UNIVERSITY OF GAZIANTEP
GRADUATE SCHOOL OF
NATURAL & APPLIED SCIENCES

MODELING AND SOLVING TRUCK-LOAD CONSOLIDATION PROBLEMS BY USING MULTI-AGENT TECHNOLOGY

Ph.D THESIS
IN
INDUSTRIAL ENGINEERING

BY
VAHİT KAPLANOĞLU
JANUARY 2011

Modeling and Solving Truck-Load Consolidation Problems by Using Multi-Agent Technology

PhD Thesis
in
Industrial Engineering
University of Gaziantep

Supervisor
Prof. Dr. Adil BAYKASOĞLU

by

Vahit KAPLANOĞLU

January 2011

T.C.
UNIVERSITY OF GAZİANTEP
GRADUATE SCHOOL OF
NATURAL & APPLIED SCIENCES
DEPARTMENT OF INDUSTRIAL ENGINEERING

Name of the thesis: Modeling and solving truck-load consolidation problems by using multi-agent technology
Name of the student: Vahit Kaplanoğlu
Exam date: 19.01.2011

Approval of the Graduate School of Natural and Applied Sciences

Prof. Dr. Ramazan KOÇ
Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of Doctor of Philosophy.

Prof. Dr. Türkay DERELİ
Head of Department

This is to certify that we have read this thesis and that in our consensus opinion it is fully adequate, in scope and quality, as a thesis for the degree of Doctor of Philosophy.

Prof. Dr. Adil BAYKASOĞLU
Supervisor

Examining Committee Members	signature
Prof. Dr. Hadi GÖKÇEN	
Prof. Dr. Rızvan EROL	
Prof. Dr. L. Canan DÜLGER	
Prof. Dr. Türkay DERELİ	
Prof. Dr. Adil BAYKASOĞLU	

ABSTRACT

MODELING AND SOLVING TRUCK-LOAD CONSOLIDATION PROBLEMS BY USING MULTI-AGENT TECHNOLOGY

KAPLANOĞLU, Vahit
Ph. D. in Industrial Eng.
Supervisor: Prof. Dr. Adil BAYKASOĞLU
January 2011, 173 pages

Logistics organizations mainly the ones providing land transportation services are facing with difficulties while making effective operational decisions. This is especially the case in making load/capacity/route planning and load consolidation decisions where customer orders are generally unpredictable and subject to sudden changes. Classical modeling and decision support systems are mostly insufficient to provide satisfactory solutions in a reasonable time for solving such dynamic problems. Agent-based approaches especially multi-agent paradigms which can be considered as relatively new members of system science and software engineering are providing effective mechanisms for modeling dynamic systems. These systems are generally operating under unpredictable environments and having high degree of complex interactions. It seems that multi-agent paradigms have a big potential to handle complex problems in land transportation logistics. Based on this motivation, in this thesis a multi-agent based load consolidation decision making approach is proposed. In the proposed approach the load consolidation decisions for the less-than-truckload orders which are dynamically dispatched to the system are made by the software agents. The less-than-truckload orders are assigned/consolidated to the trucks by the negotiation mechanism constructed within the model. The proposed approach considers the truck travel distances, number of order rejections, average costs of the transportation operations and system profitability so as to measure the performance of the proposed system. The proposed system is tested with some static problem sets and via running it under some stochastic environment.

Key Words: Third party logistics, logistics costs, load consolidation, agent based load consolidation

ÖZET

YÜK BİRLEŞTİRME PROBLEMLERİNİN ETMEN TEKNOLOJİSİ KULLANILARAK MODELLENMESİ VE ÇÖZÜLMESİ

KAPLANOĞLU, Vahit
Doktora Tezi, End. Müh. Bölümü
Tez Yöneticisi: Prof. Dr. Adil BAYKASOĞLU
Ocak 2011, 173 sayfa

Özellikle karayolu taşımacılığı yapan lojistik organizasyonları etkili operasyon kararları alabilmek için bazı problemlerle karşılaşmaktadırlar. Bu problemlerle daha çok müşteri siparişlerinin belirsiz olduğu ve bu siparişlerin anlık değişikliklere maruz kaldığı durumlardaki yük/kapasite/rota planlama ve yük birleştirme kararlarının alınması sırasında karşılaşılır. Klasik modelleme ve karar destek sistemleri genellikle bu tarz dinamik problemleri tatmin edici ve kabul edilebilir bir zaman içerisinde çözme konusunda yetersiz kalmaktadırlar. Sistem bilimi ve yazılım mühendisliğinde oldukça yeni sayılan etmen-tabanlı yaklaşımlar, özellikle çoklu-etmen modelleri bu tarz dinamik problemlerin modellenmesinde etkili bir yaklaşım sunmaktadır. Bu sistemler genellikle belirsiz bir sistem çevresinde çalışırlar ve bu sistemler problem çözümlerinde çok karmaşık etkileşimler içerirler. Çoklu-etmen modellerinin karayolu taşımacılığı alanındaki karmaşık problemlerin çözümünde etkili bir yöntem olduğu düşünülmektedir. Bu motivasyonla birlikte bu tez çalışmasında çoklu-etmen tabanlı bir yük birleştirme karar verme yaklaşımı önerilmiştir. Bu yaklaşımla birlikte bir konteynır kapasitesinden daha küçük ve dinamik olarak sevkiyatlarının yapılması gereken siparişlerin yük birleştirme kararları sistem içerisinde tanımlanan yazılım etmenleri tarafından yapılmaktadır. Bir konteynır kapasitesinden daha küçük olan siparişlerin araçlara atanmasında veya araçlarda birleştirilmesinde, önerilen model içerisine yerleştirilmiş olan görüşme (müzakere) mekanizması kullanılmıştır. Önerilen yaklaşım araçların boş konteynırlarla kat ettikleri yol miktarını, sistem tarafından reddedilen yük miktarlarını, taşıma operasyonlarının ortalama maliyetlerini ve sistem karlılığını performans göstergesi olarak kabul etmektedir. Önerilen sistem bazı statik test problemleri ile test edilmiştir. Önerilen model aynı zamanda belirsizliğin bulunduğu bir sistem çevresinde benzetim yapılarak test edilmiştir.

Anahtar Kelimeler: Üçüncü parti lojistik, lojistik maliyetleri, yük birleştirme, etmen-tabanlı yük birleştirme

ACKNOWLEDGMENTS

It is my pleasure to thank those who made this Ph.D. thesis possible. First and foremost, a very special note of thanks and appreciation is to my supervisor Prof. Dr. Adil BAYKASOĞLU who has provided me support and encouragement throughout both on my studies of M.Sc. and Ph.D. His ideas about my thesis and studies have always guided me during my studies on these tasks. He always helped me to tackle the complexities associated with distributed artificial intelligence and solving truck-load consolidation problems with multi-agent approach.

I am heartily thankful to committee member of my Ph.D. thesis and head of the department of Industrial Engineering of University of Gaziantep, Prof. Dr. Türkay DERELİ, for his thoughtful guidance, remarkable insights, continuous encouragement and suggestions during both on my M.Sc. and Ph.D. studies.

I also would like to thank to committee member of my Ph.D thesis, Prof. Dr. Rizvan EROL for his helpful guidance and valuable suggestions.

In researching the multi-agent based load consolidation approach, I have contacted with many people from the logistics sector. I am especially thankful to the personnel of EYÜP LOJİSTİK A.Ş. for their help, tolerance and collaboration during my studies. I extend my special appreciation to Eyüp BARTIK who is the owner of the company and Mahmut KARAHAN who is the general manager of the company for their valuable experience sharing about land transportation logistics.

I thank to members of Agent Research Group which was held by Prof. Dr. Adil BAYKASOĞLU. My special thanks go to Dr. Cenk ŞAHİN from Çukurova University who is firstly my friend and then my teammate. He is the people that we collaboratively worked on multi-agent systems. I wish we would work in further projects together.

I would like to express my thanks and best feelings to my friend and colleague Alptekin DURMUŞOĞLU for his encouragement, morale support and our friendship that I believe it will continue forever.

Also, I should thank to Prof. Dr. Lin PADGHAM for inviting me to take the chance of working with the RMIT University Computer Science and IT Department Agent Research Group in Australia. I also thank to members of Agent Research Group at RMIT University especially to Dhirendra SINGH, Zhiyong ZHANG and Simon DUFF for their collaboration on some aspects of my thesis study.

I also thank to my friend and colleague Koray ALTUN for supporting me during procedural processes of my fellowship program application.

I am grateful to TÜBİTAK BİDEB for funding my visiting to RMIT University Agent Research Group and for their Ph.D. scholarship program that supported me throughout my Ph.D. studies.

My special thanks are for my family… My father, mother, brother and sister… My sincere appreciation is to my children Mehmet Furkan and Beyza for tolerating a busy father that was generally away the home.

Finally, I save the best for last, to my wife Sümeyra… Thank you for always believing in me and supporting me when I really needed it.

CONTENTS **page**

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS/ABREVIATIONS

3PL:	Third-party logistics
ABC:	Activity based costing
ACL:	Agent communication language
ACPV:	Average cost per volume
ACPW:	Average cost per weigh
APPV:	Average profit per volume
APPW:	Average profit per weight
AUML:	Agent unified modeling language
b_i:	Time of beginning of service at node i
B:	Set of bids
BDI:	Belief-desire-and intention
C_k	Content of truck k
cfp:	Call for proposal
d_i:	Order i delivery location
$D(.,.)$:	Distance between two points
e_i:	Earliest service time of order i
E:	Edge set
EOQ:	Economic order quantity (EOQ)
ESW:	Economic shipment weight
FIPA:	Foundation for intelligent physical agents
G:	Complete undirected graph
GUI:	Graphical user interface
H_0 :	Null hypothesis
IDE:	Integrated development environment
JAM:	A BDI-Theoretic mobile agent architecture
k_i:	Truck agent i
KQML:	Knowledge query and manipulation language
l_i:	Latest service time of order i
L_i :	Transportation duration of order i
LTL:	Less than truckload
MABLCS:	Multi-agent based load consolidation system
MAS:	Multi agent systems
o_i:	Order i pickup location
O:	Number of order agents released to the simulation system
OA:	Order agent
ORC:	Number of order agents rejected due to cancellation
ORF:	Number of order agents rejected due to feasibility
ORP:	Number of order agents rejected due to profitability
OT:	Number of order agents transported
P:	Profit
PDM:	Prometheus design methodology
PRS:	Procedural reasoning system

T:	Number of truck agents in the simulation system
t:	Time
T_i^{ADV}:	Time between the arrival and earliest pickup time of any particular order agent
T_i^{SLK}:	Slack time between the earliest possible delivery time and latest delivery time of any particular order agent
T_i^{RES}:	Time interval in which the order agents wait for an assignment/consolidation decision.
t_{ij}:	Travel time between node *i* and *j*
TCA:	Traditional cost accounting
TCO:	Total cost of operation
TD:	Total distance
TDV:	Total distance x volume
TDW:	Total distance x weight
TL:	Truckload
TR:	Total revenue
TVT:	Total volume transported
TWT :	Total weight transported
UML:	Unified modeling language
v_i :	Order agent *i* volume
V:	Vertex set
VRPTW:	Vehicle routing problem with time windows
w_i :	Order agent *i* weight
θ:	Transportation tariff
λ:	Inter arrival times
μ:	Threshold value that is used in the truck load consolidation reasoning
μ_D :	Mean difference
τ_i^{ARV}:	Order agent *i* arrival time
τ_i^{AVL}:	Earliest pickup time of order agent *i*
τ_i^{DLN}:	Latest delivery time of order agent *i*

CHAPTER 1

INTRODUCTION

Transportation is an important domain of human activity. It supports and makes possible most other social and economic activities and exchanges (Crainic, 2003). Around the world, regions, both geographic and economic, have different strengths and weakness (Chen et al., 2005). It necessitates the transportation of goods which are produced in different regions. Due to the increased partnership between the firms, transportation has become more and more important than ever.

Worldwide interdependence of trade and flow of goods is constantly growing, therefore logistics and the planning of freight transports are of crucial relevance both for economical and ecological reasons (Fischer et al., 1996). In this age of international competition, improving information systems has forced logistics companies to use new business management techniques (Baykasoğlu and Kaplanoğlu, 2006). In order to reduce the cost of transportation, manufacturers, logistics agencies, and shippers are all in contact with each other.

As the freight rates are negotiable among the manufacturers and carriers, there are numerous different types of freight rates for the truckloads (TL) and less-than-truckloads (LTL) shipments (Jackson, 1981). As a result of the high number of freight rates, the number of transportation service types provided by the logistics companies is increased. However as the number of transportation service types are increased, the complexity of the management of the logistics operations has become complex as well. Although abundance of transportation types reduces the transportation costs (provides competitiveness and some new managerial opportunities), the business complexity of transportation companies increases proportionally. Under such a complex business environment, making effective operational decisions for the logistics companies requires new system approaches different to previous decision making mechanism. This is especially important when making the **load/capacity/route planning** decisions.

On the other hand, the information technology which is available today provides some new opportunities to handle the problems aroused from this domain. There are some new technologies which support the decision makers in their complex operational decisions in logistics such as satellite monitoring systems, geographic information systems, mobile communication systems, navigation systems and etc. There are also some novel system approaches for the management of the orders arriving to logistics systems (Lau and Lau, 2008).

It is a complex problem to coordinate a set of services from a large number of service resources under various constraints and uncertainties. Existing approaches to this problem have relied on complete information regarding service requirements and resources, without adequately addressing the dynamics and uncertainties of the environments. The real-world situations are complicated as a result of *ambiguity* in the requirements of the services, the *uncertainty* of solutions from service providers, and the *interdependencies* among the services to be composed (Wang et al., 2009).

One of the most promising operational decisions within land transportation domain is the *consolidation of the orders* according to their spatial attributes and time windows to appropriate trucks and routes. This is because the cost-based system performance of third-party logistics (3PL) companies is directly affected by the technique used to bring the loads together within a container.

Freight or *load consolidation* is a transportation option that combines different items, produced and used at different locations and different times, into single vehicle loads in order to minimize transportation costs and maximize vehicle utilization. When more than one items are put to same container this means that they are consolidated into a single vehicle unit (Hall, 1987). The rationality may be that LTL shipments require an unnecessarily large number of carriers which may duplicate each other’s delivery routes to customers (Min and Cooper, 1990). The more items that are put inside the container, the lower the transportation cost per order and per unit. Actually, the practical usage of load consolidation in logistics is very common especially for the 3PL companies. Although, load consolidation technique has been used practically for hundreds of years, the interdependence of the organizations and the volatile business environments make the problem domain very hard to model and solve (Tyan et al., 2003).

Load consolidation problems require the simultaneous solving of pickup and delivery problems as well. In the presence of time constraints the problem of finding a feasible pickup and delivery plan for a fleet is *NP*-Hard (Savelsbergh, 1995). While satisfying the time constraints of the orders (finding a feasible pickup and delivery plan), load consolidation decisions require the determination of time or quantity to wait to dispatch the vehicles. The amount of time to wait or amount of capacity utilization of the containers determines the dispatch time after a feasible (optimality is not sought) pickup and delivery plan is obtained. Some orders scheduled for a specific vehicle might have some slack time to be picked-up or there might be some unused space of the containers to wait for some future potential orders. This is because the orders have the earliest pickup, latest pickup, earliest delivery and latest delivery time attributes. The pickup and delivery time of the orders might be suited within a feasible horizon in this time-line. However, there is a trade-off between to wait a potential order or start to transport from the vehicle's view. The decision of dispatch time of the vehicles might have a considerable effect on the total transportation performance and it is one of the critical decisions made by the dispatch officers within any 3PL organization.

The orders which are to be transported have a specific time of pickup and delivery, a specific destination, physical attributes such as volume and weight and etc. The 3PL organizations should make a decision about whether the order in consideration can be transported or not. Then, if it can be transported what will the cost of transportation be? However, the dispatch officers who make these decisions in practical applications of 3PL organizations have to take into consideration all the dynamics of the system. The previous accepted orders, the positions of the resources, the physical capacities of the containers, governmental regulations, updated road information, and driver schedules are some examples of data that shape the decision of the planners. In addition to all the limitations of the on-hand data, the dispatch officers should also think the sustainability of the system by considering future positions of the resources. If there is an increasing trend of transport request in some specific region, the dispatch officers should take it into consideration to be cost effective and competitive. Within this context, the load consolidation decisions should be made by considering all the system dynamics. Therefore there should be multiple load consolidation plans that the dispatch officers should be able to select the proper one among all. The selection of the proper load consolidation method depends on the foresight of the dispatch officers. Therefore, using a single load consolidation method during all the load acceptance/rejection decisions and load planning process is not appropriate from the point of view of competitiveness and cost effectiveness.

In this thesis study, a multi-agent based general load consolidation approach is proposed in which the orders and system resources are presented as software agents where the load consolidation decisions are made collaboratively by the system itself. The performance of the model is presented with some experiments and its applicability to real load consolidation systems is discussed.

In the proposed multi-agent based load consolidation model the agents are designed within a belief-desire-and intention (BDI) concept. The empty travel distances of the trucks, number of order rejections, average costs of the transportation operations and system profitability are the final evaluation parameter of the proposed agent-based load consolidation system.

The problem solved within the agent-based load consolidation framework inherits the real operations details of continuous order-arrival environment. The design of the proposed system is handled via using Prometheus design methodology (PDM). The truck agents determined in the PDM use their own local reasoning mechanisms to reach the overall-goal of the general system. The orders arrive to agent based 3PL load consolidation system might be rejected according to system resources. Although the environmental inputs (percepts) are modeled to the system, the problem is yet a simplification of real-world load consolidation problem which includes the bin-packing sub-problem as well.

The inspiration behind this study is the dynamism of the load consolidation methodology selection in the practical logistics operations. Some selected experiments are highlighted in order to present the performance of the proposed multi-agent based load consolidation system. The basic distinction of this study is that it involves the consolidation of the LTL orders arriving to 3PL organizations with their own pickup and delivery time windows and their spatial attributes. The decision makers in the proposed multi-agent based load consolidation approach are the proposed software agents which have their own goals, beliefs and reasoning mechanisms. In the further section of this thesis study the details of the multi-agent based load consolidation system will be analyzed. In this thesis study, load consolidation problems are tried to be handled within a multi-agent approach so as to adapt the decisions made in this domain to complex and changing business environment of logistics. Within the context of this study a *multi-agent based load consolidation system* (MABLCS) is proposed.

1.1 Thesis Organization

Chapter 2: This chapter covers the research papers which are about the classical load consolidation approaches and novel approaches proposed for his domain in the literature. The advantages of using such a multi-agent based approach are discussed while stating the previous studies in the literature.

Chapter 3: Multi agent systems (MAS) and their general properties are specified. The multi agent development platforms are mentioned. The inspiration behind the usage of multi-agent paradigms to logistics domain is discussed in this chapter.

Chapter 4: The definition of the load consolidation problem is given in this chapter. The notations that will be used in the further chapters are defined.

Chapter 5: The conceptual design is made by using PDM. The details of the design phases are presented in this chapter. The interactions and interaction protocols between agent types are designed in this chapter.

Chapter 6: The proposed MABLCS is implemented by using a popular agent development language and environment, this chapter presents the implementation phase. The reasoning mechanisms of the agent types in the MABLCS are illustrated.

Chapter 7: The performance of the proposed MABLCS is tested by using a well-known problem named as *vehicle routing problem with time windows* (VRPTW). A set of problem is run on the proposed system.

Chapter 8: The performance of the proposed MABLCS is tested under stochastic environment by simulating some test bed examples.

Chapter 9: The conclusion about the thesis study is given. Some aspects of the study that might be improved as a future research are discussed in this chapter.

CHAPTER 2

LITERATURE REVIEW

In this chapter the literature review of the load consolidation decisions are presented along with the reasoning of using multi-agent technology in load consolidation decision making domain. The classical load consolidation approaches are surveyed in this section.

In today's unpredictable business environments, managers are being asked to cut costs while maintaining or even increasing the service quality in a very short period of time. In order to be competitive, a company should be able to provide low cost, high quality services/products in a short lead time (Baykasoğlu, 2003). As a result, manufacturing firms prefer transportation with minimum cost and time since transportation costs of products have a considerable portion in the total product cost. This can be achieved by consolidating the loads.

As expressed previously, freight or *load consolidation* is a transportation option that combines different items, produced and used at different locations and different times, into single vehicle loads in order to minimize transportation costs and maximize vehicle utilization. When more than one item are put to same container this means that they are consolidated into a single vehicle unit (Hall, 1987). Although, load consolidation technique has been used practically for hundreds of years, the interdependence of the organizations and the volatile business environments make the problem domain very hard to model and solve (Tyan et al., 2003). It has become very important to transport the produced goods in an economic way for the companies. Consolidation strategy answers the question of what is the most efficient way to put together the inventory, vehicle and terminal pieces (Hall, 1987).

According to Brennan (Brennan, 1981), there are three types of consolidation. They are (1) spatial consolidation, (2) product consolidation and (3) temporal consolidation. Hall (1987) classifies the consolidation methods into three types which are inventory consolidation, vehicle consolidation and terminal consolidation. Inventory consolidation, which is a temporal form of consolidation, involves storing items that are produced and used at different times, and transporting them in the same load (Hall, 1987). Vehicle consolidation involves picking-up and dropping-off items at different origins and destinations. Terminal consolidation brings items from different origins to a single location where they are sorted, loaded onto new vehicles, and taken to different directions. Hall (1987) also, described the importance of consolidation in formulating transportation strategy, explored the basic components of consolidation, and then described consolidation trade-offs. Then he concluded that properly planned, rational, and coordinated consolidation strategy can greatly reduce transportation costs without sacrificing service quality. Hall (1987) added that, the key synthesizing such a strategy is a clear understanding of the consolidation options and consolidation trade-offs. Sheffi (1986) illustrated six different forms of consolidation: (1) consolidation in vehicle units; (2) in containers; (3) in channels; (4) in networking; (5) in time; and (6) in tours (Sheffi, 1986). Tyan et al. (2003), considered freight consolidation problem that arose in the context of a global 3PL distribution alliance (Tyan et al., 2003). Three consolidation strategies were developed in their study to minimize the total cost under the constraints of capacity and service requirements. The problem was formulated as a mathematical programming model. Results of the study indicated that a collaborative consolidation strategy could benefit both the carrier and the shipper at the same time. Chen et al. (2005) presented two models for joint stock replenishment and temporal shipment consolidation decisions and compared their relative cost effectiveness. Two models differed mainly in the shipment-release scheme — the time-based or the quantity-based. Numerical examples showed that the quantity-based scheme always performs at least as well as the time-based scheme (Chen et al., 2005). Çetinkaya et al. (2006) investigated the impact of alternative outbound dispatch strategies on integrated stock replenishment and transportation decisions. Their results demonstrated that significant cost savings can be achieved by using the suggested quantity-based and hybrid strategies rather than the exact solutions of time-based dispatch models. Numerical results indicated that quantity-based dispatch strategies are always superior to time based strategies, and hybrid-quantity-based solutions are very promising. Hybrid-time-based solutions may also lead to substantial savings (Çetinkaya et al., 2006). According to Attanasio et al. (2007) consolidation can be achieved in three ways. Firstly, small shipments that have to be transported over long distances may be consolidated so as to transport large shipments over long distances and small shipments over short distances (facility consolidation). Secondly, LTL pick-up and deliveries associated with different locations may be served by the same vehicle on a multi-stop route (multi-stop consolidation). Thirdly, shipment schedules may be adjusted forward or backward so as to make a single large shipment rather than several small ones (temporal consolidation) (Attanasio et al., 2007). They also examined a consolidation and dispatching problem motivated by a multinational chemical company which has to decide routinely the best way of delivering a set of orders to its customers over a multi-day planning horizon. They developed a heuristic based on cutting plane framework. Their computational results showed that their procedure allowed achieving more savings. More detailed literature review about classical load consolidation methods can be found in the papers of (Crainic, 2003; Crainic and Laporte, 1997).

