedFrame.setPreferredSize(new Dimension(150, 85));
        finishedFrame.setBounds(250, 250, 671, 235);
        finishedFrame.getContentPane().setLayout(new BorderLayout());
        JLabel runningLabel = new JLabel("Finished!");
        runningLabel.setHorizontalAlignment(SwingConstants.CENTER);
        JButton okButton = new JButton("OK");
        okButton.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent arg0) {
                finishedFrame.setVisible(false);
            }
        });
        finishedFrame.getContentPane().add(runningLabel,BorderLayout.CENTER);
        finishedFrame.getContentPane().add(okButton,BorderLayout.SOUTH);
```

```
			finishedFrame.pack();
			runningFrame.setVisible(false);
			finishedFrame.setVisible(true);
			}
	}

	public String getResultsDirectory() {
		return resultsDirectory;
	}
}
```

D-20: Random Generator

```
package random;
import java.util.Random;


/**
 * Provides a centralised random object to ensure repeatability of results.
 *
 */
public class RandomGenerator {
	private static Random random=null;
	private static int seed=6000;

	public static void initialiseWithSeed(int s)
	{
		seed = s;
		random = new Random(seed);
	}

	public static Random getRandom()
	{
		if(random==null){
			random = new Random(seed);
		}
		return random;
	}

	public static void incrementSeed() {
		seed++;
	}
```

```
	public static int getSeed() {
		return seed;
	}
}
```