Within the 3PL operations context, shippers call the 3PL companies in order to request to carry a load from a specific origin to a specific destination within a specified time-window. The dispatch officers in 3PL companies must decide quickly whether to accept a request to move the load. However the decision of acceptance/rejection requires all the system dynamics. This is because before the acceptance/rejection decision of an order request, the cost of the operation must be calculated. The cost calculation process requires all the asset information within the system. Actually knowing all the asset data within the system does not result in a cost estimation of transportation. The dispatch officers should evaluate the transportation alternatives or optimize a complex mathematical model.

In 3PL operations when the vehicles finish their operations and they become idle, the positioning of the vehicles is also being considered. This is because the dispatch officers should position the vehicles where the potential orders might arrive. In other words the dispatch officers must decide what to do with the vehicle which completes its operation. It may be assigned another load, repositioned to another region in anticipation of future loads, or held in anticipation of future loads in the destination region.

The information of the loads is known gradually as the system operates, this is the most crucial point in load consolidation decision making. Because the conventional load consolidation techniques fixes the load consolidation limit and decides the number of orders brought together within a container or route.

Demand for the transportation orders is highly stochastic; some of the orders arrive to 3PL organizations just one or two hours before the transportation. As previously stated, load consolidation is a transportation option that combines different items, produced and used at different locations and different times; into single vehicle loads in order to minimize transportation costs and maximizes vehicle utilization. Load consolidation is a systematic attempt for reducing total transportation cost between a given origin and destination.

Managers must give a lot of decisions in planning a load consolidation such as:

- Which customer orders will be consolidated and which shipped individually?
- When will customer orders be released? What event(s) will trigger the dispatch of a consolidated vehicle load?
- Where will the consolidation be done? Should consolidation take place at the factory, on a vehicle, at a warehouse or terminal, etc.?
- Who will consolidate? Should consolidation be performed by the manufacturer, shipper, customer, carrier, or a third party?
- How will consolidation be carried out? Which specific consolidation techniques will be used? (Higginson and Bookbinder, 1994).

Figure 1.1 summarizes the load consolidation decision problem when a transportation order arrives to a 3PL organization.

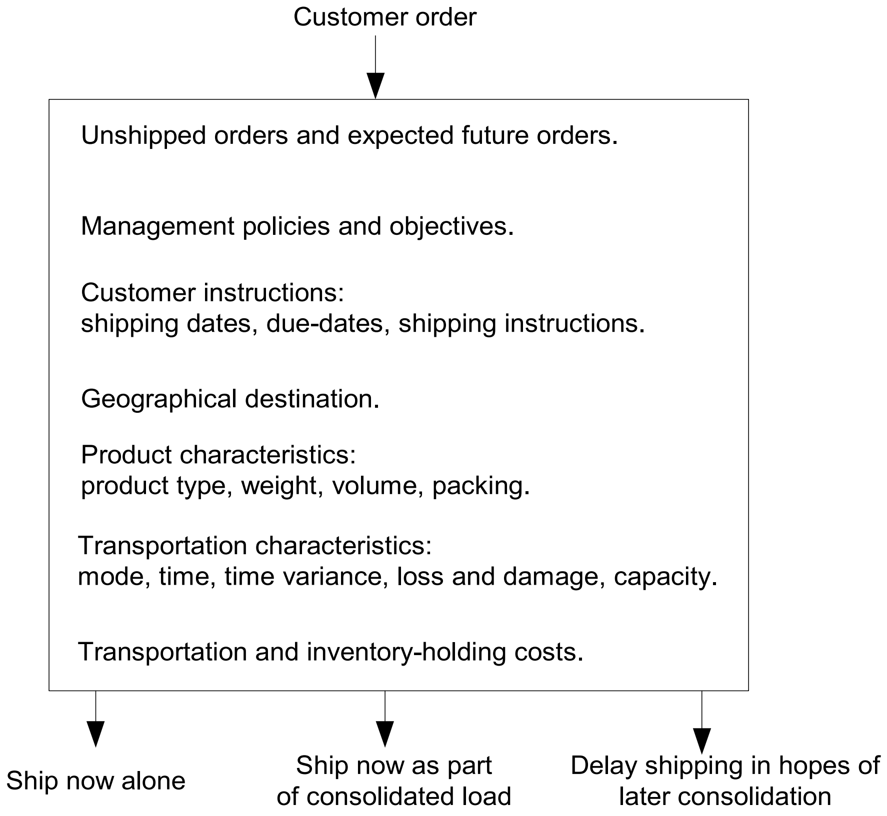


Figure 1. 1. Decision problem for a newly arrived customer order (Higginson and Bookbinder, 1994)

There are mainly three commonly used load consolidation policies in the literature which are; time, quantity and time-quantity consolidation policies. In time based load consolidation policy, the orders are dispatched at a prearranged shipping time. The amount of orders consolidated is not considered if the shipping time is reached. In a quantity-based load consolidation policy, the orders from a single center to a single destination are seized and dispatched when a consolidation level is reached. Thirdly, in time-quantity based consolidation policy, the orders are accumulated until whether the consolidation time or consolidation level is reached. In his study, Jackson (1981) states that time policy is frequently used by the transportation practitioners (36 % of respondents). But the differences in between the three policies was not large (Jackson, 1981).

In their studies, Higginson and Bookbinder (1994) have tested these three shipment release policies via simulation of the policies. The simulation results show that, there is no policy yields the lowest cost per weight under all system parameters. However, a time-and-quantity policy consistently yielded the smallest mean delay per order. The simulation results show that, the choice and performance of a shipment release strategy is greatly dependent on the order arrival rate and the length of time customers are willing to wait for shipment.

Holding time for the transportation order has a critical effect when the time based load consolidation policy is considered. The amount of time to hold the orders has a significant influence on logistics performance.

Time required for holding the orders up to the point where the orders are shipped are also analyzed by some authors. In the studies of Masters (1980), it is stated that longer holding times reduce transportation costs and increase mean delivery time (Masters, 1980). Jackson (1981) also noted that the combination of long holding times and large system volumes (high order arrival rates) results in lower costs.

There are some mathematical models that optimize the shipment quantity when the order arrival rate is known. In quantity based consolidation, if the order arrival rate is known, economic shipment weight (ESW) model is used to calculate the weight that should be accumulated in order to make shipment economical. ESW has the same role with the economic order quantity (EOQ) in inventory models (Higginson and Bookbinder, 1994). Equation 1 represents the ESW model.

$$ESW = \sqrt{{}^{2\,\hat{a}\,\mathrm{F_L E[w]}}\!\big/_{r_w}} \tag{1.1}$$

Where â is the order arrival rate, F_L is the sum of all fixed costs associated with a vehicle load, E[W] is the expected weight per customer order, and r_w is the variable cost of carrying inventory per unit weight per time period. However in order to use this model the arrival rate of the orders must be known.

The previous studies of load consolidation assume that the loads are originated from a single source and they are shipped to a single destination. However in the real business case of load consolidation, orders which have different routes might be assigned to the common container when their pickup and delivery time constraints fits to the pickup and delivery plan of the vehicle. In real 3PL operations, the dispatch officers might change the pickup and delivery plan of the vehicles up to the last minute of the operation because of the dynamism of the arriving orders. The operations decisions might also change due to the order cancellations, variability of the travel times, road congestion, weather conditions and etc.

ESW models are used in order to consolidate the loads to a point (to a capacity or time level) before dispatching under the assumption of all the loads have the same route. However, in practice orders which have different routes might be consolidated within same container. Therefore, the problem in practice is much more complex than the ones handled by ESW models.

We could interpret the previous studies on load consolidation subject as; selection of a dominating shipment policy which depends on the system. Holding time of the transportation orders, capacity levels of the organizations and desired customer service level are basic system attributes which has the primitives for the selection of a dominating shipment policy.

However, the models given for the load consolidation problems assume that the loads are originated from a single source and they are shipped to a single destination. However in real road transportation business, transportation orders which have some different routes might be consolidated within same container.

The analytic models given in such studies try to find the best shipment-consolidation policy under some conditions. However, real life consolidation / dispatching operations do not fit exactly to the models given in such studies. This is because the dynamic nature of the real business environment. For example the dispatcher officer might change the content of the consolidated batch during the process of consolidation because of some changes on order attributes or on resource conditions. Dispatcher officer might shift one load from one batch to another for the sake of route optimization and cost reduction after a change in order attribute. Thus new technologies are required to keep up with the complexity and the dynamics of the domain (Fischer et al., 1996).

Load consolidation decisions inherently comprise the problem of order-truck-scheduling. There exist efficient algorithms for solving static scheduling problems; however, classical computer science, operations research (OR), and classical centralized artificial intelligence (AI) have so far failed to provide adequate methodologies and algorithms to cope with open dynamic scheduling problems. (Fischer et al., 1996). Unfortunately, virtually any problem of practical interest falls into the latter category like the complex load consolidation problems in 3PL organizations. Calculating the optimum of a complex scheduling problem is a non-trivial task. It is not possible for such problems to calculate the best solution in a straightforward manner. Therefore all possible solutions have to be calculated, to be able to then choose the best solution (Karageorgos et al., 2003). Within this context, as the 3PL organizations are becoming integrated with other enterprises to meet the needs of the global economy, the complexity of the businesses they do will increase. Amid the complex inter-organizational relations, the performance of a logistics network is dependent on the effective coordination and collaboration of the partners (Wong and Fang, 2008). In such a business structure, agent-based technology which is a novel system approach might be considered suitable because agents can dynamically adapt their behavior to changing requirements and they can reduce the number of planning and scheduling alternatives via negotiation (Karageorgos et al., 2003).

There are numerous studies in the literature where agent-based solution is proposed for some distributed, adaptive and dynamic problem domains (Cil and Mala, 2010). However, application of multi-agent technology in supply chain management and logistics, especially the coordination of the transportation orders, has become a new and emerging research area (Lu and Wang, 2007). Multi-agent application in supply chain and logistics management mainly covers building multi-agent architecture for the logistics network and provides cooperation mechanism between agents (Lu and Wang, 2007; Wong and Fang, 2008). Load consolidation in 3PL organizations as a problem domain can be classified within this dynamic and complex structure. The transportation services are planned in order to coordinate resources (vehicle fleet, containers, vehicle drivers and etc.) under some constraints and uncertainties.

Load consolidation decision is a systematic method in order to reduce total transportation cost between a given origin and destination. However, in practical logistical operations (ones in the 3PL companies) LTL orders which have different routes from different customers might be consolidated within a container. When the vehicles visits the sub-destinations and then reaches to the final destinations the dispatch officers tries to organize new orders to make the trucks utilized on the way of home. This problem structure somehow has common properties with vehicle routing problems.

A typical vehicle routing problem can be described as the problem of designing least cost routes from one depot to a set of geographically scattered points (cities, stores, warehouses, schools, customers etc). The routes must be designed in such a way that each point is visited only once by exactly one vehicle, all routes start and end at the depot, and the total demands of all points on one particular route must not exceed the capacity of the vehicle. The vehicle routing problem with time windows is a generalization of the standard vehicle routing problem involving the added complexity that every customer should be served within a given time window (Bräysy, 2001).

In practice the load consolidation problem also considers the vehicle routing problem with pickup and delivery and vehicle routing problem with time windows (VRPTW). However, the pick-up and delivery points and time windows of all the orders are unknown for a static point of time. In practice, the load consolidation decisions are conducted by considering all the available order attributes and system parameters simultaneously.

Below in figure 2.1 , shipment framework of a typical 3PL company is presented (Baykasoğlu et al., 2007). The model supports the management by giving a step-by-step shipment procedure. As depicted in figure 2.1, demands are accepted after their evaluations and then they are consolidated in step 3. The proposed model continues until the transportation of the loads or their rejection. Step 3, 6 and 7 are the some difficult steps of the proposed model. This is mainly due to the selection of the consolidation strategy. Load consolidation method has a considerable impact on the logistics costs. Therefore, the consolidation operations and tools used to make such decisions are very important. Step 6 of figure 2.1 requires detailed system information, because within a dynamic transportation environment obtaining the capacity information requires a well-established information technology infrastructure. Step 7 of figure 2.1 also requires a consolidation *sub-model* in order to offer a cost to the customer. In summary, the whole shipment framework necessitates a well established load consolidation method. As presented in the figure, it may be necessary to turn down requests for service if time window or regional or system wide capacity constraints cannot be met.

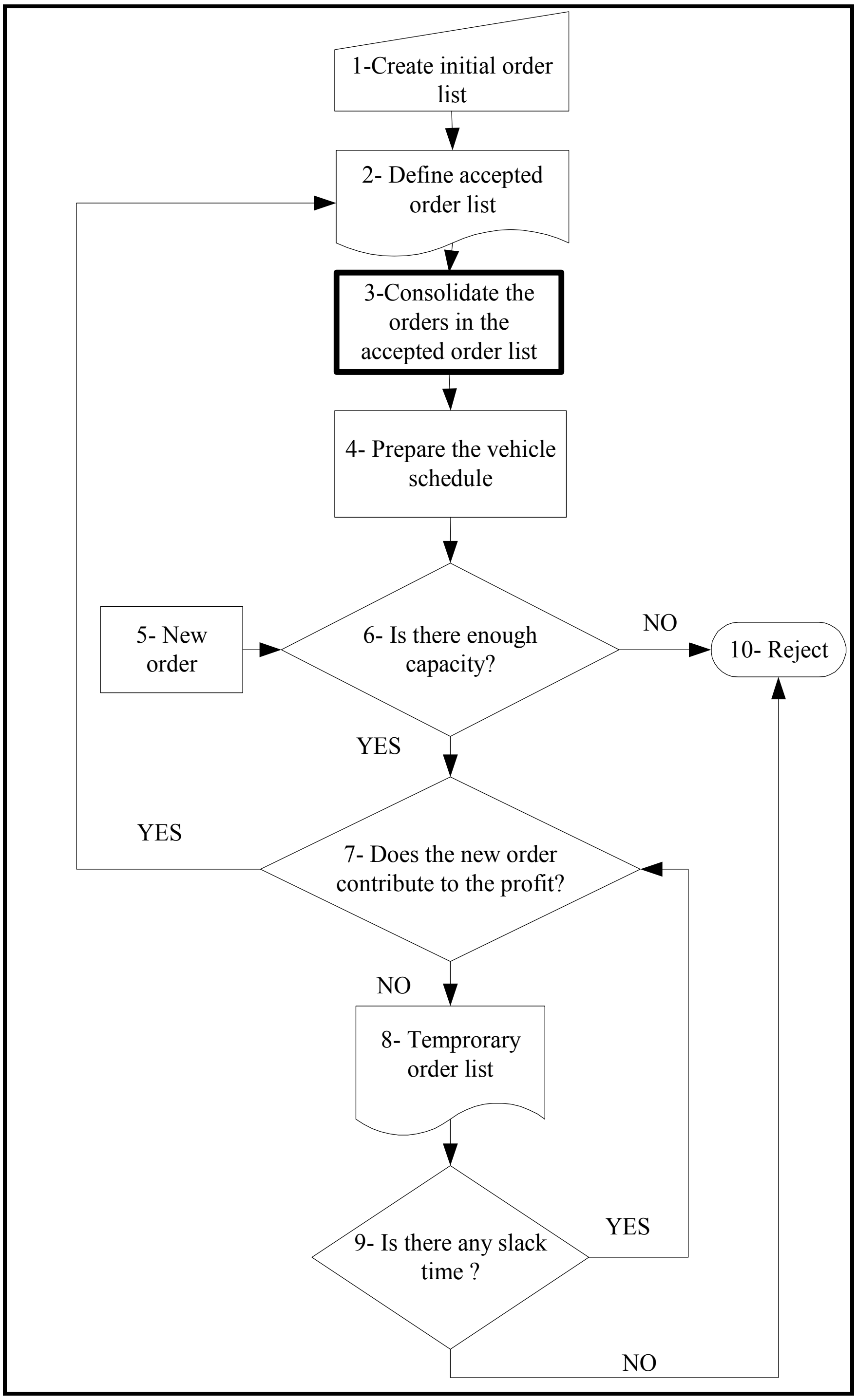


Figure 2. 1. Shipment framework for 3PL companies (Baykasoğlu et al., 2007)

CHAPTER 3

MULTI-AGENT SYSTEMS FOR LOAD CONSOLIDATION DECISIONS

In this chapter, the agent based systems are introduced by defining the software agents. Functionalities of agents in some dynamic systems are discussed. The usage of the multi-agent systems in load consolidation decisions is also introduced in this chapter.

3.1 Multi-Agent Systems

As is expected from a fairly young area of research, there is not yet an universal consensus on the definition of an agent (Padgham and Winikoff, 2004). However the most widely accepted definition of the agent is of Wooldridge and Jennings (1995). According to their definition, an agent is a computer system that is situated in some environment, and that is capable of autonomous action in this environment in order to meet its design objectives (Wooldridge and Jennings, 1995). It is designed to achieve its defined goals and desires. A software agent is a component that can exhibit reasoning behavior under both proactive (goal directed) and reactive (event-driven) stimuli. Agents also have the attributes of; mobile, truthful, benevolent and rational. Agents are also able to learn (Wooldridge, 2002).

The agent which wants to solve a problem broadcasts a call for bids, waits for a reply for some length of time, and then awards a contract to the best offer(s) according to its selection criteria. When an agent is instantiated, it will wait until it is given a goal to achieve or experiences an event that it must respond to (Hahn et al., 2009). Agents are social, because they cooperate with humans or other agent types in order to achieve their tasks. Agents are reactive, because they perceive their environment and respond in a timely fashion to changes that occur in their environment. In addition to that, agents are proactive, because they do not simply act in response to its environment but are able to exhibit goal-directed behavior by taking initiative (Bellifemine et al., 2007).

There are some basic properties of software agents that specify them and distinguish them from classical software.

- The first property, being *autonomous*, means that agents are independent and make their own decisions. They are at least to some extent capable of autonomous action – of deciding for themselves what they need to do in order to satisfy their design objectives. This is one of the properties that distinguish agents from objects of object oriented programming. When we take a system into account which consists a number of agents, then a consequence of the agents being autonomous is that the system tends to be decentralized (Padgham and Winikoff, 2004).

- The second property of agents which is *situatedness* that does not constrain the notion of an agent very much since virtually all software can be considered to be situated in an environment. However, where agents differ is the type of environments. Agents tend to be used where the environment is challenging; more specifically, typical agent environments are *dynamic, unpredictable and unreliable*.

- The third property of the agents is being reactive. Agents are often situated in dynamic environments that change rapidly. In particular, this means that an agent must respond to significant changes in its environment. In other words, agents need to be *reactive*, responding in a timely manner to changes in their environment (Padgham and Winikoff, 2004).

- Fourth property of the agents is that they pursue goals over time, that is, they are ***proactive.*** One property of goals is that they are **persistent**; this is useful in that it makes agents more robust: an agent will continue to attempt to achieve a goal despite failed attempts. Agents must be able to recover from failures of actions, that is they must be ***robust.*** A natural way of achieving robustness is to be ***flexible***. By having a range of ways of achieving a given goal, the agent has alternatives that can be used when a plan fail.

- Finally, agents always need to interact with other agents, that is, agents are ***social.*** They are capable of interacting with other agents- not simply by exchanging data, but by engaging in analogues of the kind of social activity that we all engage in every day of our lives: cooperation, coordination, negotiation, and the like (Wooldridge, 2002).

In addition to that, the term agent is widely used to describe a range of software components, varying in capability from procedural wizards, found in popular desktop applications, to information agents that are used to automate information search and retrieval, and, finally, to intelligent agents capable of reasoning in a well-defined way (AOS, 2009).

When more than one software agents are suited together collaboratively or competitively, these systems are called as *multi-agent* systems. Multi-agent systems are generally used where the problem domains are particularly complex to handle. The single agents within multi-agent environment use the most suitable method of its own particular problem. Even if an agent system can be based on a solitary agent working within an environment and if necessary interacting with its users, usually they consist of multiple agents (Bellifemine et al., 2007). Multi-agent systems are systems composed of multiple interacting *agents.*

As the agent technology is a newly emerging research area the specification of the agent technology is newly constructed. For example, Parunak (1999) lists the following characteristics for an ideal application of agent technology:

- Modular, in the sense that each entity has a well-defined set of state variables that is distinct from those of its environment and that the interface to the environment can be clearly identified.

• Decentralized, in the sense that the application can be decomposed into stand-alone software processes capable of performing useful tasks without continuous direction from some other software process.
• Changeable, in the sense that the structure of the application may change quickly and frequently.
• Ill-structured, in the sense that all information about the application is not available when the system is being designed.
• Complex, in the sense that the system exhibits a large number of different behaviors which may interact in sophisticated ways (Parunak, 1999).

The Foundation for Intelligent Physical Agents (FIPA, http://www.fipt.org) is a multi-disciplinary group pursuing software standards for heterogeneous and interacting agents and agent-based systems. This organization has made available a series of specifications to direct the development of multi-agent systems (Lin and Lin, 2006). Currently the most used and studied agent communication language is the FIPA Agent Communication Language (ACL), which incorporates many aspects of KQML (Knowledge query and manipulation language) (Bellifemine et al., 2007). The primary features of FIPA ACL are the possibility of using different content languages and the management of conversations through predefined interaction protocols. Coordination is a process in which agents engage to help ensure that a community of individual agents acts in a coherent manner (Nwana et al., 1997).

Agent oriented programming is highly suited to many application areas, including distributed business systems, command and control, intelligent appliances and simulation. Although still young and under development, it has already shown particular promise in a variety of distributed problem solving tasks such as fleet organization, air traffic management and air combat simulation. Because it offers such a modular and elegant solution to many of the problems faced in reactive processing, agent-oriented programming is ideally suited to these environments (AOS, 2009).

Other specific characteristics of multi-agent systems are that; (1) each agent has incomplete information or capabilities for solving the problem and thus, has a limited viewpoint; (2) there is no system global control; (3) data are decentralized; and (4) computation is asynchronous (Sycara, 1998). Agents are defined with their attributes and functionalities. For example, Ferber (1999) describes agents in two main approaches: cognitive agents systems and reactive agent systems (Ferber, 1999).

3.2 Multi Agent Systems for Load Consolidation

In real logistics business environment, some expert knowledge and experiences are required in order to make the load consolidation decisions; this is also the inspiration behind the multi-agent paradigm. From the viewpoint of agent definition, 3PL operations resources and entities are all candidate to be a software agent type within the multi-agent system. This is because the system resources and entities within a 3PL can act autonomously during their operations in order to reach their goals (such as delivering the loads with the possible least cost). The trucks operate for any 3PL company and the services provided are naturally distributed and the dispatch officers are to organize the vehicle resources within a dynamic business environment. Therefore the system resources represented as agent within this problem domain fits to the agent definition. As a result of distributed representation of the resources and entities of the system there is no need for a central load consolidation unit. Thus one very complex central load consolidation decision is replaced with several (depends on the number of trucks, drivers and orders) smaller consolidation auctions allowing for quick reactions without global re-planning or re-consolidation.

As the multi-agent systems are preferred for the systems where the complex information systems are in consideration (Cohen and Stathis, 2001), load transportation domain could be modeled and solved within this structure.

The agents which represent the resources used within the 3PL operations have the properties of a regular agent. This is because the business environment of the system resources within this domain is dynamic, unpredictable and unreliable. The system resources within this domain must react proactively within such business environment in order to reach to goals of the system and agent itself. The system resources within a 3PL operation have the property of proactiveness. This is because the system resources within a 3PL company act according to its goals (such as delivering the loads on board etc.). During the transportation process of any truck agent, the goal of timely-delivering of the orders on board is persistent. The agent types within 3PL domain have also the property of flexibility and sociality. This is because the resource elements can change their plans in order to achieve their goals (flexibility) and they can interact with other system resources (trucks, drivers etc.) in a social environment.

There are some studies of multi-agent systems in logistics and supply chain system (Brintrup, 2010; Forget et al., 2009; Gaudreault et al., 2009; Monteiro et al., 2007). However, load consolidation models and methodologies discussed in the literature are inadequate to solve the open-dynamic load consolidation problems. The load organization and load consolidation decisions in the practical logistics domain are open and dynamic problems. The multi-agent based approach enables the system to response some unexpected events without global re-planning and re-consolidation. The global re-planning or re-arrangement of the orders requires numerous modeling and optimization efforts. Figure 3.1 represent how the agent based methods approximates the optimum of a typical model.

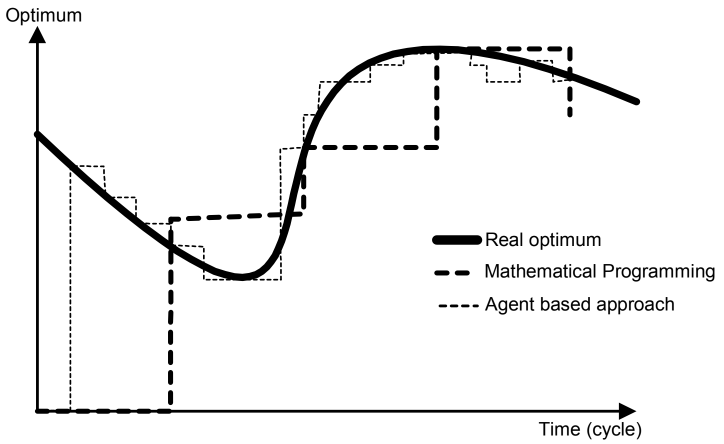


Figure 3. 1. Different modeling approaches in reaching optimum in dynamic situations (WhiteStein)

Because the domain of transportation especially road based ones inherently distributed and complex, multi-agent systems are especially suitable for it (Dastani et al., 2004; Ying and Dayong, 2005). Earlier studies on this subject reveal that agent-based systems seem very suitable for the transportation domain, but that this subject needs to be verified by more deployed system (Davidsson et al., 2005; Fox et al., 2000).

In this thesis the system resources such as trucks maintain their own load consolidation initiatives thus the decision of very complex problem of load consolidation is given by some system resources and some system entities (orders) collaboratively. This construct enables the system to response some unexpected events without global re-planning and re-consolidation.

What is different in this proposed approach is that the 3PL companies themselves do not have to give the load consolidation decisions centrally. Thus one very complex central load consolidation decision is replaced with several (depends on the number of trucks and orders) smaller consolidation auctions allowing for quick reactions without global re-consolidation to unforeseen events, such as attribute change in transportation orders.

CHAPTER 4

PROBLEM DEFINITION OF LOAD CONSOLIDATION DECISION WITH MULTI-AGENT SYSTEM

In this chapter, the load consolidation problem with multi-agent system is defined formally. The problem is expressed by using some notations.

In the dynamic nature of the 3PL operations, at any given instant, each vehicle has an associated status: (a) moving loaded, (b) moving empty, (c) idle and available for assignment, and (d) idle and unavailable. A vehicle changes status at the occurrence of certain events that mark the completion of the corresponding activity. For instance, a vehicle's status changes from "idle and available" to "moving empty" (towards to load origin) on receiving an assignment from the dispatch center (Regan et al., 1998). The dynamism of the business environment affects the attributes of the vehicles operating in the system.

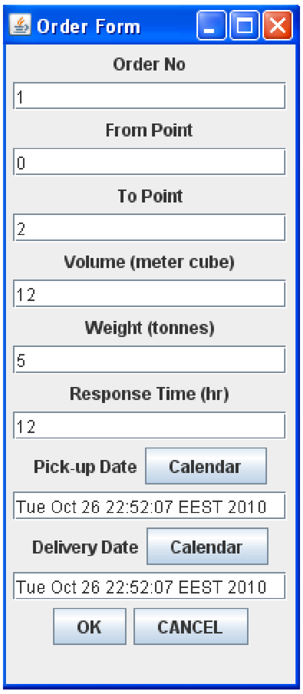
The business dynamism is also valid for the orders that are arriving to the system. The orders might have last minute changes in their attributes. The shippers might cancel the transportation of the order. These are all classified as stochastic events in this problem domain. In most of the studies about dynamic pickup-delivery and vehicle routing problems the traffic jams are considered. However in this proposed multi-agent based approach, traffic congestion is not of primary concern because the transportation operations are on intercity context where the effect of traffic on the transportation times can be neglected.

As stated previously, 3PL companies are facing with difficulties for making effective load consolidation decision. Classical modeling and decision support systems are not adequate for providing on-line solutions for this problem domain. In this thesis, a multi-agent based load consolidation system is proposed for 3PL companies. So as to accomplish the multi-agent based load consolidation decisions the problem is formally defined. In order to define the proposed agent based load consolidation system some notations which are adapted from the notations of Yang et al. (2004) are used.

The problem in this study comprises the decisions of acceptance and rejection of the arriving orders according to feasibility of the system by considering the pickup and delivery deadlines. Systems in which loads have associated pickup and delivery deadlines that must be met usually operate at lower utilization levels (for the same overall demand) than those in which deadlines are either nonexistent or nonbinding, because some service requests must be refused (Regan et al., 1998).

In this study, time evolution is represented with a continuous variable $\mathrm{t} \in [0, \infty)$. The start of the multi-agent based load consolidation system (MABLCS) is assumed as the time zero. At the time of zero of agent-based system all the truck resources are assumed as empty and located at one of the network node.

Order pickup and delivery points and geographical truck positions are assumed within expandable transportation network. The shortest distances between any two points (candidate pickup or delivery points) are denoted with $D(.,.)$. The distances between origin and destination of any order is known and can be obtained by using a third-party software (mapping software), therefore there is no need to define the distances between any points when the system operates on real-time. Transportation orders enter to the system when $\mathrm{t} \geq 0$. The arrival of the orders are represented by an increasing real numbers $(\tau_i^{ARV})_{i \geq 1}$ where τ_i^{ARV} denotes the arrival time of order *i* (orders have an *id* which is unique for each of them). In practical applications, the orders arrive to system as an input through filling such a form represented in the figure 4.1. At the arrival time of any order agent, its attributes are held within a 7-tuple $OA_{\mathrm{i}} \equiv (\mathbf{o}_i, \mathbf{d}_i, \mathbf{v}_i, \mathbf{w}_i, T_i^{ADV}, T_i^{SLK}, T_i^{RES})$. When the form given in figure 4.1 is filled then it means that a new order which has seven attributes is released to the system.



Order No
1
From Point
0
To Point
2
Volume (meter cube)
12
Weight (tonnes)
5
Response Time (hr)
12
Pick-up Date
Tue Oct 26 22:52:07 EEST 2010
Delivery Date
Tue Oct 26 22:52:07 EEST 2010

Figure 4. 1. Sample order arrival form

The arrival time of the orders are known by default with the system time of the arrival epoch (τ_i^{ARV}). The notations used in the 7-tuple order vector are as the following;

$\mathbf{o}_i$ and $\mathbf{d}_i$ are the pickup and delivery locations of the orders.
$\mathbf{v}_i$ and $\mathbf{w}_i$ are the respective volume and weight of the orders
T_i^{ADV} is the time between the arrival and earliest pickup time of any particular order
T_i^{SLK} is the slack time between the earliest possible delivery time and latest delivery time of any particular order
T_i^{RES} is the time interval in which the orders wait for a assignment/consolidation decision.
Figure 4.2 illustrates the time parameters of any particular order where;
τ_i^{ARV} is the request time when the order arrives to the system; τ_i^{AVL} is the earliest pickup time of the order (which is represented as *Pick-up Date* in the form in figure 4.1) and L_i is the distance between T_i^{ADV} and T_i^{SLK} that presents the transportation duration of the order. Therefore the joint sequence $(\tau_i^{ARV}, OA_{i\geq 1})$ characterizes any particular order.

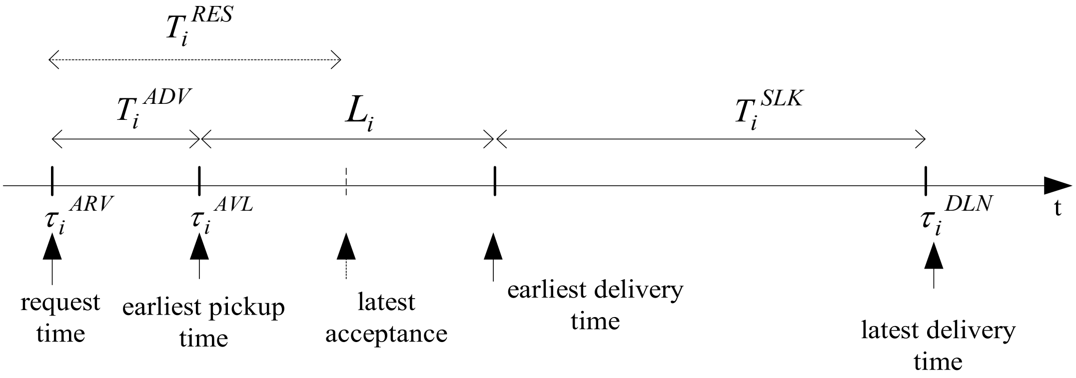


Figure 4. 2. Time parameters or order agents

If an order is accepted by the 3PL provider then it must be transported to its destination within the specified time window. The transportation details such as the exact pickup and delivery times are determined by the system decision makers after the order is accepted. The truck which the order will be assigned or consolidated might change up to the time point of pickup. This is because the business environment of logistics is extremely volatile. In other words the plan of the trucks might change up to the point of latest loading time of orders which is represented as $\tau_i^{DLN} - L_i$, where τ_i^{DLN} denotes the latest delivery time of any particular order (which is represented as *Delivery Date* in the form in figure 4.1).

When a new order arrives to the 3PL provider, the plan which is active might change so as to include the newly arrived order. However if any order arrived to 3PL system is granted to be transported (assigned / consolidated to a truck agent) then any other order arrival does not change the acceptance decision of particular order. The acceptance/rejection decisions of 3PL provider become permanent by time $\tau_i^{ARV} + T_i^{RES}$ and cannot be changed then (it is presented as *latest acceptance* in figure 4.2).

Each truck within the 3PL system is denoted by k, $1 \leq k \leq K$ that is at any time t has either idle, moving empty, or moving loaded attribute.

In this thesis, when an order is scheduled to an empty truck then it is called as *assigned* and when it is scheduled together with any other order then it is called as *consolidated*. Finding the best order assignment/consolidation to available trucks requires much more effort with classical operations research algorithms because the number of parameters of the orders arriving to the system and their dynamic nature. Here in this proposed multi-agent based approach the problem of load consolidation according to their attributes are tried to be found by utilizing the distributed nature of the MAS. By doing so the contribution of such systems to complex truck-load consolidation decisions could be analyzed by evaluating some system performance parameters such as truck utilization and transportation costs.

The system that the multi-agent system will be proposed operates under the third party logistics system. The 3PL company has a fleet of *K* trucks. Contrary to studies of Yang et al. (2004), each truck can carry more than one order at a specific time (t) (Yang et al., 2004). This is the basic principle behind the load consolidation decisions. When a new order arrives to company in consideration via telephone call, e-mail or request form on internet the company is given the order attributes. The order attributes are; the pickup location, the delivery location, the earliest pickup time, the latest delivery time, order volume, order weight and type of the order (fragile, perishable and etc.) and etc. The 3PL company has a respective time for each arriving order according to its property in order to decide whether it will carry the order or not. The customers are informed up-to the time that acceptance/rejection decision ends. The acceptance/rejection decision is made by considering the available truck capacities. The problem considers the empty travel of the truck agents to pickup the next order. If truck agent does not have any order entity within it and travel to pickup the next order agent then an empty travel cost occurs.

The arrival of the orders does not follow a sequence. That is at any time epoch t there might be a new order arriving to the system. When the order arrives to the system the acceptance/rejection decision should be made by considering all of the system dynamics. Figure 4.3 summarizes how the new order changes the operations data and schedule of the trucks. When a new order arrives to the system, the previous operations might be (if feasible) cancelled or delayed by inserting the new order to the schedule of any truck.

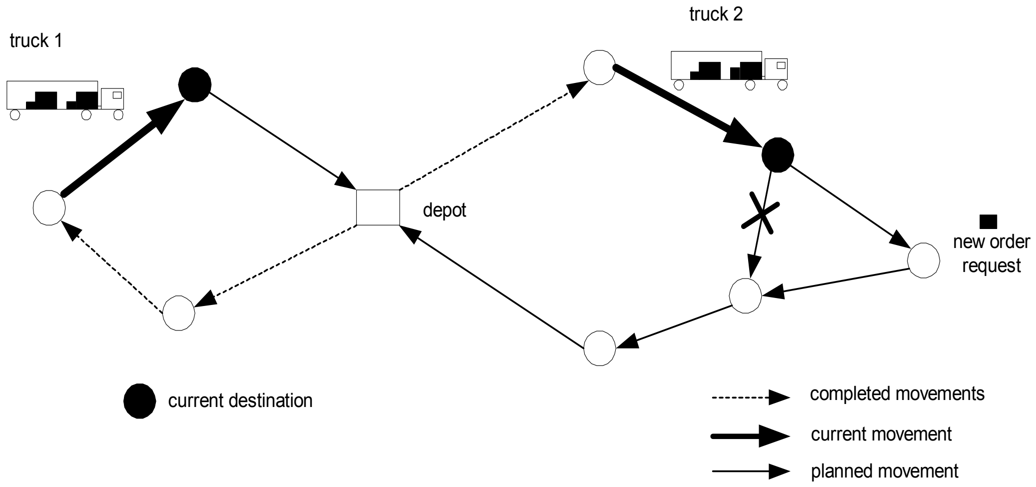


Figure 4. 3. On-line routing of trucks adapted from (Gendreau et al., 1999)

The context for this study is both truckload and less-than-truckload assignment/consolidation operations, in which each assignment/consolidation involves a vehicle moving through the set of pickup and delivery locations. This approach is proposed for the 3PL companies which might have assets, namely a fleet of vehicles and a pool of drivers, to provide service to a set of customer load origins and destinations, distributed over a typically wide geographic region, on an ongoing basis, over time. Ignoring the distinctions of company-owned and owner-operator fleets assumed that the vehicle fleet (and driver pool) is under the operational control of a central authority, referred to as the dispatcher. The system elements are represented as software agent and the operations decisions are made by the agents with a self-emerging manner.

Based on these definitions, in this thesis, the research question is as follows; "***how the multi agent-based approach can be used to solve the truck-load consolidation problems in third party logistics (3PL) organizations and what is the contribution of such systems to complex truck-load consolidation decisions by evaluating some system performance parameters such as truck utilization, transportation cost and load transportation profits***".

CHAPTER 5

DESIGN OF MULTI-AGENT BASED LOAD CONSOLIDATION SYSTEM

In this chapter, the design of the proposed multi-agent based load consolidation system is conducted. The proposed system is designed by utilizing Prometheus design methodology.

In this study, the problem which is defined in chapter 4 is designed with the multi-agent paradigm. As the multi-agent paradigm is a methodology at its infancy, there are some novel approaches to model such systems. There is not a high-level development methodology for the agent systems. In principle, agent architectures can be divided into four main groups: logic based, reactive, BDI and layered architectures (Bellifemine et al., 2007). Multi-agent design and modeling are implemented by some software systems that uses BDI architecture in their modeling construct (e.g. Procedural Reasoning System (PRS) (Georgeff and Lansky, 1987), JAM (A BDI-Theoretic mobile agent architecture) (Huber, 1999) and JACK (Howden et al., 2001)). In most of the agent-based models BDI architectures are generally used (Rao and Georgeff, 1995).

One of the most well-known BDI architectures is the PRS (Georgeff and Lansky, 1987). This architecture is based on four key data structures: beliefs, desires, intentions and plans, and an interpreter (see figure 5.1).

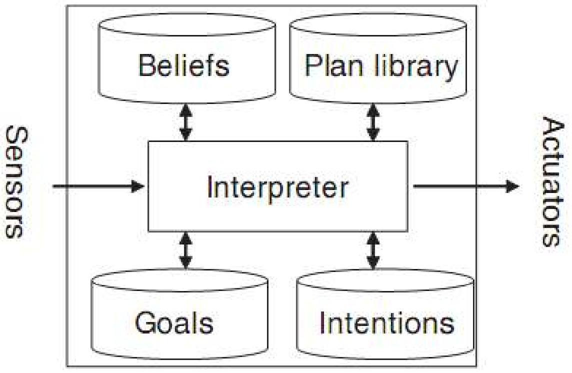


Figure 5. 1.The PRS agent architecture (Bellifemine et al., 2007)

In the PRS system, beliefs represent the information an agent has about its environment, which may be incomplete or incorrect. Desires represent the tasks allocated to the agent and so correspond to the objectives, or goals, it should accomplish. Intentions represent desires that the agent has committed to achieving. Plans specify some courses of action that may be followed by an agent in order to achieve its intentions. These four data structures are managed by the agent interpreter which is responsible for updating beliefs from observations made of the environment, generating new desires (tasks) on the basis of new beliefs, and selecting from the set of currently active desires some subset to act as intentions. Finally, the interpreter must select an action to perform on the basis of the agent's current intentions and procedural knowledge (Bellifemine et al., 2007). The central concepts in the BDI model are (Georgeff and Rao, 1998);

Beliefs: Environment information
Desires/Goals: Objectives
Intentions: The currently chosen course of action
Plans: Means of achieving certain future world states

A plan is a way of realizing a goal; for example, a plan for achieving the goal of *load consolidation* might necessitate the following steps; check whether the load can be consolidated, find the route when the load is assigned to truck, calculate the cost of transportation when the load is consolidated, consolidate the load (if it is relevant), and transport the load to destination.

In this thesis, Prometheus methodology is selected to design the proposed multi-agent based load consolidation system. The methodology is developed for specifying and designing agent-oriented software elements that will build the proposed system. Prometheus is a general purpose methodology for the development of software agent systems, in that it is not tied to any specific model of agency of software platform (Padgham and Winikoff, 2004).

Prometheus as a methodology is for the development of software agent systems. Associated with the methodology, a freely available software design tool is developed (Prometheus Design Tool (PDT)). Prometheus design tool (PDT) is so useful during designing agent based systems by helping to be consistent. Prometheus is also selected in this study as a design methodology because of its harmony with JACK^{TM} Autonomous Software Platform on which this study is implemented. Prometheus methodology complies with the PDT and architecture of JACK Autonomous Software Platform.

The Prometheus methodology comprises three distinct but interconnected stages. Figure 5.2 below, represents the stages of Prometheus agent modeling methodology.

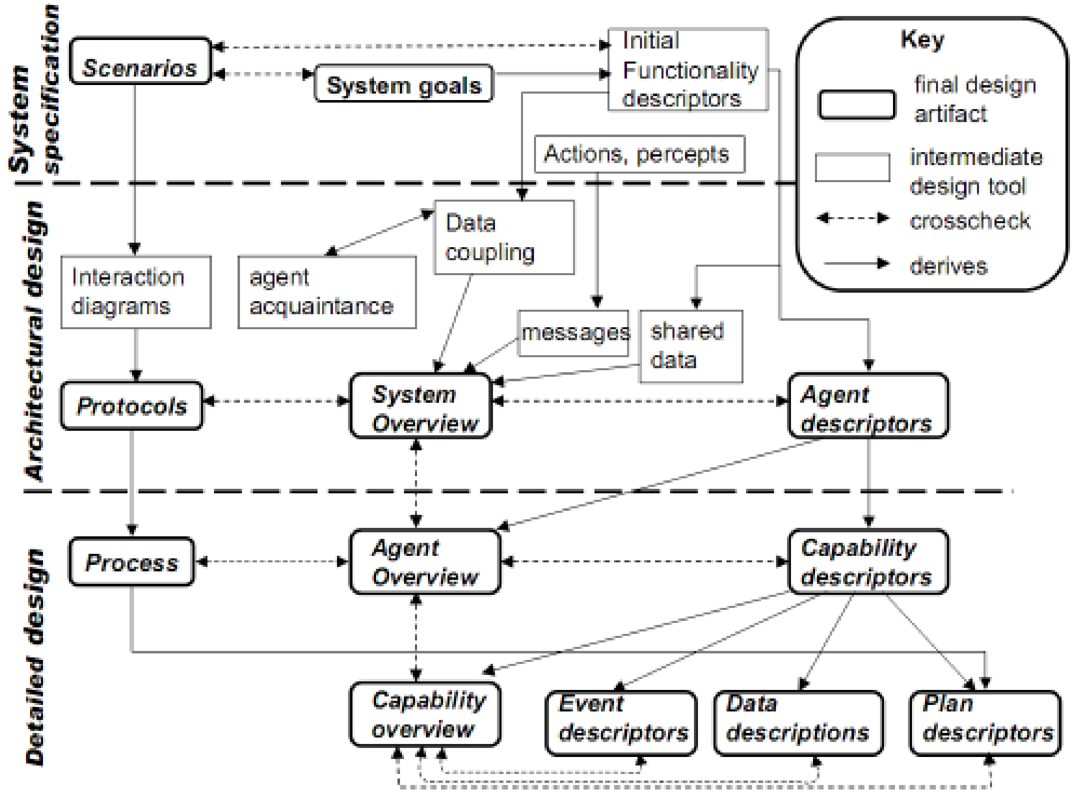


Figure 5. 2.The phases of Prometheus methodology (Padgham and Winikoff, 2004)

As expressed previously, in this thesis, Belief-Desire-Intention (BDI) agent model structure is selected in order to design the multi-agent load consolidation system. It is because the BDI model has a wide application area and it is well documented. Prometheus methodology also complies with the BDI architecture. Before implementing the MABLCS the details of the systems are modeled in Prometheus methodology. This methodology is not related with any agent development systems. The stages of the Prometheus methodology is as follows;

System specification stage: In this stage, the system is specified by focusing on the goals and basic functionalities of the system. System specification stage focuses on the *system goals*, *use case scenarios*, *system functionalities* and *interface* between system and its environment (actions and percepts).

Architectural design stage: In this stage, the agent types are determined according to system specification stage. It is similar to building class diagrams from the requirements documents of UML. The overall system structure is captured by using the *system overview diagram*. Dynamic behavior of the system is described by using *interaction diagrams* and *interaction protocols*. The functionality descriptors within system specification stage are used in order to build agent types. Each agent type consists of one or more functionalities. The grouping of the functionalities according to agent types are made by utilizing *data coupling* and *agent acquaintance* diagrams. The messages between agents and their shared data are also designed within architectural design stage. These elements constitute the *system overview*. System overview therefore combines agents, data, external input and output and states the communication between agents (excluding timing of communication). The timing of the communication between agent types are firstly specified within the system specification. The communication between agents with their timing is designed explicitly in *agent interaction diagrams*. Interaction protocols are used to define all possible interactions between agents.

Detailed design stage: Detailed design stage is divided into two sections. First section includes agent overview diagram and plan descriptors**.** Second part includes process specifications. In first stage, agent types are equipped with the plans for accomplishing their goals. In Prometheus methodology, each agent is equipped with *internal events*, *plans* and detailed *data* structures. Plans are refined in terms of events and data. The dynamics of the agents are refined with the process specification in the detailed design stage.

In this chapter, multi-agent based load consolidation system is designed in a step by step manner that is compatible with Prometheus design methodology.

By completing all the steps of Prometheus design methodology the pathway of the agent development is achieved. In *specification stage* the general system elements are determined. The goals of the agent systems are determined in this step. The scenarios that might happen during the runtime of the agent system are also determined in this step of the methodology. Within *architectural design stage* the types of the agents are determined by evaluating the system specifications. And, the agents are articulated with the respective capabilities and plans in order to achieve their goals. And in the *detailed design* phase all the detailed relations between agent types are determined.

5.1 System Specification

In this stage of the system design, Prometheus design methodology focuses particularly on specification of goals. In addition to that, it requires specification of *functionalities and scenarios*, related to the identified goals. In this stage of modeling, the interfaces with the environment in which agent types are suited are determined. The interfaces are determined in terms of *percepts* that arrive from the environment. Also the actions that impact the environment are determined in this stage of the modeling.

5.1.1 Goal Specification

The proposed MABLCS specified in this study has the following system goals;

MAIN GOAL: Increase utilization level of system resources and decrease total transportation costs

System goals derived from the main goal:

- *Consolidate the transportation orders while satisfying the system constraints (e.g. Pick-up and delivery times, volume and weight capacity of containers etc.) in real time*

- *Supply decision support for the dispatch officers during their decision process*

In Prometheus methodology, system goals are refined by asking the question of "how" for each of the system goals. The answer of this question is the sub-goal of each system goal. Thereby, the refined goal list is obtained.

- ***Consolidate the transportation orders while satisfying the system constraints (e.g. Pick-up and delivery times, volume and weight capacity of containers etc.) in the real time***

  Sub-goals

1. Get the order information from customers in real time

2. Monitor each truck (geographic position, availability condition, capacity level etc.)
3. Select proper truck to assign the order on hand (e.g. Hazard category of dangerous goods might require some special equipment to handle)
4. Check truck routes in order to exchange orders between trucks
5. Get the truck attribute changes from truck monitoring system
6. Get the order attribute changes from the user interfaces
7. Change / exchange transportation orders between trucks when necessary
8. Provide order delivery schedule to the user interfaces (when orders have not a definite pick-up and delivery time)

- ***Supply decision support for the dispatchers during their operations***

  Sub-goals

1. Get the load consolidation decisions from the system
2. Get the real time system data
3. Monitor the trucks during their operation in real time
4. Monitor the orders during their operation in real time
5. Provide user interfaces to make the system users to monitor the system changes
6. Provide user interfaces to make the system users to be aware of the system transportation proposal (when orders have not a definite pick-up and delivery time)

In the further section of goal specification similar goals are coalesced. Below the system goals are coalesced into four different groups.

Order Management

- Get the order information from customers in real time
- Get the order attribute changes from the user interfaces
- Monitor the orders during their operation in the real time

Truck Management

- Monitor each truck (geographic position, availability condition, capacity level etc.)
- Check truck routes in order to exchange orders between trucks
- Get the truck attribute changes from truck monitoring system
- Monitor the trucks during their operation in the real time

Load Consolidation

- Select proper truck to assign the order on hand (e.g. Hazard category of dangerous goods might require some special equipment to handle)
- Change / exchange transportation orders between trucks when necessary
- Get the real time system data

General System Management

- Provide order delivery schedule to the user interfaces (when orders have not a definite pick-up and delivery time)
- Get the load consolidation decisions from the system
- Provide user interfaces to make the system users to monitor the system changes
- Provide user interfaces to make the system users to be aware of the system transportation proposal (when orders have not a definite pick-up and delivery time)

5.1.2 Functionalities

Functionality is the term which is used for a stack of behavior, which consists of a grouping of related goals, as well as percepts, actions and data relevant to the behavior (Padgham and Winikoff, 2004). The actions and triggering events, information used and information produced are the elements stated within the functionalities descriptor of system specification stage. Functionality descriptor is stated for each cluster of system goals as following;

Order Management Functionality: The goals of order management group have the "getting the order data from customer via internet or telephone call", "getting the order attribute changes from the user interfaces via internet or telephone call" and "monitoring the orders during their operation in the real time" actions. The system is triggered by a new order arrival or an attribute change of an order. The data of the newly arrived order or the data of order attribute change is used. After the order is assigned (consolidated) to a proper container, the data of containers (for example capacity level in terms of volume and weight and route information) is obtained after the activities of this functionality.

Truck Management Functionality: The goals of truck management group have the "monitoring trucks", "checking truck routes in order to exchange orders between trucks", "getting the truck attribute changes from truck monitoring system" and "monitoring the trucks during their operation in the real time" actions. The system is triggered by a truck attribute change. The data of geographical position of the truck is used. When the transportation operations are completed the operation information is generated.

Load Consolidation Management Functionality: The goals of load consolidation management are consolidation the orders while "changing / exchanging transportation orders between trucks" and "getting the real time system data" actions. The system is triggered by the truck management and order management functionality actions. The data of geographical position of the trucks, the data of capacity utilizations of the trucks and order attribute changes are used. The system is also triggered by the change in order attribute. The orders might be re-consolidated with other trucks after they are assigned to a particular truck. In this functionality, the operation data are generated.

General System Management Functionality: The goals of general system management group have the "providing order delivery schedule", "getting the load consolidation decisions from the system", "providing user interfaces" actions. The actions of order and truck management functionality and load consolidation management functionality trigger the actions of general system management functionality. The data produced from the load consolidation management functionality is used as an input within the functionality. This functionality checks whether any order has a pre-determined order pick-up and delivery time. If there is a pre-determined pick-up and delivery time for an order then this functionality check whether the data provided by load consolidation management functionality fits to time window. If not, *order transportation proposal* is generated as output data in this functionality.

5.1.3 Scenario Development

Scenarios are complementary to goals in that they show the sequences of steps that take place within the system. In developing goals, some attributes of the scenarios are already being built up. Scenarios are primarily used to illustrate the running state of the system. As scenarios are developed, it becomes evident where there is a need for information from the environment and where actions are required. The scenarios in agent based load consolidation system shows the sequence of operations that is required when any system change occurs. The scenarios of the proposed MABLCS are stated as follows;

Order arrival to the system: When the system is triggered by the acquisition of a new order. This scenario encapsulates the order management and load consolidation management functionalities.

Truck attribute change: Change in geographical position of any truck and breakdowns of trucks. This scenario encapsulates truck management functionality.

Order attribute change after plan is prepared: Inconsistency of the orders arrived to the system. This scenario encapsulates load consolidation management functionality.

Selecting proper trucks from the available trucks within the system: This scenario occurs during the real time order assignment. This scenario encapsulates load consolidation management functionality.

Change / exchange transportation orders between trucks: The scenario which occur when the system requires order change/exchange. This scenario encapsulates load consolidation management functionality.

Order move (transportation): Transporting the orders. This scenario encapsulates truck management functionality.

Delay in transportation (because of Traffic jam etc.): This scenario occurs when some unexpected events occur in the MABLCS that result in delay of transportation. This scenario encapsulates truck management, order management and load consolidation management functionalities.

5.1.4 Interface Description

Interface description is also complementary to scenario development in that they appear during the occurrence of scenarios. The proposed MABLCS is situated in a changing and dynamic environment. In this part of the *system specification stage*, the input and output elements about the environment while the system is running are modeled by naming them as *percepts* and *actions* respectively. At this stage, the data used and produced are also modeled. The percepts, actions and data within this proposed agent model is as following;

Percepts:

- New order arrival information
- Truck network position change information
- Loading / unloading information
- Order attributes change
- Traffic jam
- Truck breakdowns
- Driver breaks
- Delays in customs
- Search for new consolidation alternatives

Actions:

- Truck change of orders after re-assignment or re-consolidation
- Delivery time change
- Order route change
- Re-assignment
- Re-consolidation
- Truck route change
- Container elements change

Data:

Transportation Management System Database: It is database system where all the operations information is stored for retrieval when necessary. The database used within this system is the database of the management system of 3PL logistics operations. All the standard operations are recorded within this database.

5.2 Architectural Design: Specifying the Agent Types and Interactions

Firstly, the agent types interacting with each other within agent systems are determined in this stage of Prometheus methodology. And then the interaction protocols between these agent types are determined.

Each agent within the agent system is derived by utilizing the functionality information. The functionalities which have the common properties are grouped so as to build agent types. Figure 5.3 depicts the data coupling diagram for load consolidation decisions made within 3PL domain. Three clusters are obtained by grouping the four different functionalities. Order management functionality with its relational persistent database of orders and orders data object built up the order agent type. Truck management functionality with its respective persistent database of trucks and truck data object build up the truck agent type. General system management functionality and Load consolidation functionality holds for regional load consolidation agent type with the business rule database and respective order and truck data objects. Therefore there are three distinct agent types in the system which are; *Order Agent*, *Truck Agent* and *Regional Load Consolidation Agent*.

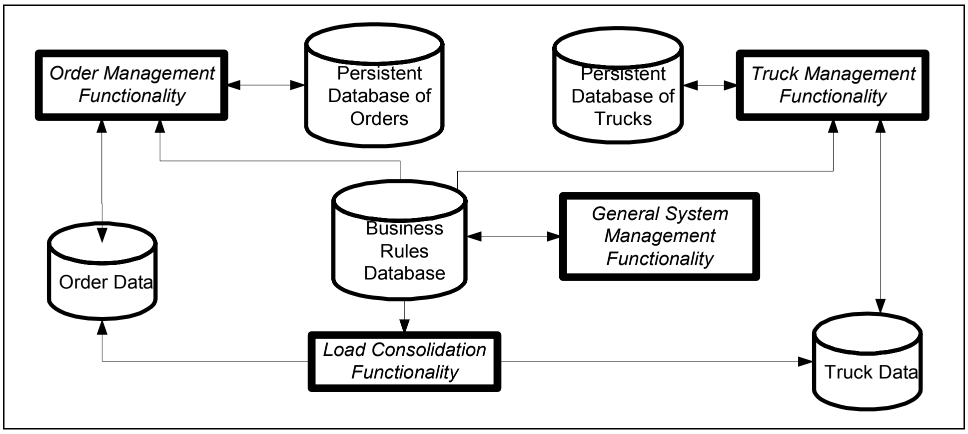


Figure 5. 3. Data coupling diagram for load consolidation decisions in 3PL

Agent Descriptors: There are a number of questions that needs to be resolved during the agent specification stage in order to describe each agent according to data coupling diagram, these are;

- How many agents of each type will there be?
- What is the lifetime of each agent?
- Agent initialization - what needs to be done?
- Agent demise – what clean up needs to be done?
- What are the goals of the agent types?
- What percepts will this agent react to?
- What actions (if any) will it take?
- What data does the agent use or produce?

Agent descriptions are obtained with the following tables.

Table 5. 1. Order agent descriptors

Name: Order agent

Description: Gets the order data from customer, assign itself to a proper truck agent

Lifetime: Initialized when a new customer order reaches to the system. Demise when the order is rejected from the system or it is transported

Initialization: Obtains the order request form

Demise: Closes all the data connections when it leaves the system

Functionalities included: Order management functionality

Uses data: Order data, business rules database, persistent database of orders

Produces data: Order data, persistent database of orders

Goals: Assign itself to a proper (with least cost) truck container. Goal detail of order agent depends on the customer who requests it to be transported (in connection with the order attribute)

Percepts responded to: Arrival of a new order to the system, change in order attribute

Actions: Change in the persistent database of orders

Protocols and interactions: Truck finding with truck agents, change/exchange with load consolidation agent

Table 5. 2. Truck agent descriptors

Name: Truck agent

Description: Transport the orders while satisfying the due date requirements of order agents

Lifetime: Initialized when a new truck resource is added to the fleet of 3PL system. Demised when the truck resource leaves the system

Initialization: Obtains the truck attribute form to be initialized

Demise: Closes all the data connections when it leaves the system

Functionalities included: Truck management functionality

Uses data: Truck data, business rules database, persistent database of truck

Produces data: Truck data, persistent database of trucks

Goals: Transporting the accepted transportation orders (with least cost) within the time window of orders.

Percepts responded to: Order proposal, change in truck attribute

Actions: Change in the persistent database of trucks, change in geographical position, and change in capacity level

Protocols and interactions: Order assignment with order agents, change/exchange with regional load consolidation agent

Table 5. 3. Regional load consolidation agent descriptors

Name: Regional load consolidation agent

Description: Consolidate the orders while satisfying system constraints

Lifetime: Initialized when the system starts to operate. Demise when the system exits.

Initialization: Obtains the truck and order data from their respective data sources

Demise: Closes all the data connections when it leaves the system

Functionalities included: Order and truck management functionality

Uses data: Order data, truck data, and business rules database

Produces data: Operational data is produced

Goals: Consolidate the orders and maintain general system management while satisfying system requirements.

Percepts responded to: Change in order and truck attribute

Actions: Update in the truck route, update in order delivery time

Protocols and interactions: Route updating with truck agents, delivery time updating with order agents

Order agents: Order agents start to live in the MABLCS after their creation with the 7-tuple attributes given in chapter 4 (problem definition). They enter to the system via internet, telephone call or with direct interview with the customers. The order agents live in the system whether for the T_i^{RES} amount of time or until the end of their transportation (completion time) according to their acceptance by a particular truck agent. Therefore the life-time of the order agents depends on the state condition of the transportation network and truck agents. When an order agent enters to the system it is first directed to its respective regional load consolidation agent. In order to be assigned / consolidated to a particular truck agent they first negotiate with a relational regional load consolidation agent to find suitable truck agents to negotiate. The assignment/consolidation of the order does not mean for the order agent to reach to final. The order agents don't have the authority of negotiating with truck agent directly. The goal of the order agents are to be assigned/consolidated to a truck.

Truck agents: The truck agents are created within the system with the 5-tuple attribute vector which are; *name*, *volume capacity*, *weight capacity*, *network position* and *status*. At any instant, each truck agent has an associated status: (1) moving loaded, (2) moving empty, (3) idle and available for assignment, and (4) idle and unavailable. The status of the truck agent change according to their operations. For example, a truck agent's status might change from "moving loaded" to "moving empty" when it deliver an order to its destination and goes for the pick-up of another one with empty container. The truck agents live in the system up to the time point of their exclusion from the system by the system manager. Otherwise they live in the system. Basic operations of the truck agents are loading, transporting, unloading and communication with order agents and other truck agents. Truck agents have the data object (beliefset) which holds the sequence of consolidated order agents. The goal of the truck agents is to consolidate as much as orders into their containers while satisfying the constraints of the assigned or consolidated orders and container capacity limitation. In order to achieve these goal truck agents uses BDI reasoning mechanism to be adaptive to the changing environment.

Regional load consolidation agent (Regional mediator agent): When an order agent enters to MABLCS with its 7-tuple attributes it directly communicates with the regional load consolidation agent to find the truck agents to negotiate. The order agents have not the authority to directly negotiate with the truck agents. This enables the system computation to be relaxed. The regional load consolidation agent is responsible for making the order and truck agents to negotiate while increasing the load consolidation occurrences. Therefore, regional load consolidation agent has the goal of consolidating the orders to the respective truck agents. Regional load consolidation agent continues to operate when there is not a new order agent entering to the system. It tries to find some alternative load consolidation options in order to improve the consolidation level. In order to achieve its goal it might hold negotiations between order and truck agents even after the assignment/consolidation of any particular order agent. This is because, the newly assignment of the orders might create new load consolidation opportunities.

The defined agent types are presented within the MABLCS environment in figure 5.4.

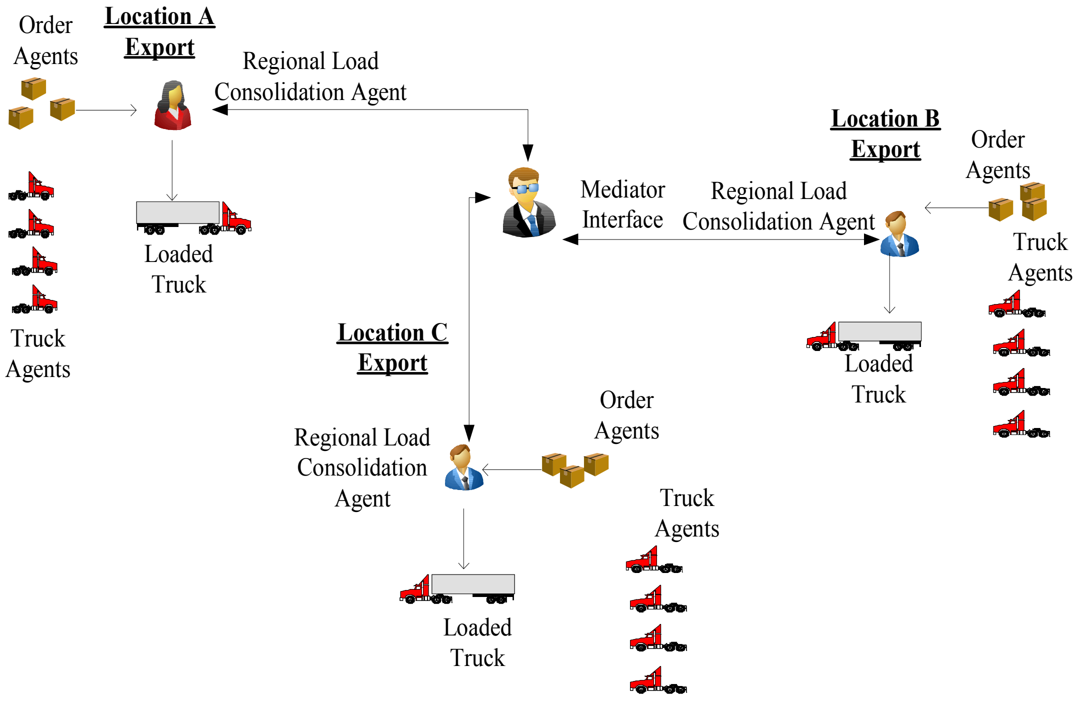


Figure 5. 4. Agent-based load consolidation environment

In some cases, the order agents might not be matched to a proper truck agent because of schedule or capacity limitations of the orders. In such cases the order agents might be routed to some other regional load consolidation agent so as to be assigned/consolidated to truck agents in other regions.

Mediator interface: Mediator interface provides global information base in order to support the regional load consolidation agents. Mediator interface organizes the inter-regional load consolidation agent communication. In other words, mediator interface works like a central switch to route the order agents which can not find any truck within its own region.

The inter-agent interactions are also held in this phase of the design. Interaction diagrams are used while presenting the negotiations between agent types in Prometheus methodology. In interaction diagrams, time increases as one move down the diagram. Each agent has a lifeline, graphed with a vertical line with the agent name. Messages are depicted as horizontal arrows between lifelines. Percepts and actions would also be represented within the interaction diagrams.

There are five main interaction diagrams of the proposed system as;

- Order arrival percept-order agent interaction diagram
- Order-truck interaction diagram
- Truck attribute change percept– truck agent interaction diagram
- Order attribute percept - order agent interaction diagram
- Regional load consolidation - truck agent interaction diagram

5.2.1 Order arrival to system

Scenario: Order arrival to the system

This interaction diagram consists of arrival of a new order by any system resources as a percept (such as telephone call or internet). Order agent dynamically builds itself from the data of customer order form. The time passes to the point of response to the percept is for finding a suitable truck and route. During this process order agent builds necessary connections with regional load consolidation agent to find suitable truck agents to be assigned or consolidated. Figure 5.5 below represents the arrival of order percept within order agent interaction diagram.

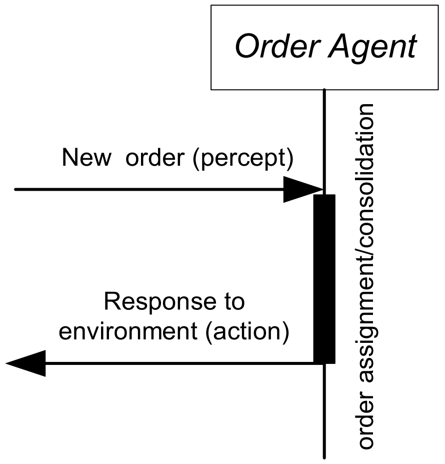


Figure 5. 5. Environment- order agent interaction diagram

After the order agent receives the percept, it spends some time to answer to the user interface about the load consolidation result. The percept obtained by the order agent includes the data about the order. Order agent uses the data of percept to make itself to be consolidated. Time to be consolidated to a proper truck agent depends on the order attribute. If the order has an attribute of *quick response by the system*, then order agent suits in the system up to the predetermined time. Within this time period, if the order agent finds a suitable truck agent it proposes a schedule for the environment (customers), if order agent can not find a suitable truck agent then it is rejected. If the order does not have a *quick response by the system* attribute then order agent exists in the system until to be consolidated or assigned to a proper truck. Therefore the goal of order agent depends on the data arriving with the *new order* percept.

5.2.2 Order-truck interaction diagram

Scenario: Selecting proper trucks from the available trucks within the system:

This interaction diagram operates when the “selecting proper trucks from the available trucks within the system” scenario occurs. This interaction diagram is one of the most critical interaction diagrams which supply the load consolidation mechanism to work. After the arrival of any order to the system via any communication ways (internet, telephone, or face-to-face interview) it enters to the MABLCS as an order agent. After its entrance to the system it behaves as a particular agent within the system according to its pre-determined system goals. By using this interaction diagram, the order agents are routed to a set of truck agents which are previously defined to the system. The regional load consolidation agents route the order agents to respective truck agent set which have a common network area. This increases the system operation performance because the order agents do not have to interact with all the truck agents within the system. However, if there is not any available truck agent (it is also possible) in the region where the order origins, the regional load consolidation agent routes the order agent to other regional load consolidation agents (within any geographical area) by asking to mediator interface.

Figure 5.6 represents interaction diagram between the order agent and the available truck agents. *The regional load consolidation agent does not enumerate all possible load consolidation alternatives.* This is the basic inspiration point behind MABLCS. Load consolidation solution is formed by the negotiations of the order and truck agents. The regional load consolidation agent holds the goal of load consolidation for the sake of reduction of global transportation cost, and matches the relational trucks to any order agent.

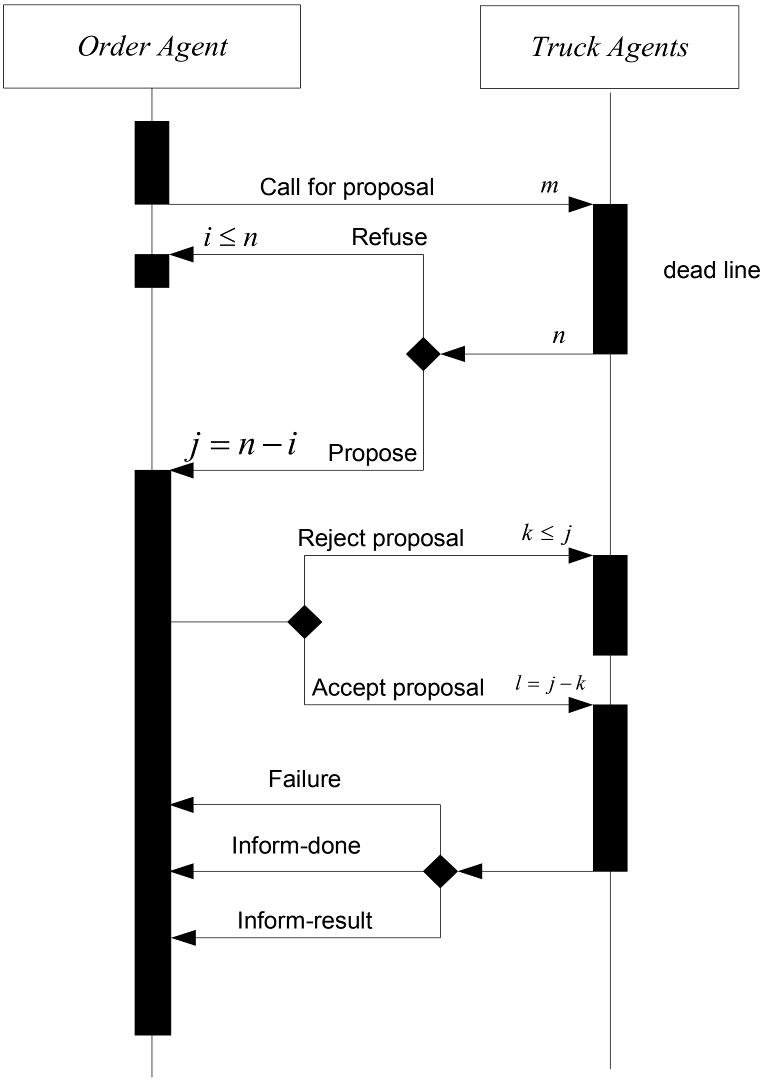


Figure 5. 6. Order-trucks interaction diagram

When an order agent *o* enters to the system and the available truck agents are determined by the regional load consolidation agent, it has to bid to truck agents for a call for proposal. As presented in the figure 5.6, the order agent calls for proposal to the truck agents which are provided by the regional load consolidation agent. If there is *m* number of truck agents within the truck agent set provided by the regional load consolidation agent, the order agents call for proposal to these *m* trucks within the system. *n* is the number of truck agents responding to the call for proposal. *i* is the number of rejection, *j* is the number of acceptance of truck agents. *k* is the rejection number, and *l* is the acceptance number of order agent.

The parameters of τ_i^{AVL} and τ_i^{DLN} represent the earliest pickup time and latest delivery time of any particular order agent respectively. The goal of any particular order agent is to be transported within these time window.

The truck agent which is called for proposal by the order agent in consideration calculates the cost of operation of the order agent if accepting the order agent is feasible for both its schedule and its capacity limitation. Any particular truck agent k calculates a respective cost to the proposal of respective order agent by;
$(k, \text{cost}(C_k + o) - \text{cost}(C_k))$ where C_k is the current content (schedule) of truck k. In other words, the truck agent calculates the cost of transporting the newly arriving order by adding it to its current transport schedule. The order agent gets the cost information from only the truck agents which are feasible to accept the order. It is probable that there is no truck agent which calculates a cost so as to response to the respective order agent because of its pickup, delivery, volume, weight and route limitations. The cost (C_k+o) denotes the additional costs for k when executing o given C_k.

Order agent o receives a set of bids from the truck agents; $B = [(k_1,c_1), \ldots, (k_n,c_n)]$, $n \in IN$, where c_i specifies the costs that truck k_i will produce when executing order o. The order agent o selects the (k_{min},c_{min}). And finally agent o sends a grant to the truck agent k_{min} notifying it that, it will be granted to be assigned with the 7-tuple order agent; $OA_\text{i} \equiv (\mathbf{o}_i, \mathbf{d}_i, \mathbf{v}_i, \mathbf{w}_i, T_i^{ADV}, T_i^{SLK}, T_i^{RES})$. In order to prevent the negotiation protocol problems, truck agent k_{min} informs the agent o about the assignment result. After the consolidation or assignment of the order agent to a proper truck agent, the truck agent starts to make its reasoning about its position and position times.

After the consolidation or assignment of the order agent to a proper truck agent, the order agent starts to wait for the time of being transported. When the order agent reaches a time point where it has to be transported it sends a message to the truck agent previously consolidated or assigned (see figure 5.7).

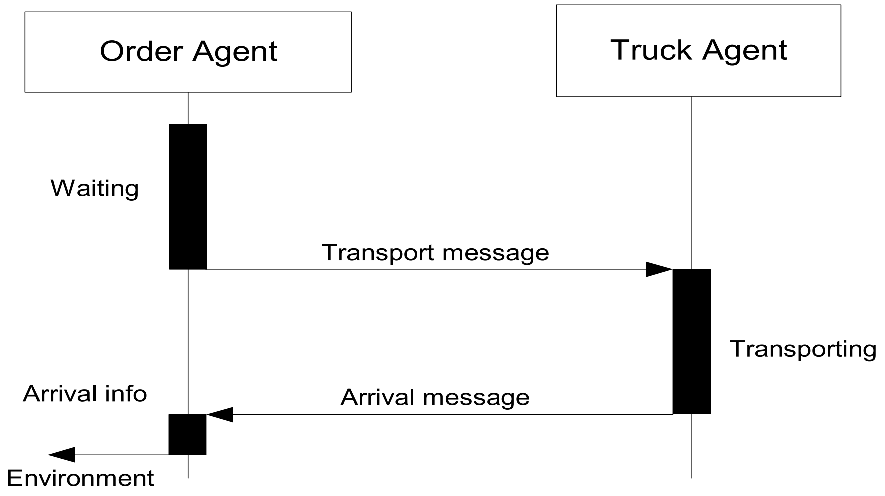


Figure 5. 7. Order – trucks interaction diagram-2

5.2.3 Truck agent-regional load consolidation agent interaction diagram under truck attribute change condition

Scenario: Truck attribute change

This interaction diagram is for the attribute changes of trucks during the transportation activity such as breakdown, break of truck drivers etc. When such an attribute change occurs, the truck agent obtains a percept defining the type of the attribute change. Firstly truck agent evaluates the percept whether if it can be handled. If the truck agent can not handle the problem it throws an action to the regional load consolidation agent to assign the order(s) within the respective truck agent to other truck agents to make the system performing. After the regional load consolidation agent assigns (consolidates) the order which are in the container of truck agent, it responses to the truck agent in consideration. Truck agent evaluates the content of itself and responses to the system interface (dispatch officers and customers). Figure 5.8 illustrates the interaction diagram in consideration.

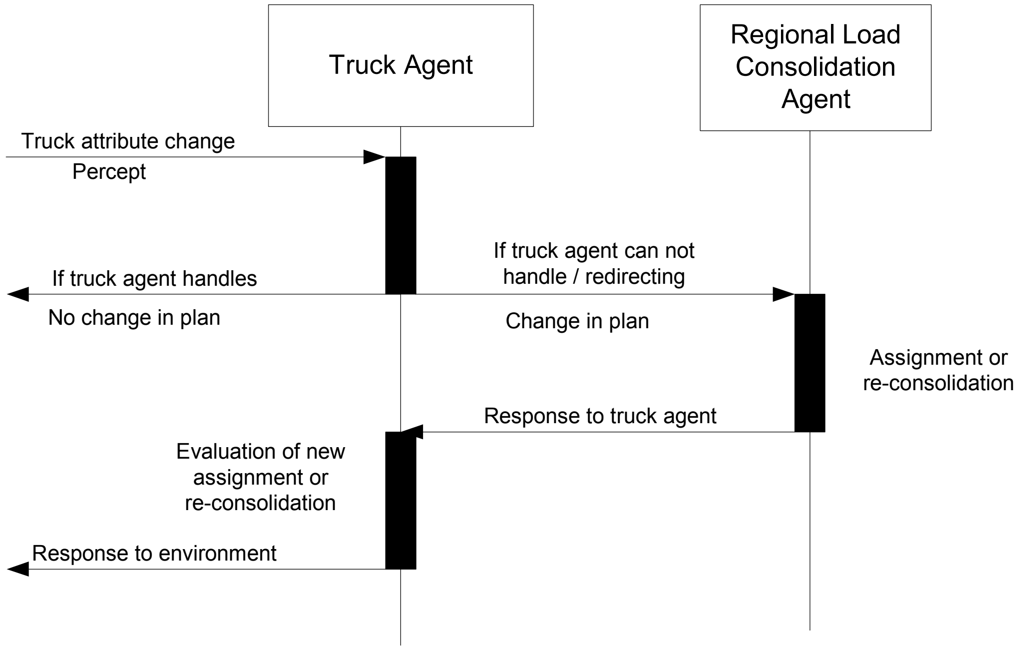


Figure 5. 8. Truck agent-regional load consolidation agent interaction diagram

5.2.4 Order agent attribute change

Scenario: Order attribute change after plan is prepared:

This interaction diagram is for the attribute changes of order agents after they are assigned or consolidated to a truck agent. When such an attribute change occurs, the order agent obtains a percept defining the type of the attribute change. Order agent makes an action by redirecting the percept to the truck agent. Truck agent evaluates whether the order attribute change has an effect on its schedule. If the attribute change within order agent has an effect on truck agent then truck agent sends an action to regional load consolidation agent where the truck agent locates. The regional load consolidation agent consolidates the updated order agent to other available truck agents. After consolidating the order agent by changing/exchanging between trucks, the regional load consolidation agent responses to truck agent about the change/exchange. Truck agent checks its condition and responses to order agent which is in consideration. The truck agent also informs the previously assigned/consolidated orders because change in the truck schedule might require some updates on the orders previously accepted to the truck schedule. And finally order agent responses to system interface about its condition. On the other hand, if the attribute change within order agent has not an effect on truck agent then truck agent directly responses to order agent and as a result order agent responses to system interface. Figure 5.9 illustrates the interaction diagram.

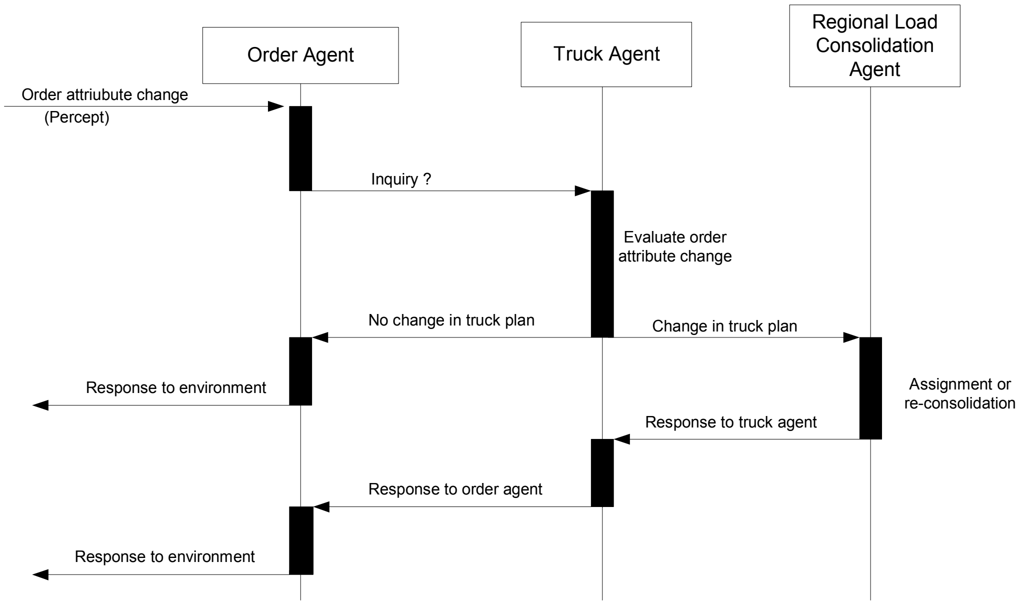


Figure 5. 9. Order attribute change – order agent interaction diagram

Regional load consolidation agent in the figure 5.9 performs the "assignment or re-consolidation" activity by interacting with the truck agents within the system. Regional load consolidation agent interacts with the truck agents which exists in the same region and builds a connection between the order agent which has an attribute change and truck agents in the same region. Order-truck interaction diagram is used during the assignment or consolidation of the orders as coordination between the agent types.

5.2.5 Regional load consolidation - truck agent interaction diagram

Scenario: Change / exchange transportation orders between trucks

The load consolidation goal is also given to the regional load consolidation agent types. This interaction occurs when the "change / exchange transportation orders between trucks" scenario is active in the system. This interaction diagram represents the interaction between the regional load consolidation agent and truck agents within the system. The regional load consolidation agents search for a better load consolidation agent. The regional load consolidation agent interacts with the truck agents with its own initiatives in order to search a better consolidation alternative. If regional load consolidation agent finds any better load consolidation alternative, it responses to order agents and matches them with the available truck agents (by using order-trucks interaction diagram). As a result of this process some order agents might be changed or exchanged between truck agents for the sake of a better load consolidation.

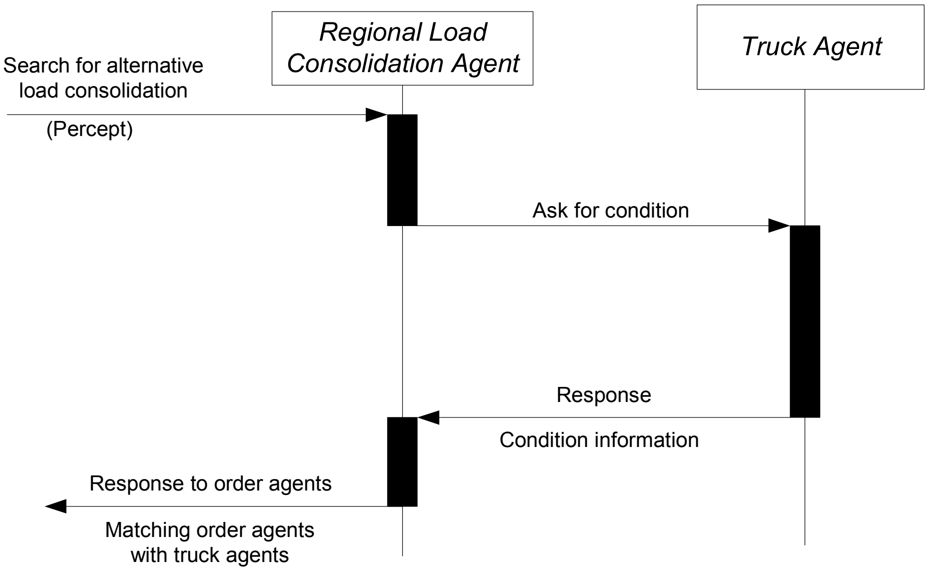


Figure 5. 10. Regional load consolidation agent-truck agent interaction diagram

5.2.6 Interaction protocols

In order to have a complete and precisely defined agent interactions, interaction protocols are derived from interaction diagrams. There are a range of notations for describing for such protocols such as; UML, AUML, Petri nets etc. The interaction protocols are generated as asking the possible sequences of messages between the agent types. The agents might respond to other agents according to their goals and beliefs. The interaction protocols between agent types are represented within the interaction diagrams. For example in figure 5.8, if there is an attribute change of a truck agent then the agent tries to handle the problem, then if it can not handle the problem it re-directs it to regional load consolidation agent. Each scenario in the scenario development phase of this study represents a protocol. For example order-trucks agent interaction diagram (figure 5.6) encapsulates the interaction protocol which includes two types of agents. There are 8 messages within this protocol. Each scenario within an agent system represents a different protocol between agent types.

5.2.7 Message descriptors

Descriptors allow system to gather information about the situations of each entity. For example, the message descriptors between order and truck agents encapsulate all the required information about the order. The truck agents reply to order agent according to the situation of order agents. The message descriptors include the parameters of; *name, description, agents included, purpose, and information carried.* There are 8 different messages between order and truck agents. The first message is call for proposal. This message is between a single order agent and *m* number of truck agents. In below table, a single message descriptor is given for the interaction protocol for the "Selecting proper trucks from the available trucks within the system" scenario. Each message is described in the same format like the one below.

Table 5. 4. Order-truck agent call for proposal message descriptor

Name: Call for proposal
Description: Carries the data about the attributes of the order agent.
From Agent: Order agent
To Agent: Truck agent
Purpose: To find a truck which provides the minimum cost to transport
Information carried: Order attributes such as weight, volume, pick-up point, delivery point, pick-up date, delivery date etc.

5.2.8 Finalizing the architectural design

In the final step of architectural design, firstly the boundary of the agent system is determined. Secondly, percepts, actions and the relationships between these two and agent types are described. Thirdly, the external persistent data and internal data shared by the agent types are described. Finally, system overview diagram is developed. All the modeling information is brought together in the *system overview* *diagram* in Prometheus design methodology. System overview diagram finalizes the architectural design stage of Prometheus methodology.

5.2.8.1 Boundaries of agent based load consolidation system

Agent systems are suited to an environment thus the environment that the agent system is in should be specified. There are some criteria while making the decision of whether to include something within the agent system or leave it within the environment of agent system;

1. Is there any benefit of the agent paradigm with aspects such as autonomy, flexibility and goal orientation?

2. Does keeping it separate make for less coupling?

3. What is the simplest approach to integrating, or interfacing to existing code? (Padgham and Winikoff, 2004)

The boundary of the proposed MABLCS is determined by taking these criteria into consideration. In agent based design of 3PL, the traffic that the trucks are in is the environment that the truck agents are in. The customers and customer requests with dynamic change in their attributes are the environment that the order agents are in. Customs rule, governmental regulations that expose to change are the environment of regional load consolidation agents. This is because regional load consolidation agents are directly expose to the business rules during their operations while consolidating the order agents to truck agents.

5.2.8.2 Describing the percepts and actions

In this step of 3PL agent design the descriptors for the inflow and outflow are developed. Percepts are the information that the agent receives from the environment, while actions represent the effects the agent can have on the external environment. Many percepts result in an update of the agent’s knowledge as well as potential action on the part of the agent. Knowledge updates resulting from the percept should be explicitly identified.

Percepts: The percept identification is obtained by considering the goal of the agents identified in goal specification stage of this design. In Prometheus design methodology percepts are described by; name, description, information carried, knowledge updated, source, processing, agent responding, and expected frequency. Below the "Delay" percept in customs is described as an example for the percept description of 3PL logistics operations decision.

Table 5. 5. Delay percept descriptor

Name: Truck delay in customs
Description: Occurs when the truck agents arrive to customs and wait for the formal transactions
Information carried: Time duration of waiting, reason of waiting
Knowledge updated: Expected delivery date of order agent is updated
Source: All indications from the custom officers and respective customs systems
Processing: When time required to pass the custom is more than the expected time duration
Agents responding: Truck agent responses to this percept and notifies regional load consolidation agent where the truck locates
Expected frequency: Frequently

Actions: Actions, similar to percepts, are the reaction of agent types to the environment in connection with the goals of them. Like percepts, actions are described with a standard descriptor template which include; name, description, parameters, temporality (durational/instantaneous), failure detection, partial change, and side effects. Below the action of "Response to environment after the re-assignment/re-consolidation" is described as an example with a standard descriptor template of Prometheus methodology.

Table 5. 6. Response to environment after the re-assignment/re-consolidation action descriptor

Name: Response to environment after the re-assignment/re-consolidation
Description: Occurs when the regional load consolidation agent re-assigns or re-consolidates the order agents after the truck delay perceived from the environment and redirected to regional load consolidation agent
Parameters: Names of the truck agents in which the load assignment/load consolidation occurs
Temporality: Instantaneous
Failure Detection: No available failure detection
Partial change: Change in the route of truck agents which are in consideration
Side effects: None

5.2.8.3 Defining shared data objects

The persistent data that is used by the system functionalities are designed in the system specification stage. For the shared data, the main issue is the consistency of the data usage. The shared data objects might be persistent or temporary. Some important questions to ask at this stage about persistent data objects are as follows:

- Is there an in-memory version of the data?
- When is the external data updated?
- Are any consistency checks needed on this data-if so what?
- How do agents access this data? (via requests to another agent, via DB access, reading a file, direct access to attributes of data object, via methods on a data object, and so on?) (Padgham and Winikoff, 2004)

In this MABLCS design, the data objects used are represented in figure 5.3. *Order Management Functionality* uses both *Order Data* and *Persistent Database of Orders.* Order Data is not persistent –that is, it exists only at run time. However, *Persistent Database of Orders* is a persistent database as it is referred from its name. In a similar manner, *Persistent Database of Trucks* are persistent and *Truck Data* are not persistent (it exists only at run time). In addition to the databases of Orders and Trucks, the *Business Rules Database* is also persistent.

The Order Data in figure 5.3 holds the data of a specific Order Agent. The data lives with the order agent to the point where the order agent is being transported and arrived to the destination point. Order agent holds the order data during its life-time within the agent system. After the order agent arrives to the destination point, the data within the order agent is recorded to Persistent Database of Orders.

On the other hand, Truck Data object also holds the data about any specific truck agent which is living in the agent system. As mentioned before, the Truck Data object is not persistent. Truck agent lives in the system with the *Truck Data* object. It holds the data about the truck agent such as container capacity, route information, fuel information, geographic location and etc. All this data structures are shared with regional load consolidation agent during the interactions. Some of them are sent to other agents within messages and some of them might be used with an action during interaction with the environment.

In addition to messaging and data sharing, this data objects are also used during the *goal achievements* of agent types. For example, truck agents use the data of consolidated orders in order to calculate the cost for a newly arriving order agent. Another example of usage of shared data is the finding the minimum route for the orders consolidated to the container of any specific truck agent.

In Prometheus design methodology the shared data is described with data descriptors template which includes the following parameters; Name, description, data type, produced by, used by, persistent, initialization and used when.

By using these parameters below a sample shared data is represented for *Truck Data* object

Table 5. 7. Truck data descriptor

Name: Truck data
Description: It holds the container capacity, route information, fuel information, and geographic location data
Data Type: Object
Produced by: Truck agent
Used by: Truck agent, regional load consolidation agent
Persistent: None
Initialization: When a new truck is released to system
Used when: Specified in the interaction diagrams

5.2.8.4 System overview diagram

System overview diagram is for the representation of the entire design process. The design stages up to this stage are represented within a single diagram. The symbols used in this diagram are given in figure 5.11.

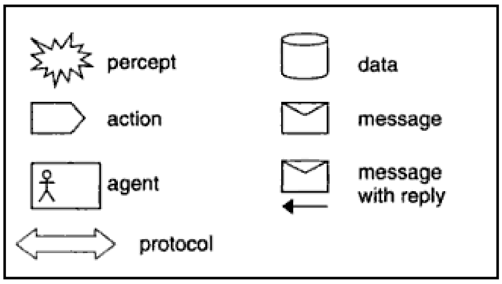


Figure 5. 11. Graphical symbols used in the system overview diagram in Prometheus methodology

Firstly, each agent type is placed in the figure with an agent symbol. Then, it is linked the percepts to agent types that use them, and actions to agents that are responsible for them. Thereafter, an incoming link from each agent to an external data object and an outgoing link from each data object to each agent type that directly accesses its data are added. Double-headed links indicate both read and write.

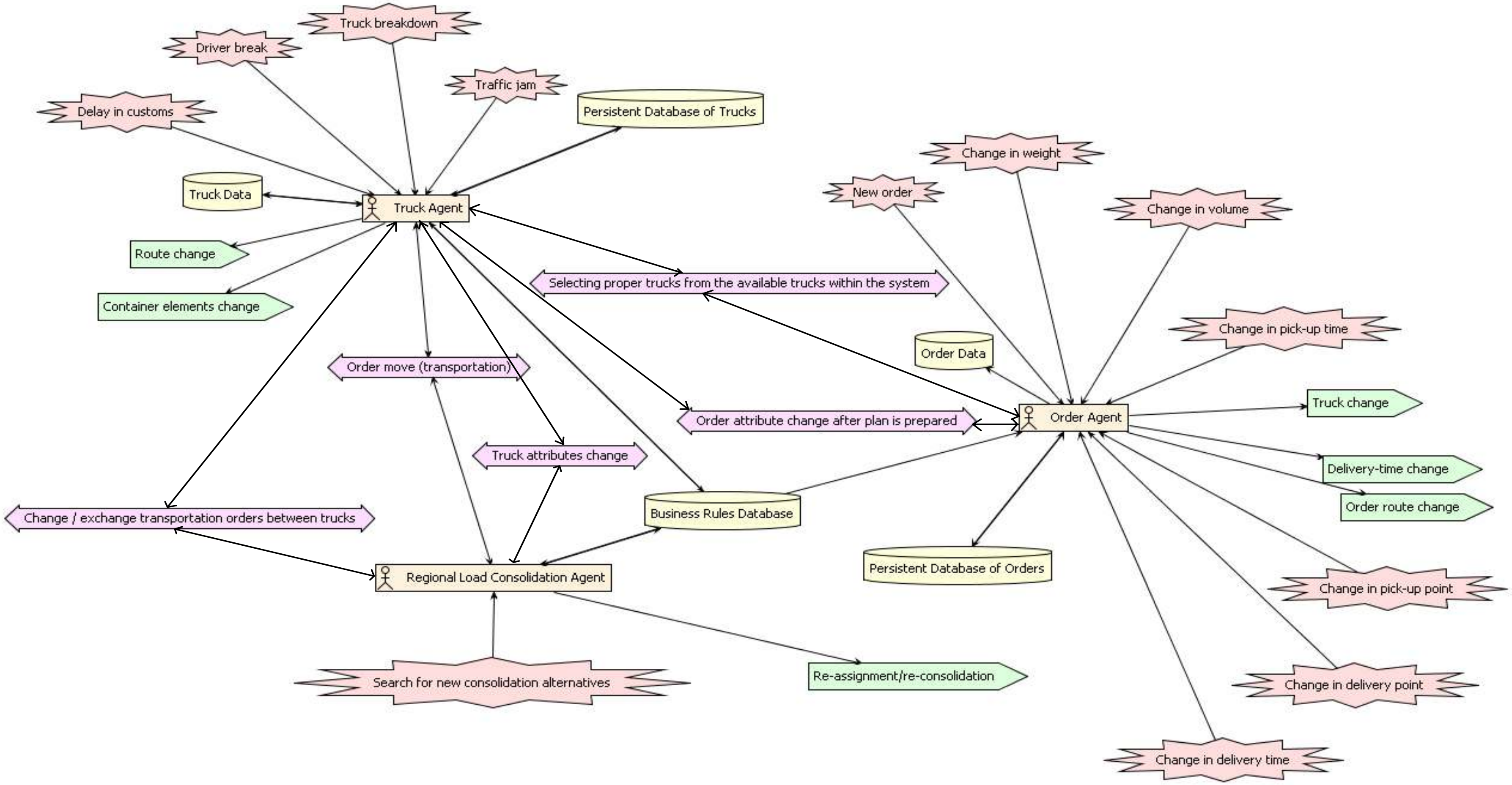


Figure 5. 12. System overview diagram

Note that the system overview diagram in figure 5.12 is not for agent instances. The system overview diagram includes the percepts, actions, protocols between agent types and data objects. Within this proposed MABLCS three types of agent exists which are; truck agent, order agent and regional load consolidation agent. By giving the system overview diagram the architectural design stage is completed. In the further section of the design of the proposed MABLCS, the detailed design will be handled.

5.3 Detailed Design

In the final step of Prometheus design methodology, the proposed MABLCS is designed in detail. System specification and architectural design phase show all the agent types, percepts, actions and data sources that the agent types use together except the capabilities of the agent types. However the reasoning capabilities of the agent types should be designed in detail. In order to design the system in detail the order acceptance/rejection, cost calculation, truck agent load consolidation and regional load consolidation agent decision mechanisms are designed in a step by step manner.

The order agents which enter to the MABLCS negotiate with the truck agents so as to find a truck agent which assign or consolidates it. Therefore the operations of the truck agents (movement, waiting idle etc.) do not trigger the system. The system is triggered with the arrival of a new order agent.

The most critical aspect of agent-based modeling of 3PL systems is the modeling of the BDI reasoning for the agent types in order to make them reach their goals which are defined in the system specification stage of Prometheus methodology.

There are some detailed operational tactics which should be integrated to the truck agent reasoning mechanism. For instance, if the truck agent changes its plan according to the context of the network, it increases the chance of getting the other elements in the destination nodes. Because the plan of the truck depends on the queue of the orders, as soon as the truck agent depart from its current location means that it will reach to the next destination as soon as possible. Therefore it might change its plan by adding the new order agent to its plan. This shows that, truck agents should use their BDI reasoning system to have a human-like behavior.

Truck agent also uses BDI reasoning to re-position itself on the transportation network. This is again for consolidating more loads into its container. The truck agent checks the network context again and it chooses a point to go as empty or to stay in its current position. The truck agent can change its plan of pickup and delivery completely when a new order agent is given grant to be transported. This is because the new order might provide a good chance of capacity utilization of the container. In addition to that, the truck agent may also exclude some order agents from its plan if their response time has not finished yet. When the response time is finished, the truck agent asks for the regional load consolidation agent to change/exchange order with other truck agent if it can do. It tries, if its trial fails then it continues with its current plan. This is another human-like behavior of the proposed MABLCS.

In the real 3PL system, the orders arriving to the system do not have any intelligence to be assigned or consolidated to a truck. However the order agents in the MABLCS have an artificial intelligence that help them to negotiate with the truck agents in the system. The order agents can do that via having the proactive behavior of the BDI agent types. They live in the MABLCS to be assigned or consolidated to a truck agent up to time of being rejected or being transported. Therefore, the assignment or consolidation of the order agents occurs within a self-emerging decision environment within the proposed system. The negotiation between order and truck agents determines the operational decisions.

The detailed design of the proposed MABLCS is performed by designing; *order acceptance/rejection mechanism of the truck agents*, *cost calculation mechanism of the truck agents*, *truck agent load consolidation mechanism* and *regional load consolidation agent decision mechanism*.

5.3.1 Order acceptance/rejection mechanism of truck agents

In the management of transportation logistics systems the most critical point is to make the decision of order acceptance/rejection to the current system. This is because the acceptance/rejection decision necessitates all the system data. According to the flowchart given in figure 2.1, the determination of the list of accepted order requires a good knowledge and management of the system data. This is why so many dispatch officers are employed in the 3PL companies. There are some studies in the literature of the transportation logistics which emphasizes the importance of the order acceptance and rejection and proposes some methods to accept/reject the arriving orders. However these methods are generally proposed for the central management of the truck fleet. Some of the order acceptance/rejection mechanisms are;

- *System and vehicle capacity check prior to load acceptance/rejection:* In their studies Regan et al. (1998) used this load acceptance strategy for accepting or rejecting transportation request by evaluating the number of TL waiting in the queue of overall system.
- *Feasibility based load acceptance*: If no feasible assignment is found, the load is refused. This method is single–pass that the acceptance/rejection decision does not live for a time period.
- *Profit based load acceptance:* In such acceptance/rejection methods the orders which does not contribute to the profitability of the overall system are rejected.

In central order management systems, the acceptance/rejection mechanisms require continuous updates on the status and geographical locations of the system resources. However, the proposed agent system provides the required system dynamics to the respective agent. The agent types live in the system with their own data and with the data of environment up to the point of their life ends.

In the proposed MABLCS, the order acceptance/rejection decisions are made by individual truck agents therefore there is not a central acceptance/rejection decision. The orders are represented by agents which lives in the system with their own attributes and the acceptance/rejection decision are made with a self-emerging mechanism.

When a new order arrives to the 3PL system via electronically or manually the data of the order arriving is used so as to create the order agent which will live within the proposed MABLCS until its transportation ends or it is rejected from the system.

The order agent entering to the MABLCS might be rejected by all of the truck agents which are provided to it by the regional load consolidation agent. However this rejection does not mean that the order agent is permanently rejected from the MABLCS. As it is presented in the figure 5.6, order agent calling for a proposal might be rejected by all of the truck agents because its ***7-tuple*** attributes might not fit to the schedule and capacity of any of the truck agents. However, the order agent behaves proactively to be assigned / consolidated to a feasible truck. The order agent which is temporarily rejected at the end of the negotiation protocol by all of the truck agents ***waits*** within the system and tries to be assigned or consolidated until its response time finishes ($\tau_i^{ARV} + T_i^{RES} < CurrentTime$). Figure 5.6 only represents a single session of negotiation between a single order and available trucks agents. The order agent could hold this negotiation session more than one so as to find a truck agent to be assigned or consolidated.

On the other hand, getting a grant from a truck agent does not mean a permanent assignment/consolidation of the order agent to respective truck agent. This is because when the order agent is assigned/consolidated to a truck agent, truck agent first checks the response time (T_i^{RES}) of newly accepted order agent. Truck agent does not react until the end of the T_i^{RES} of the newly accepted order agent. When the end of the T_i^{RES} is reached then truck agent makes reasoning about the contribution of the order agent to its transportation profitability. If accepting the order agent is not profitable for the truck agent then truck agent rejects the order before T_i^{RES} of the order agent ends.

The truck agent functions like a merging point for the order agents to be consolidated. Therefore the truck agent has the flexibility of rejecting an order agent up to T_i^{RES} of order agent in consideration. In this point, there are two load consolidation mechanism is run by both truck and order agents.

- **Order agent mechanism:** The order agent tries to find another order agent to be grouped after its so-called acceptance (order agent does not know the final decision of truck agent until the end of its T_i^{RES}). This is because there exists a possibility of rejection at the end of the T_i^{RES} . This is why the order agent continues to search a better assignment/consolidation alternative. In other words, the order agent does not have a guarantee of assignment/consolidation till the end of its T_i^{RES} ends.

- **Truck agent mechanism:** The truck agent might accept another order agent while it is waiting for the T_i^{RES} of previous order agent. Therefore truck agent functions like a merging point of order agents. When a new order is consolidated with the previous order agent which is pending for it's T_i^{RES} , the previous order agent is informed about the acceptance of the new order agent.

Therefore, we can conclude that the higher T_i^{RES} for an order agent means that a higher possibility of consolidating with other order agents in the system (this is why the dispatch officers in 3PL companies request time from their customers to make a good decision).

As explained in the previous section, the negotiation between any particular order and truck agents starts with the call for proposal (*cfp*) of the order agent to the truck agents. When the particular truck agent gets the *cfp* from the order agent, it also gets the ***7-tuple*** attributes of the order agent which is calling for proposal. Truck agent first checks its schedule whether if it can accept the order into its schedule. In order to check, it first inserts the order agent to its schedule and it tries to find a feasible schedule which includes the order agent which is calling for proposal. If the order agent fits to the schedule of the truck agent, then truck agent checks for the capacity limitations (both for weight and volume capacities). It is because; at any time in its transportation the occupied volume or occupied weight can not exceed the truck default volume and weight capacities.

In the figure 5.13, the list of accepted orders of a particular truck agent is represented. These data is stored within the *truck data* object of the truck agents (see figure 5.13). o_1, o_2 … o_n are the previously assigned/consolidated orders to the truck schedule.

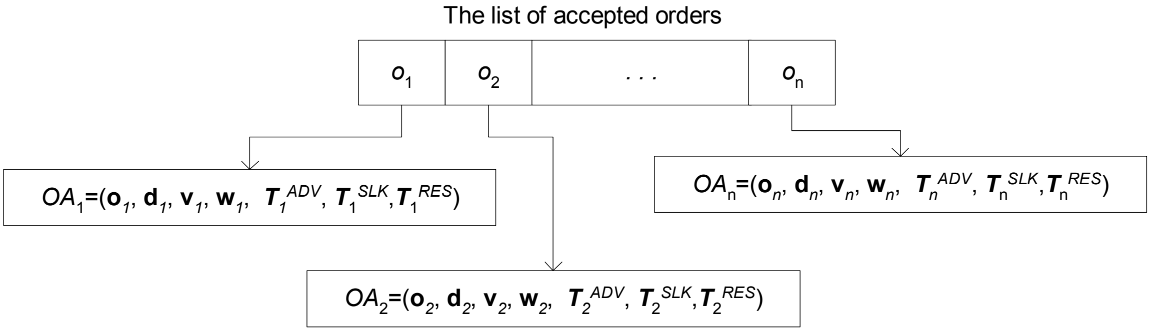


Figure 5. 13. Truck agent accepted order list

When the truck agent gets the *cfp* from the order agent it stores the order attributes within its beliefset to check its schedule with the newly arrived order agent. Truck agent holds the data of accepted orders for its future operation. The truck agent holds a special beliefset to check the feasibility of any order agent calling for proposing. It holds one operation element for the pickup and one for delivery of any particular order agent. Therefore, the truck agent can control the feasibility of any order agent by checking its both pickup and delivery time windows. Therefore, upon the acceptance (it might be a permanent or temporal acceptance in the future) of any order agent to its schedule, truck agent creates two operation elements within its schedule. In the further section of this study *operation element* term will be used in order to refer to any pickup or delivery operations. The truck agent builds its beliefset in order to generate solutions and test their feasibility. The following figure shows a sample pickup and delivery schedule of a particular truck agent.

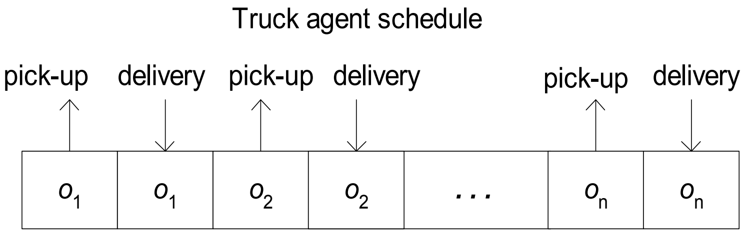


Figure 5. 14. Truck agent schedule representation

The data of order agents which are previously accepted by the truck agent are recorded within the truck agent schedule with respect to their sequence. When a new order agent calls for proposal, the truck agent inserts two data object (one is original for pickup operation and the other is carbon copy of the original for delivery operation) to its beliefset. After inserting the data of the new order agent, it tries to find a feasible schedule to pickup and deliver of newly arrived order with the previously accepted orders. Truck agent uses a simple heuristic to find a feasible schedule. First, it generates an initial solution which includes the candidate order agent. In each candidate schedule generation, the truck agent first checks the pickup and delivery constraint of the generated solution. If the generated solution is feasible for the pickup and delivery time windows, then it checks for capacity and calculates the cost which will be proposed to respective order agent. In other words truck agent tries to find a solution for its own vehicle routing problem after inserting the order agent which is calling for proposal to its schedule.

For the truck agent the feasibility of the generated solution is a prerequisite before calculating the respective transportation cost. It checks the feasibility of any schedule by using a backward algorithm. The element before the last element of the schedule is checked first. The checking algorithm depends on the type of the operation (whether it is a pickup or delivery). There are four possibility of sequence of any two consecutive operations which are; pickup-pickup, pickup-delivery, delivery-pickup and delivery-delivery. If we present the generated candidate solution with the following string;

o_{ip} $o_{id\dots}$ o_{np} o_{nd} where o_{ip} represents the pickup of the order agent i and o_{id} represents the delivery of order agent i. Truck agent checks whether it can deliver the order agents before their latest delivery time. And it checks whether the time of pickup of the orders are not earlier than the earliest pickup time of the orders (see figure 5.15).

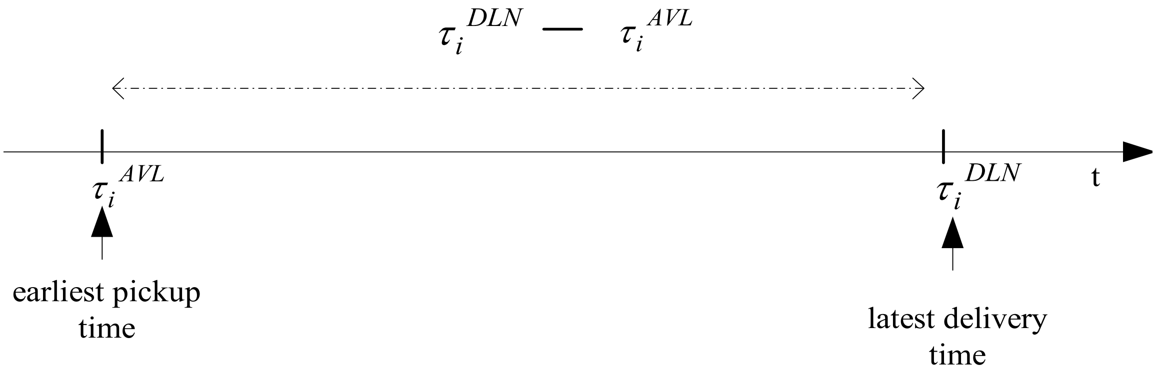


Figure 5. 15. Time window of a typical order

Therefore there exists an epoch which represents the *latest pickup time* for each order in the schedule. For the truck-load (TL) operations it is the difference of latest delivery and transportation time of particular order (τ_i^{DLN}- L_i). Truck agent can use the τ_i^{DLN}-L_i statement if there is only one order in its schedule. Therefore for TL operations any particular order which is picked up within the time interval of τ_i^{AVL} and (τ_i^{DLN}-L_i) and delivered before its latest delivery time is feasible for the truck agent. However, it is not the case in LTL operations of 3PL systems because truck agent might pickup consecutive order without making any delivery operation. As a result, the latest pickup time of any particular pickup *operation element* is ***schedule dependent.*** Therefore, truck agent should check the feasibility of latest pickup time of any pickup *operation element* recursively.

If the generated solution string (o_{ip} $o_{id\ldots}$ o_{np} o_{nd}) is feasible according to the time windows of the order agents accepted to the schedule, the truck agent checks for the capacity limitations at each pickup times (or points). Each truck agent is created with its own volume and weight capacity according to its physical properties. Therefore truck agent should check its own volume and weight capacity when the proposed solution string is under checking process. The truck agent holds its current volume and weight utilization (referred as *occupied volume* and *occupied weight*), future network position and future volume and weight utilization in its data object in order to check the capacity feasibility of its schedule. This is because the truck agent can negotiate with the order agents while conducting a transportation operation.

As stated previously, the truck agent should solve a vehicle routing problem with time window (VRPTW) of its own accepted order set. It uses its own simple search heuristic to find a feasible routing plan by considering all time windows of the accepted orders and capacity limitations of the containers. Even if solving a VRPTW optimally is a very difficult task when routing a ***fleet***. However, the truck agents solve only its simple VRPTW therefore it is not so difficult for the truck agent to find a feasible schedule (in general 3PL operations at most 4 orders are consolidated within a container, this data is obtained by interviewing with the 3PL company operations personnel).

If the order agent calling for proposal is feasible for both the schedule and capacity limitations of the truck agent then the order is accepted *temporarily*. The term temporarily refers that the order might be rejected any time between its acceptance and its T_i^{RES}. Then the cost of the proposed schedule is calculated and sent to the respective order agent by the truck agent. After sending the cost of transportation to the respective order agent, truck agent continues its operations according to its schedule; truck agent also continues to operate during its negotiation with the order agent. This means that, truck agents might change their schedule (and also routing) while they are in transportation operation. The truck agent does not know its next operation certainly (in 3PL operations trucks are routed dynamically according to changes occurring in the system). Cost calculation of the truck agents are given in the next section.

5.3.2 Cost calculation mechanism of truck agents

Cost calculation of truck operations depends on the platform where the proposed MABLCS is implemented. Because cost centers might change according to the structure of the 3PL system. And, the revenue of the 3PL companies depends on their own transportation tariffs. However, the revenue earned by the truck agents is generally proportional to the distance and physical attributes of the order that it transport (their respective weights and volumes). Therefore during the negotiation process between the truck and order agents all the order attributes are evaluated by the truck agent (order agent's pickup point, delivery point, pickup and delivery time windows, volume, weight and etc.).

The transportation operations in 3PL system are not stick to a standard schedule. Therefore we can not fix the transportation cost according to trip types. In other words there is not a scheduled trip that the 3PL companies serve. Therefore, truck agent can not determine a *fixed cost* for its operations. This is why the MABLCS is proposed for such operation decisions.

Truck agent calculates the cost of its current schedule in order to response to order agent which is calling for proposal. Therefore, the direct and indirect costs are calculated according to the current schedule of the truck agent.

Within the proposed MABLCS any truck agent has not a constant schedule (there is not a trip). Therefore truck agent should calculate the cost of its schedule every time a new order calls for a proposal or whenever a change occurs in the truck schedule due to some order cancellations, truck breakdowns and etc. Then total cost is calculated by summing the *direct cost* and the *indirect cost* occurring during the transportation of all the operations elements in the truck schedule. The indirect costs are the overheads while the direct costs are the operation dependent costs which occur when the system operates.

The direct cost is divided into three parts;
1) Empty travel cost; **2)** Loaded travel cost; **3)** Route dependent cost (schedule dependent cost such as tolls, ferry tickets and cost occurring while crossing through multiple countries)

Therefore the truck agent should consider all the direct costs during responding to the order agent in negotiation protocol (**Note:** In this study, the driver wages are assumed to be included within the travel cost. It is assumed that whenever truck agent makes a transportation operation a driver is available). The indirect costs are the overheads occurred during the transportation operations such as fleet maintenance, truck depreciation and etc. The 3PL companies hold their overheads data within their accounting records.

Truck agent responses to the order agent with a unit cost during the negotiation session (see figure 5.6). The unit cost is calculated by dividing total cost occurred for its ongoing schedule to the total amount of *distance x weight*. The following equation presents how the truck agent calculates unit transportation cost in the proposed MABLCS;

$$UC_i = \frac{uic \int_0^{es} \omega(x)dx + \sum_{i=1}^{n} C_{dc}}{\sum_{i=1}^{n} weight_i distance_i} \qquad \left(\frac{\$}{ton*km}\right) \tag{5.1}$$

Where UC_i denotes the unit cost that will be proposed to calling order agent *i*; the variable $uic \int_0^{es} \omega(x)dx$ is used in order to represent the total amount of *weight x distance* of the current schedule of the truck agent; *uic* denotes the *unit indirect cost* of transporting each weight for each distance; *es* denotes the end of schedule point; *n* is the total number of orders in the truck schedule ($\sum_{i=1}^{n} C_{dc}$ represents the direct cost occurring during picking up and delivering all the *operation elements* within the truck schedule including the order agent calling for a proposal); $weight_i$ is the weight attribute of the order agent *i* and $distance_i$ is the distance between the order agent *i*'s origin and destination.

In this thesis study, the revenue is assumed proportional to the distance and weight of the order agent. For instance, if an order has a weight of 5 tons and the distance between its origin and destination is 1,000 kilometers then the revenue is calculated as 5,000 km x ton x tariff. Tariff is the price that the 3PL sells each *km x ton* transportation service.

Due to the dynamic nature of the proposed MABLCS, truck agent only takes care about the profit just before the T_i^{RES} of the next order element in its schedule ends. If the unit cost provided to the order agent during the negotiation protocol is less than the tariff (*$ / distance x weight*) then truck agent fixes the order agent to its schedule. In other words the order agent is permanently accepted by the truck agent. In contrast to this condition, unit cost provided to the order agent during the negotiation protocol might be more than the tariff (*$ / distance x weight*). In such case, truck agent permanently rejects the order agent. This time epoch of rejection is the end of the load consolidation chance of the order agent. Here, truck agent waits till the end of the response time of the next order via behaving proactively. By doing so it increases the chance of load consolidation. Also in practice, if the orders in the truck schedule do not contribute to truck profitability then they are rejected at the end of their response time. This is because there might be another last minute order that can be merged with the pending order.

This mechanism provides MABLCS to assign or consolidate any order agent even if there is no truck agent who permanently accepts it. The purpose of the truck agents are the occurrence of future potential consolidation. This mechanism is done by both the truck agent and order agent reasoning plans.

When the order agent is assigned temporarily to the schedule of the particular truck agent; regional load consolidation agent tries to match the order agent with other truck agents that might fix the order agent to their schedules. Therefore the performance of the regional load consolidation agent also affects the performance of the total system. Regional load consolidation agent holds the negotiations between the order and truck agents in order to merge the orders. The truck agent which the orders are assigned before their T_i^{RES} end performs the task of being a buffer for the future possible transportation orders. This behavior of the proposed MABLCS is human-like, in the real 3PL operations any order which has a T_i^{RES} time remaining are not rejected from the context of the system. It is being waited within the system up to the point of response time. The details of the reasoning of the truck agents on cost calculation and rejection will be analyzed in the experimentation section of this thesis.

5.3.3 Truck agent load consolidation mechanism

The truck agents within the MABLCS have the goal of consolidating the orders so as to reduce the total transportation cost. When any truck agent accepts to transport any particular order agent then it has the opportunity to adjust its time of dispatch. And, the dispatching time of the trucks affects the amounts of orders consolidated into its container. As presented in the figure 5.16, the truck agent can use its own initiative to wait to the *latest pickup time* of the order that is in consideration or just pickup the order in consideration on the time of its earliest pickup. Waiting till the latest pickup time of the order in consideration on the origin point of the order agent increases the chance of consolidation other order agents which also have the same origin into its container. However, waiting till the latest pickup time might result in some loss of load consolidation opportunity on the consecutive pickup or delivery points. Truck agents use their internal reasoning during the decision making process of their own. While making the decision of waiting on a specific point, truck agents uses the system scope data. Truck agent behaves according to its context. The goal of load transportation is therefore given to individual truck agents.

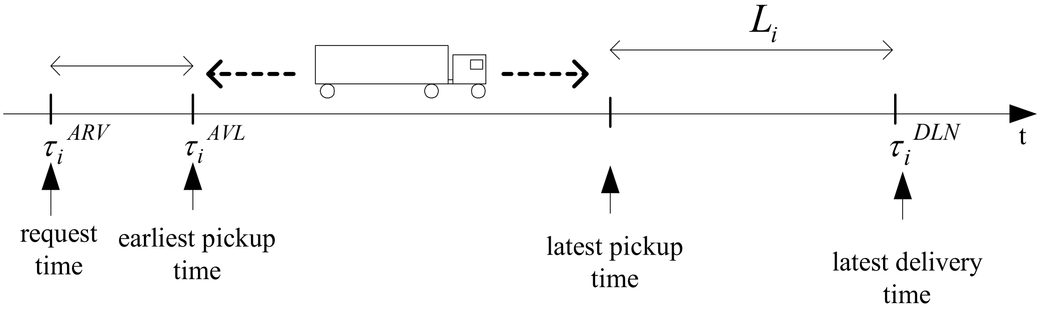


Figure 5. 16. Flexibility of pickup time of any particular orders

If there are potential orders which might be consolidated at the further pickup or delivery points of the truck schedule, then the truck agents do not wait for the latest pickup time of the current order at order agent's origin point. Therefore it makes some adjustments in its transportation schedule. By making adjustments in its schedule it tries to achieve a good load consolidation alternative. Figure 5.16 is valid when the truck agent transports only a single order at a time. However the proposed MABLCS is designed for LTL orders. Therefore the latest pickup time of any order is *schedule dependent*. The load consolidation reasoning mechanism might occur in two different ways; one is before any delivery point and the other is before a pickup point. Because the proposed system is for LTL then there is need for the distinction between pickup and delivery operation elements. The details of the truck agent reasoning are as follows;

Before delivery point

If the next *operation element* on the hand of truck agent is a *delivery operation* then the arrival time of truck agent to the delivery point depends on its load consolidation reasoning. The latest delivery time of the *operation element* is known exactly by the truck agent (the order agent attributes are recorded in the truck beliefset). Therefore, truck agent might have a slack time that provides it the flexibility of going to delivery point as soon as possible or waiting at previous location and then going the delivery location.

Waiting in its current position provides truck agent to capture the potential orders on its current position. If there are some order agents waiting to be assigned/consolidated to a truck agent at the point of delivery then truck agent might select another plan to directly go to the delivery point and spend its slack time there. Therefore it increases the chance of consolidating the orders into its schedule or container at the delivery point. Figure 5.17 summarizes the truck agent load consolidation mechanism before a delivery operation.

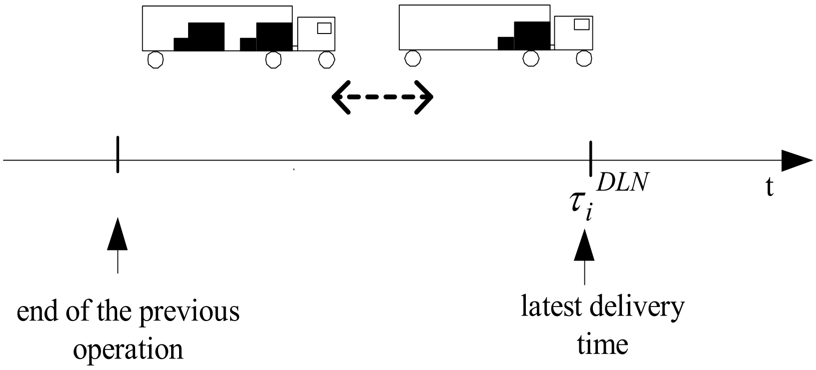


Figure 5. 17. Load consolidation reasoning of truck agent before a delivery operation

Truck agent makes reasoning for the time spent at the delivery point of the order element on hand. Truck agent can select an epoch to arrive to the delivery point which is within the limit of arrival time and latest delivery of the order on hand. Of course, the truck agent could spend the slack time provided to it anywhere between the previous operation and the current operation (it can reduce its speed during making the delivery operation). In other words, truck agent has the flexibility of suspending the delivery of the order element on hand till the latest delivery to catch up the potential orders which might call for proposal on the point of the previous operation (type of the operation element is not important). Or truck agent can deliver the order element on hand as soon as possible to reach to further stops as soon as possible to catch up the potential orders which might call for proposal at the further stops of the truck agent. In fact the decision made by the truck agent is the time to wait at its current position. Truck agent makes reasoning according to demand context of the transportation network. Figure 5.17 summarizes this context. Here the truck agent needs the data of number of order agents waiting in the system on its route, whose T_i^{RES} have not finished yet. This data is provided to the truck agent by the regional load consolidation agent. The reasoning mechanism of the truck agent makes it to make a dispatching decision.

Before a pickup point

If current order element of the truck agent is a pickup operation then truck agent knows the exact pickup time of the order element. However, there might be some slack time between its previous operation (pickup or delivery) and the latest pickup time of the operation on hand. Therefore truck agent should make reasoning about the dispatch time of its previous position to the pickup point of the current operation. If it directly goes to the pickup point then it would have chance of waiting at the pickup point of the current operation. On the other hand if it waits at the point of previous operation it can catch the future potential orders at there. Therefore the truck agent should select a plan to implement its goal of load consolidation. However, the adjustment in its schedule does not affect the overall schedule of the truck agent. It just utilizes its consecutive slack time to reach a better consolidation alternative without affecting its schedule. Figure 5.18 summarizes the selection of dispatch time of the truck agent.

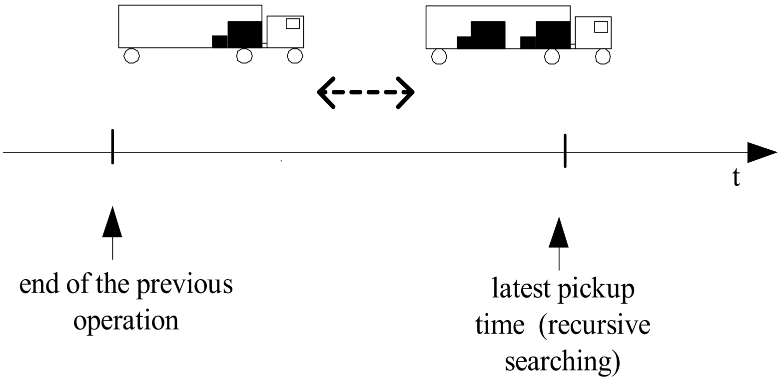


Figure 5. 18. Load consolidation reasoning of truck agent before a pickup operation

Finding the latest pickup time of the pickup operation element on hand is not a straightforward task for the truck agent when we compare it with the latest delivery time of any operation element. This is because the latest pickup time of the current operation depends on the remaining operations in the truck schedule. Therefore it is schedule dependent. For instance, if the next operation in the truck schedule after the current operation is also a pickup operation then truck agent would not obtain the latest pickup time directly subtracting the distance duration from the pickup time of the next element. This is because it is not known directly whether the pickup time which is used for the next operation element is the latest pickup time. Therefore truck agent should search the first delivery operation after the current pickup operation recursively. When it reaches to the first delivery operation in its schedule then it subtracts the distance time from the latest delivery time of the first delivery operation in its schedule. Then it subtracts other distance elements from the value previously found recursively. Finally pickup operation elements come up with a latest pickup time.

Figure 5.19 is given to show how the latest pickup time is obtained for the current operation. Figure 5.19 presents a sample truck agent schedule which includes six operations elements. According to the schedule, truck agent should determine the latest pickup time of the o_1 to make the decision of its dispatch time from its current position (current position might be a pickup, a delivery point or a point where the truck agent waits as empty). Then, the truck agent should make a recursive operation to calculate the latest pickup time of o_1. While calculating the latest pickup time of

o_1, the truck agent should know the origin and destinations of the orders in its schedule.

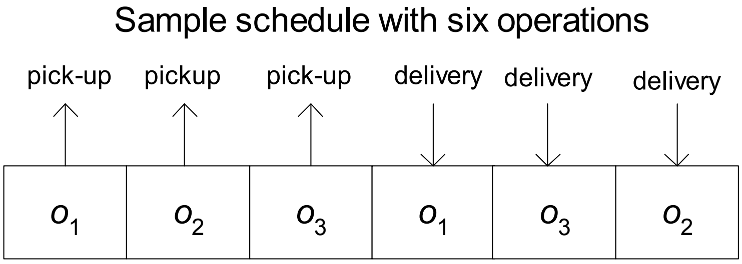


Figure 5. 19. Sample schedule with six operations elements

Truck agent reaches to the first delivery operation in its schedule, in this example it is the delivery of o_1. Truck agent gets the latest delivery time of o_1. Then it subtracts the time between delivery position of o_1 and pickup point of o_3 from the latest delivery time of o_1 then it obtains the latest pickup time of o_3. Then truck agent subtracts time distance between pickup points of o_3 and o_2 from the latest pickup time of o_3. Therefore truck agent finds the latest pickup time of o_2. And finally, truck agent subtracts time distance between pickup points of o_2 and o_1 from the latest pickup time of o_2. And therefore it comes up with the latest pickup time of o_1. And then decides about whether it has a slack time to wait in its current position.

5.3.4 Regional load consolidation agent decision mechanism

Truck agent tries to achieve its goal of load consolidation by its own reasoning system. It behaves proactively to adjust its waiting time in its previous pickup or delivery point. Therefore, it decides on its operations locally. However, regional load consolidation agent directs the arriving order agents to the set of truck agents which are suitable for assignment or consolidation (by considering their origin, destination, volume, weight and etc.). Therefore regional load consolidation agent has the capability of accessing the general regional system data. It holds the negotiation between the truck and order agents. By doing so, regional load consolidation agent performs its initial load consolidation goal. It reduces the number of negotiations between order and truck agents.

The regional load consolidation agent has also the capability of re-matching the order and truck agent after the assignment/consolidation of any particular order to a particular truck agent. Therefore, it tries to find any alternative load consolidation alternatives after the order within its region is assigned or consolidated to a truck agent. The arrival of a new order to the system or any attribute change in truck or order agents might trigger the regional load consolidation agent to behave proactively in the system. This is because after any assignment or consolidation of an order to a truck agent the arrival of a new order might provide new consolidation advantages. Or some environmental changes might drive the system to find a better consolidation alternative. As a consequence of re-matching the order-truck negotiation protocol some orders might be changed/exchanged between truck agent schedules so as to reach a better load consolidation alternative. Figure 5.20 represents the order exchange mechanism between truck agents.

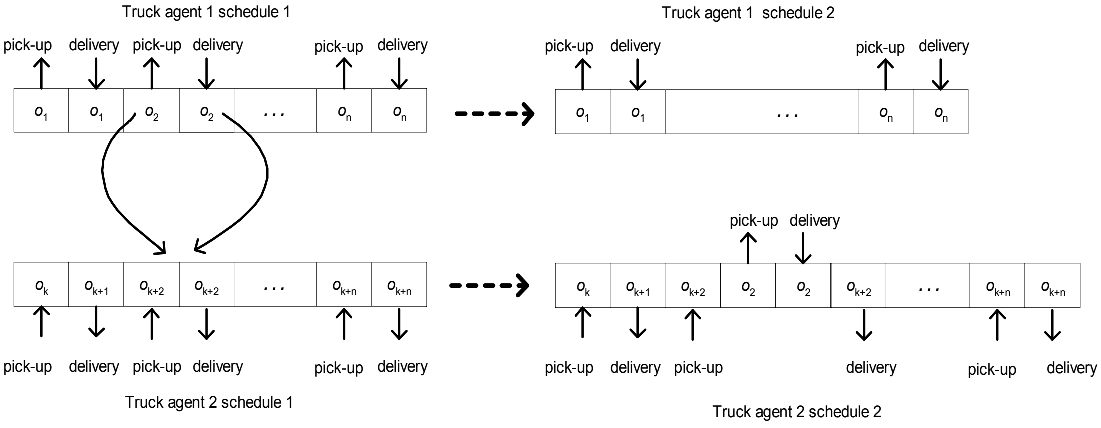


Figure 5. 20. Truck agent schedule change after holding negotiation by the regional load consolidation agent

There are two different truck agent schedules in the figure 5.20. The particular order agent (o_2) in the schedule of truck 1 makes its own reasoning with the support of regional load consolidation agent and thinks that being consolidated with the order agents within truck 2 would result in a better consolidation alternative while negotiating with truck 2. Order agent (o_2) needs the help of regional load consolidation agent to select the proper truck agent to negotiate otherwise it has to ask to all the truck agents within the system for possible load consolidation alternative and this might result in a high computational time for the platform where the agents are running. Of course it might be a method for the order agent to negotiate all the truck agents within the system to be re-consolidated however the computational complexity problem arises respectively.

The order agents hold the data of assignment/consolidation to a truck agent in its beliefset. When the regional load consolidation matches the order agent with a set of truck agents, it directly starts to perform the same protocol as in the previous assignment (it runs the protocol provided in the figure 5.6). The result of the negotiation with the set of the truck agents might result both in the rejection of the orders and in acceptance of the order with a specified cost for the order agents. If new negotiation results in with a reduced cost of transportation then the order is changed between the truck agents by deleting both of the order elements (pickup and delivery) from the truck 1 agent and inserting it on the truck 2 agent. The critical point here is that, the regional load consolidation agent matches the order agents which are not picked up. When any order is loaded to the container then they can not be changed / exchanged between truck agents (in order to protect system from resulting in logic errors). In other words both of the order elements of a particular order agent must be on the schedule of the truck agent; otherwise there might be operation errors which cause unexpected logic errors. The change/exchange mechanism is performed by the synchronization of the deletion and insertion mechanism on both truck agents. If there exists a synchronization problem, there might be operation errors which are not easy to detect on the software agents. Here it is assumed that, the change/exchange decision between truck agents is so fast when we compare it with the transportation operation durations, that there is not an atomicity problem in the system. In other words, the time of the getting the next order element from the schedule for a truck agent does not happen in the same time of the schedule change. The schedule atomicity is guaranteed by the truck agent. Only one operation can access the truck schedule on a single time point. If both of them occur simultaneously then there would be some loss of data.

The function of the regional load consolidation agent and the reasoning of order agent together bring a dynamic schedule for the truck agent where at every time there might be a change of schedule of a particular truck agent. Therefore, the cooperation between **order - truck - regional load consolidation** agent types results in a dynamic load consolidation system where truck agent could change their pickup and delivery plan dynamically to adapt itself to newly arriving orders even after permanently accepting an order agent.

Regional load consolidation agent operates also in the situations where unexpected stochastic events occur. There might be some different events which change the system attributes. Some of which are;

- Truck delay due to traffic or breakdown
- Order cancellation
- Order attributes change (last minute changes) and etc.

In such events, the regional load consolidation agent plays an important role where it matches the relational order and truck agents to hold the interaction protocol between them to react to stochastic events occurring while pursuing its load consolidation goal proactively.

In the case of stochastic events, regional load consolidation agent first checks the event type and it makes reasoning about the event. For instance, if the unexpected event is an order cancellation then the order agent asks to regional load consolidation agent to make the respective truck agent be aware about the event. Then, truck agent changes its schedule while removing the cancelled order from its schedule. However, removing an order agent from a truck agent's schedule might result in cost changes of other order agents within the truck agent's schedule. This is because removing any transportation element might result in a route change and so it might change the cost of its current operations. If the cost is changed for the remaining order agents in the truck schedule (it is the general case), regional load consolidation agent makes the order agents aware about the new condition. After informing them about the new cost of transportation, regional load consolidation agent triggers the order agents to start a new negotiation with some other available trucks to find a better load consolidation alternative. Therefore the agent types in the MABLCS work collaboratively to obtain a better clustering of the orders.

CHAPTER 6

IMPLEMENTATION

The proposed MABLCS is implemented within JACK ^{TM} IDE. JACK ^{TM} is one of the most popular agent development environments working under Belief/Desire/Intention model of rational agency with extensions to support the design and execution of agent systems where team structures, real-time control, repeatability and linkage with legacy code are critical. The major features of the JACK™ Agent Language (a mature implementation of the BDI paradigm written as an extension to Java™), and the JACK™ Development Environment (JDE) that it provides graphical tools to support the design, implementation and execution tracing of BDI agents (Evertsz et al., 2003).

This chapter of the thesis explains how the agent types which were designed with the Prometheus design methodology are implemented (coded) to JACK ^{TM} IDE, how the negotiation between agent types are implemented and how the BDI reasoning capabilities are supplied to the agent types.

6.1 Order-Truck Interaction Diagram Implementation

The most critical section of the implementation of the proposed MABLCS is the implementation of the order-truck interaction diagram in JACK ^{TM} platform. This is because all the other interaction protocols are interconnected with this section of the application.

Figure 6.1 represents sample negotiation protocol of the order and truck agents in JACK ^{TM} which was previously designed in Prometheus design methodology. While implementing any design on JACK ^{TM}, design elements can be dragged and dropped to the design area from the project window of the IDE.

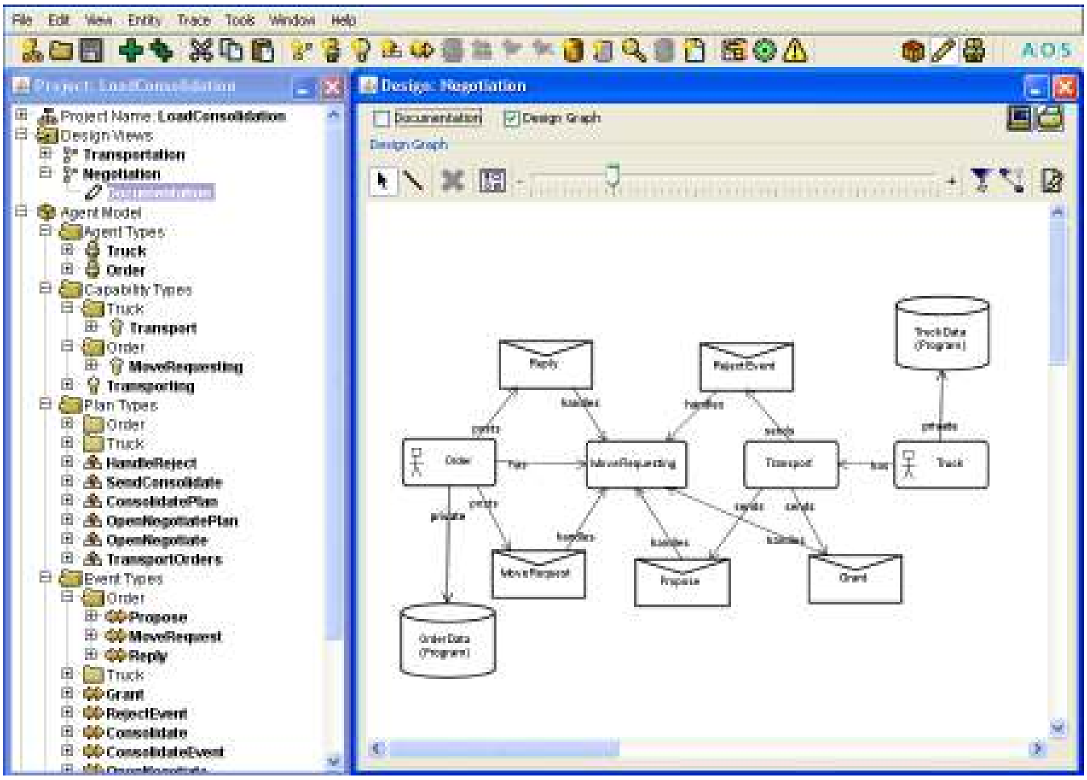


Figure 6. 1. Sample messaging between order agent and truck agent

Figure 6.1 represents how the order-truck negotiation protocol provided in figure 5.6 is implemented in JACK ^{TM}. The inheritance relations and class import operations between any two types are carried out by connecting them with a connector line.

Figure 6.1 presents only two types of agents which are order and truck agents. As it is seen in the figure there are entities between agent types which are in the mail icon. Those are the events that are used during the negotiation protocols in JACK^{TM} IDE. There are basically two groups of event types defined in the BDI reasoning system of JACK ^{TM}. First group is *MessageEvent* types which are sent from any agent type to any other agent types defined in the system. The second group events are *BDIGoalEvents* which are sent by an agent type to itself to make its own reasoning.

As illustrated in the figure 6.1, order agent sends the *MoveRequest* event to the truck agents by using its *MoveRequesting* capability in order to find a feasible and cost effective truck agent to be assigned or consolidated. The order agent sends the

MoveRequest event to the truck agents by invoking a static method on that event type. While invoking that method, order agent transmit its attributes to the truck agent for usage in further decisions of the truck agent.

The order agent uses a list of truck agent that will be called for proposal. The list of the truck agents are provided to order agent by its respective regional load consolidation agent. The order agent calls for proposal as soon as it arrives to the MABLCS. However, it can negotiate with the truck agents which are not negotiating with other order agents at the time of its call for proposal. In order to prevent the orders to send a call for proposal to the truck agents which are negotiating with other order agents, truck agents are supplied with a *negotiation lock* that the order agents can get the lock similar to round-robin fashion. This mechanism is provided to the system in order to protect the truck agent's data atomicity. Whenever a truck agent is not negotiating then it is available for negotiation. The transport status of the truck agent does not affect this mechanism because the transportation operations are carried out in another thread during the program run-time. This property makes the truck agent to have human-like behaviors that it can change its plan while it is traveling to a destination. However the negotiation plan runs in a single thread in order to prevent logic errors. The order agents which are calling for proposal have to wait for the negotiation process ends.

When the order agent arrives to the proposed MABLCS it calls for proposal and waits for the responses from the truck agents. Order agent in figure 6.1 *waits until* it receives all the responses from the truck agent that it is in negotiation. As it is presented in the figure 6.1, some of the truck agent might refuse the call for proposal of the order agent. However the total of the received messages must be equal to the number of messages sent to truck agents which are on the list of order agent (the number of messages sent is m in the figure 5.6, therefore the total of number of rejection and the number of cost proposals must be m). There are two possible conditions for the order agent that would occur while waiting for the responses of the truck agents. The first is the condition when none of the truck agent accepts its call for proposal; the second condition is that at least one truck agent accepts its call for proposal. The reasoning mechanism of the order agent under these two conditions is as following;

Condition 1. When condition 1 occurs, truck agent behaves proactively by waiting in the system for a while and then re-calling for proposal if it's T_i^{RES} has not ended yet. Here the frequency of calling for proposal depends on the number of the orders and truck agents in the system. High frequency means a high possibility of assignment or consolidation however it increases the computational complexity of the system. The order agent adjusts the frequency of re-calling for proposal according to its remaining T_i^{RES}. If an order agent has a long remaining response time then it would have a small frequency re-calling for proposal and vice versa. Condition 1 continues as a loop until receiving any proposal from any truck agent or reaching the end of its response time. The negotiation protocol presented in figure 5.6 and implemented in JACK ™ on figure 6.1 encapsulates the re-calling for proposal plan of the order agent. Order agent runs its message sending method after waiting in a sleeping mode for a predetermined re-calling for proposal frequency. Condition 1 may result in either permanent rejection of the order agent from the system or shifting to condition 2 during looping in its reasoning.

Condition 2. Condition 2 occurs when order agent receives at least one proposal from any truck agents in the system. In this condition, order agent selects the truck agent which proposes minimum cost as presented in the figure 5.6. The order agent in figure 6.1 sends the message of *Reply* to the truck agent (now the message is sent to only single agent that has proposed the minimum cost). After the truck agent receives the *Reply* message it informs the order agent about its final decision. If the truck agent has not a schedule change in its beliefset during the time-slice between its proposal to order agent and receiving the *Reply* event, then it sends a *Grant* message to order agent. This section of the negotiation is carried out because there is a possibility of schedule change of the truck agent before the negotiating with the order agent ends. Although the probability of simultaneous occurrence of these two events is so small, the truck agent must take it into consideration in order to prevent any possible conflict. Because there is a possibility of inserting the order agent in consideration to its schedule while the schedule that the order is inserted is not the one the truck agent derived the cost of transportation and proposed to the order agent. In other words, the truck agent might have performed another operation on its schedule during its negotiation process with the order agent in consideration. In this condition there is a possibility that the order agent can not receive a *Grant* message from the truck agent. In such circumstance the order agent follows condition 1.

The most critical aspect in both of the conditions is that, the list that comprise the names of the truck agents which are called for proposal by the order agent changes dynamically due to the dynamic nature of 3PL operations. The order agent is provided with the data of positions of the truck agents by their respective regional load consolidation agent. In each loop of condition 1 order agent should update the list of the truck agents which are available for negotiation. Therefore it is a dynamic list that it might expand or shrink.

6.2 Acceptance/Rejection Reasoning

Upon getting the call for proposal from an order agent, the truck agent checks its schedule in order to decide about the newly arriving order. The first thing that the truck agent checks is the compatibility of the arriving order to its container. The compatibility means that whether the type of the order calling for proposal does fit to the type of the container the truck agent has. For instance fragile or perishable items might require some special containers to transport. Truck agent checks the compatibility of the order agent with its container via getting the order attributes with *MoveRequest* event. Therefore, there is no central checking process for both order and truck agents. The checking process completely relies on the negotiation between the multi-agents.
Figure 6.2 presents the accept/reject plan of the truck agent. Truck agent handles the *Move* event which is received from the order agent with its *AcceptReject* plan as presented in the figure 6.2.

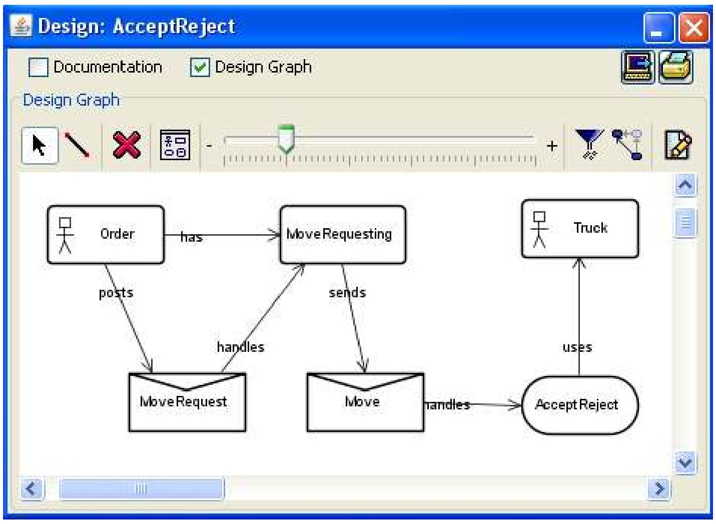


Figure 6. 2.Truck agent acceptance/rejection decision

In figure 6.3 the code of reasoning of the truck agent accept/reject plan is given. Truck agent firstly checks the compatibility of the order agent which is calling for proposal. The plan of the truck agent conduct the task of control with its **context**() method (the function of the **context**() method will be explained).

```
 public plan AcceptReject extends Plan { // start of the plan
  #handles event Move moveET;
   #posts event RejectEvent rejectEventET;
   #uses interface Truck self;
static boolean relevant(Move moveET)
  {
     return true ;
  }
   context()
  {
     (!moveET.isCompatible()| self.negotation | !self.truck.CheckFeasbility(moveET.order));
  }
  #reasoning method
  body()
  {
      @send(moveET.from, rejectEventET.rejectEventMethod(self.truck.getTruckName()));
  }
  } // end of the plan
```

Figure 6. 3. Accept/Reject Plan of truck agent

The compatibility check reduces the computational complexity of the system. This is because; the system excludes the infeasible solutions without checking its schedule and calculating cost of transportation. Checking the schedule feasibility and cost calculation requires a promising computation time therefore eliminating the infeasible orders from the context reduces the system complexity.

After checking for the compatibility, the truck agent checks the schedule feasibility when it inserts the order agent which is calling for proposal to its schedule. If the order agent that is calling for proposal is feasible for the schedule of the truck agent then it starts to calculate the cost of operation by considering all the operation elements within its schedule (*TruckData* in figure 5.3). Truck agent checks the feasibility of the proposing order agent by running its *AcceptReject* plan. This plan is the default plan of truck agent and it runs directly when the call for proposal is received from the order agent. Figure 6.3 presents the mechanism behind the tuck agents that performs the acceptance and rejection of the proposing order agent. As it is seen in the figure, there are three types of default methods of JACK^{TM} plans which are; **relevant** ( ), **context** () and **body** (). These methods are run under the agent type thread. Therefore for each truck agent created in the system, these methods are run in a new thread. This enables the designing of multi-agent system. As it can be seen, pure JAVA codes can be inserted into the default JACK^{TM} methods.

The static boolean **relevant** () method is the initial checkpoint of any plan. It is for to check of whether the plan will be run. Here, the **context** () method is used by the truck agent. Truck agent controls the negotiation attribute of the truck agent. The truck agent is locked when negotiating and changing its schedule. This makes the truck agents consistent with their decisions and provides atomicity of its data objects. If the truck agent is not negotiating with another order agent then the second element of the **context** () method is proved. Then, the truck agent checks for the feasibility of the arriving order agent by calling its **checkFeasibility** () method with the *moveET.order* parameter. This parameter holds the **7-tuple** of the order agent which is calling for proposal to the truck agent. If either one of the elements in the **context** () method of the Accept/Reject plan are true, then the *AcceptReject* plan is proved and its **body** () method works which rejects the call for proposal received from the order agent. The **body** () method is the reasoning method of the *AcceptReject* plan. Therefore the *#reasoning method* statement is added to the start of the method.

When the call for proposal of the order agent is rejected by the truck agent, the decision of the truck agent is sent to order agent by using the *@send* statement of JACK ^{TM}. The parameter inside the *@send* statement shows the address of the order agent which has called for proposal to the truck agent. *MoveET.from* parameter shows the address of the order agent to the truck agent. The **from()** method of *moveET* object returns the name of the order agent which has called for proposal. When the truck agent sends the rejection decision to the order agent, order agent evaluates this decision and it records the response of the truck agent in its beliefset.

6.3 Cost Calculation Reasoning

When the acceptance/rejection plan fails for a truck agent, in other words the order agent who is calling for proposal is feasible for the truck agent, and then the truck agent calculates the cost of the feasible schedule derived in the acceptance/rejection plan. Figure 6.4 presents the cost calculation plan of the truck agent. As can be seen both the **relevant** () and **context** () methods of the *CostCalculate* plan returns true. This is because the truck agent directly calculates the cost of the schedule which is derived in the acceptance/rejection plan of the same agent without any prerequisite.

```
public plan CostCalculate extends Plan {

  #handles event Move moveET;
  #sends event Propose proposeET;
  #uses interface Truck self;

  static boolean relevant(Move moveET)
  {
     return true;
  }
  context()
  {
     true;
  }
  #reasoning method
  body()
  {
    self.negotiation=true; // Indicates the respective agent that it is negotiating with an order
    agent
    self.calculateCost();
    @send(moveET.from, proposeET.request(cost));
   }
 }
```

Figure 6. 4. CostCalculate plan

The body method of the plan firstly makes the negotiation attribute to *true* so as to block the truck agent to other negotiations. Then it calculates the cost of the schedule derived in the acceptance/rejection plan of the truck agent. After it calculates the cost of the schedule then it sends the cost to the order agent which called for proposal. Truck agent finds the order agent with its address. Truck agent uses the statement of *@send(moveET.from, proposeET.request(cost));* in order to find the order agent which called for proposal. The negotiation attribute is unlocked in the further sections of the negotiation protocol. When the truck agent sends the *Grant* event to the order agent then the order agent makes its final decision about the assignment/consolidation. Then order agent sends the *OpenNegotiateEvent* to the truck agent in order to unlock the truck agent that it was negotiating (see figure 6.5). The **body** () method of OpenNegotiate plan of truck agent sets the negotiation attribute of the truck agent to *false*.

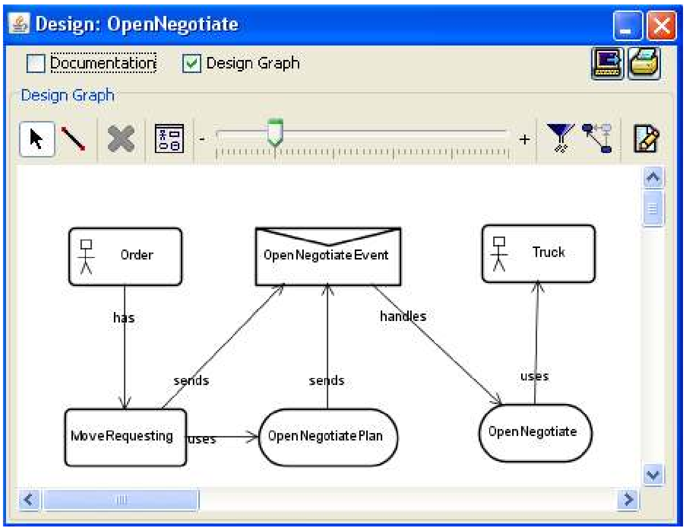


Figure 6. 5. Unlocking the truck agent after negotiation

6.4 Truck Agent Load Consolidation Mechanism

As stated in the design section of the proposed MABLCS, truck agent uses its reasoning to decide on the time of its dispatch. They select the most appropriate time to leave the position they are on. The design of the load consolidation of the truck agents are presented in section 5.3.3. Truck agent who is defined within JACK^{TM} uses a series of plans that provides them to be proactive and goal-oriented on their time of dispatch. Figure 6.6 presents the truck agent reasoning on JACK^{TM}. When truck agent finishes an operation (here it is not important the type of the operation truck agent performs, it might be whether pickup or deliver operation) then it gets the next operation element from its schedule. The next order element might be a pickup or delivery operation depending on the sequence obtained during its negotiation with the order agents. The truck agent first determines the type of the order element retrieved from its schedule. It is done in the JACK^{TM} environment by using the **relevant**() method of the agent plans. The plans first check the relevance of the next order. For instance, if the next order retrieved from the schedule of the truck agent is a pickup operation, then either of the *JustGotoPickUpPoint* or *WaitThenGoToPickUpPoint* plans works. Otherwise if the next order element retrieved from the schedule is a delivery operation then *JustGoToDeliveryPoint* or *WaitThenGoToDeliveryPoint* plans work.

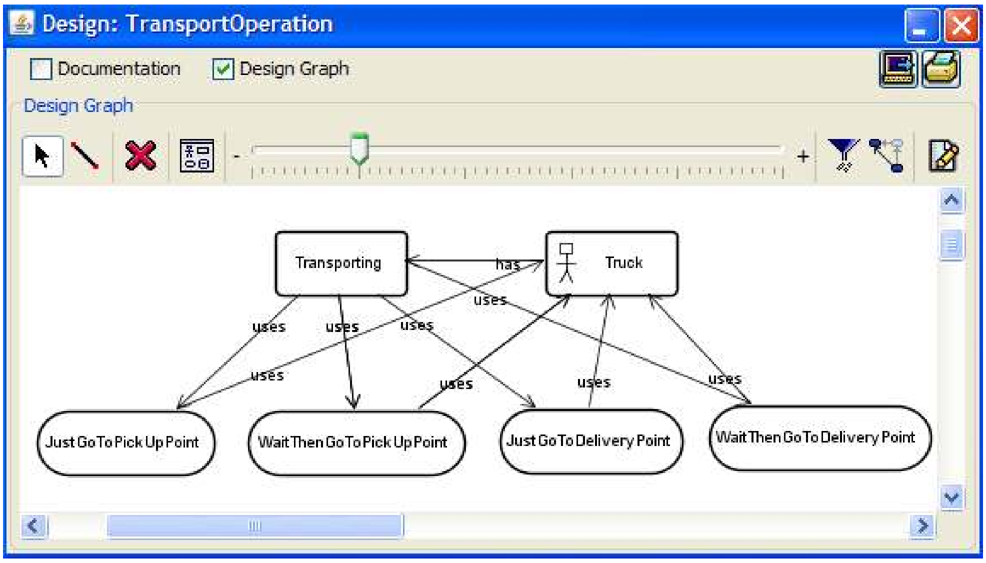

☐ Documentation ☑ Design Graph

Figure 6. 6. Truck agent load consolidation reasoning

Figure 6.7 presents the reasoning mechanism of the truck agent during its pickup operation. Before running this plan, the truck agent first checks the type of the operation whether it is a pickup or delivery operation. If it is a pickup operation the truck agent checks the context of the transportation network. It looks at the number of orders waiting at the pickup point which is on the progress of being assigned or consolidation (remember that order agents suit in the MABLCS till its assignment/consolidation or end of their response time). There is a *thresholdValue* specified in the figure 6.7. The value indicates a threshold value that is a parameter to be set before the system run. The value of the *thresholdValue* has a critical importance on the consolidation strategy. For instance, if the number of waiting order agents to be assigned or consolidated is more than or equal to a particular value then this plan is selected by the truck agent. Otherwise the truck agent stays at its previous position to get potential orders to assign/consolidate to its container. A *thresholdValue* zero means no waiting at the current position. Truck agent selects this plan whenever it has a pickup operation. The **body** () method of the *JustGoToPickUpPoint* plan makes the reasoning of taking the truck agent from its current position to the point of pickup (during simulating the proposed system, the truck agents makes their transportation operation with *@sleep* reasoning, however in the real system these data are derived from the truck geographic positioning system(GPS)). When the truck agent behaves according to this plan then it changes its network position as soon as possible and as a result it increases its chance of consolidation order at the pickup point of the next order. At the end of the transportation operation, *JustGoToPickUpPoint* plan informs the truck agent that it has arrived to the pickup point with the statement of *@post(infoET.infoMethod());*

```
public plan JustGoToPickUpPoint extends Plan {

  static boolean relevant(SendTransport sendTransportET)
    {
     return (isOrderElementType==PickUp);
    }

   context()
    {
     (numberOfOrdersOnPickupPoint › = thresholdValue);
    }

   #reasoning method
   body()
    {
     @sleep(distanceTime);
     @post(infoET.infoMethod());
    }
  }
```

Figure 6. 7. JustGoToPickUpPoint plan

Figure 6.8 present the *WaitThenGoToPickUpPoint* plan, it is the complementary of the *JustGoToPickUpPoint* plan. The difference of these two plan is at their **context** () method. This plan is applied by the truck agent if the number of orders on the pickup point is less then the *thresholdValue* which is specified also in the *JustGoToPickUpPoint* plan. If this plan is applicable by the truck agent then its **body** () method is run. The difference is that the amount of time spent at the current position.

```
public plan WaitThenGoToPickUpPoint extends Plan {

  static boolean relevant(SendTransport sendTransportET)
    {
     return (isOrderElementType==PickUp);
    }

   context()
    {
     (numberOfOrdersOnPickupPoint < thresholdValue);
    }

   #reasoning method
   body()
    {
     @sleep(timeOfCurrentPosition);
     @sleep(distanceTime);
     @post(infoET.infoMethod());
   }
  }
```

Figure 6. 8. WaitThenGoToPickUpPoint plan

The truck agent waits on its current position up to the time of latest pickup time of the next order element. Therefore, it increases its chance of consolidating the future potential orders while staying in its current position.

The other plans are run when the next operation element is a delivery operation. Truck agent again uses a *thresholdValue* to select between its plans. As it can be seen the plans are mutually exclusive and collectively exhaustive. This is because there should be an applicable plan for the truck agent to do. Whenever truck agent retrieves the next operation element from its schedule then there must be a plan which would be applicable.

6.5 Order Change/Exchange Mechanism

The order agent within the MABLCS behaves proactively to be consolidated to a proper truck which reduces the cost of transportation of the order agent. After its assignment/consolidation to a truck agent, order agent does not wait for the transportation passively. It searches for alternative load consolidations in order to reduce overall transportation cost proactively. Therefore the duty of order change/exchange is also carried out by the order agent in addition to regional load consolidation agent type.

This property of the order agent is very critical for the overall system performance. This is because, in 3PL companies, the system changes very dynamically. This functionality is provided to order agents by both regional load consolidation agent and order agent by itself. Regional load consolidation agent drives the order agents to hold a negotiation with truck agents in order to make re-assignment/re-consolidation of the orders to the trucks. The result of this negotiation is the change/exchange of the orders between the truck agent schedules. When the order agent finds a truck agent which proposes less cost than the truck agents previously proposed to it, then the order agent informs the truck agent which previously granted it to transport to exclude it from its schedule. Then order agent is assigned / consolidated to new truck which proposed better cost to order agent.

The parameter that should be adjusted for the order agent is the frequency of attempts to be re-consolidated. Of course, the frequency of attempts affects the overall system computational complexity. Order agents are directed by the regional load consolidation agents to adjust the negotiation frequency. If the searching frequency is high, then the system communication complexity also increases. Therefore regional load consolidation agent should prioritize some order agents to negotiate with the truck agent to change/exchange (remember that by changing/exchanging the goal of load consolidation is achieved). Before any pickup operation truck agent makes the reasoning of its capacity utilization. If the physical attributes of the order agent is more than the level that a candidate truck agents free capacity then the regional load consolidation agent prevent order agent to negotiate with that candidate truck agents. Therefore, regional load consolidation agent reduces the computational complexity and increases the chance of communication of the order agent with the truck agents that have the potential of accepting the order agent to their schedule. Therefore regional load consolidation agent tries to drive the order agent to consolidate with such truck agents that are also feasible for consolidation.

The implementation of the change/exchange uses the same protocol of order-truck interaction diagram which is given in the figure 6.1. If the order agent is attempting to be re-consolidated to a more advantageous truck, then it calls for proposal to the available truck agents in the system which was provided by respective regional load consolidation agent. However, this time the T_i^{RES} is set to 0. This is because while trying to be changed / exchanged, order agent has a single try in each attempt to be consolidated. Therefore only condition 2 occurs when the order agent is attempting to be changed / exchanged between truck agents. If none of the truck agents proposes to the order agent then the protocols ends for the order agent and it remains in its current schedule of truck.

While attempting to be changed / exchanged between truck agents, the order agent checks its attributes. When an order agent is picked-up by a truck agent this means for that order agent it is the end of attempts to be changed/exchanged. If the order agent continues its attempt of change/exchange although it is picked-up by a truck agent there occur some logic errors. Figure 6.9 depicts the agent types and messages types in JACK^{TM} IDE. This figure does not show the data objects of the agent types so as to reduce the messy view of the diagram.

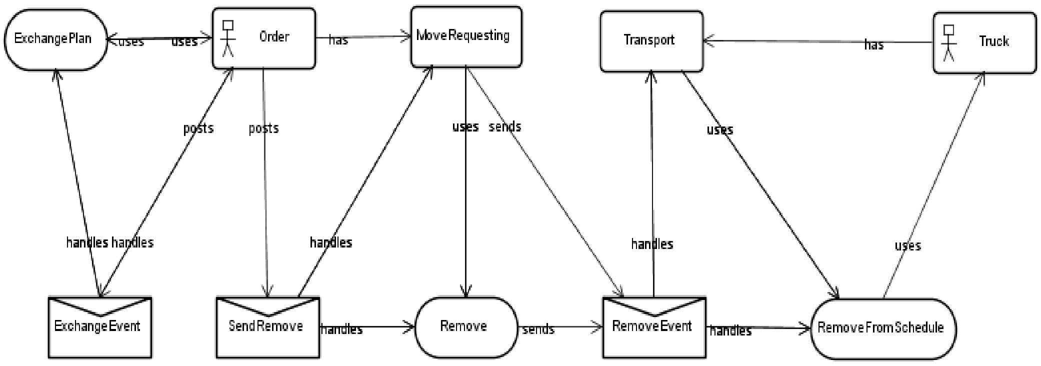


Figure 6. 9. Change/exchange between truck agents

As it is presented in the figure 6.9, the order agent has an *ExchangePlan* which runs in a different thread to trigger the order agent to attempt to be changed/exchanged. As previously stated, the triggering mechanism depends on the regional load consolidation agent. If the order agent is triggered by its *ExchangePlan* then it sends an *ExchangeEvent* to itself which runs the interaction protocol to negotiate with the available truck agents in the system. Upon the negotiation of the order agent with the truck agents if it finds a truck agent which proposes a better unit cost then it send the *SendRemove* message to the truck agent which it was previously scheduled.

When the truck agents receive the *RemoveEvent* message its *RemoveFromSchedule* plan works and it removes the order from its schedule. However, in this point there arises a new business problem. When the truck agent removes the order agent which has sent the *RemoveEvent* message, the cost of the operation of the other order agents within its schedule has changed. This is because when we remove any element from the schedule, the route and the truck capacity utilization change as well. Therefore, when any order is removed from the schedule of a truck agent, the other order agents who were previously assigned to the schedule of the respective truck agent are informed about this condition. By doing so, the truck agent which removes one of the order agent from its schedule sends *Inform* message to all of the other order agents (except the ones previously picked up, because there is no need for their cost update). The order agents then receive the *Inform* message and they update their unit cost. Updating the unit cost for the order agents also triggers their *ExchangePlans* which result in consecutive possible change/exchanges between the truck agents.

There is change/exchange possibility for the order agent up to the point of its pickup time. The order agent is informed about its attribute by the truck agent when it is picked up. Again this mechanism is provided to the system by using the messaging utility of JACK^{TM} IDE. In this system any agent can send a message to any other agent if it knows its name. Therefore when truck agent physically picks up an order it informs the order agent about its current condition. Then the order agent stops to attempt to search better load consolidation alternatives. The truck agent also informs the order agent after delivering the order.

CHAPTER 7

STATIC TEST OF THE MABLCS

In this chapter, the proposed MABLC system performance is tested with static vehicle routing problem. Although the evaluation is not adequate with the static case, there is not so much alternative to test the system performance of the proposed system. The MABLCS is used as a static problem solver in this chapter although it is designed for dynamic and stochastic logistics business conditions.

It is not possible to compare the performance of the proposed agent based system directly with other approaches as there is not an equivalent model with published results in the literature. The performance of the proposed approach is tested with some static vehicle routing problems from the literature which are solved optimally. It is clear that such a comparison is similar to comparing apples with bananas as MABLCS is not proposed an optimizing approach for static vehicle routing problems. However, the purpose in this chapter is just to see whether the MABLCS is able to follow optimal results for some static vehicle routing problems which are very special cases of the studied problems in this thesis study. The static benchmark problem sets given by Solomon et al. (Solomon, 1987) which consists of 12 test sets of 100 orders describing instances of the vehicle routing problem with time windows (VRPTW) are used for evaluation of the MABLCS (A sample set is given in APPENDIX A) . All orders are launched to present multi-agent system as arriving order agent instances. The order agents tried to be created by using the attributes given in the data set (Solomon, 1987).

Solomon has generated six different sets of problems in order to highlight several factors that affect the behavior of routing and scheduling algorithms. In this set of problems there is not a defined transportation network like the one defined for the MABLCS. However the distances between any two pickup (or delivery) points are mathematically calculated (Euclidian distance). The problem set given by Solomon

(1987) differs with respect to the width of the time windows that the truck agents must pickup the orders within that period.

The vehicle routing problem (VRP) involves the design of a set of minimum cost routes, originating and terminating at a central depot, for a fleet of vehicles which services a set of customers with known demands. Each customer is serviced exactly once and, furthermore; all the customers must be assigned to vehicles such that the vehicle capacities are not exceeded (Desrochers et al., 1992). Note that even finding a feasible solution to the VRPTW when the number of vehicles is fixed is itself an *NP-complete* problem (Gendreau et al., 1999; Savelsbergh, 1995).

In the literature there are many practical approaches that handle the classical VRP and many of its practical cases (Bodin et al., 1983; Laporte and Nobert, 1987; Magnanti, 1981). Although, there are many static scheduling algorithms in the literature proposing to solve these test problems and some of them reach a good solution of routing and scheduling, the orders within the real business environment do not always have a static form. Order requests generally arrive to system with their dynamic attributes.

7.1 Formal Definition of Static Vehicle Routing Problem with Time Windows

If we define the vehicle routing problem with time windows within the graph theoretical construct, the problem can be expressed as the following; Let $G = (V, E)$ be a complete undirected graph with vertex set $V = \{v_0, v_1, v_2, \dots, v_n\}$ and edge set $E = \{(v_i, v_j): v_i, v_j \in V, i < j\}$. In this graph, vertex v_0 is the depot and the remaining vertices are customers to be serviced. Each vertex has a time window $[e_i, \ l_i]$, where e_i and l_i are the earliest and latest service time, respectively (with e_0 , the earliest start time and l_0, the latest end time of each route). Finally, a symmetric distance matrix $D = (d_{ij})$ that satisfies the triangle inequality is defined on *E,* with travel times t_{ij} proportional to the distances.

Given a fixed size fleet of m identical vehicles, the goal is to find a set of minimum cost vehicle routes, originating from and terminating at the depot v_0, such that:

- each vehicle services one route;
- each vertex $v_i, i = 1, \ldots, n$ is visited exactly once;
- the start time of each vehicle route is greater than or equal to e_0;
- the end time of each vehicle route is less than or equal to l_0;
- the time of beginning of service b_i at each vertex $v_i, i = 1, \ldots, n$ is greater than or equal to the earliest service time e_i; if the vehicle's arrival time t_i is less than e_i, a waiting time $w_i = (e_i - t_i)$ is incurred (Gendreau et al., 1999).

7.2 Adaptation of the MABLCS to VRPTW

As a comparison problem with the proposed agent based load consolidation approach, the VRPTW is a vehicle routing problem which consist the extra complexity of time windows. In VRPTW, the pickup or delivery operations can commence at the time of the customer requests. The time window for the designed order agents which is given in the figure 4.2 directly fits to the properties of the VRPTW problem. Some of the time attributes of the order agents given in figure 4.2 fits to the *ready time* and *due date* of the VRPTW problem. Earliest pickup (τ_i^{AVL}) time is the counterpart of *ready time* of VRPTW and latest pickup time ($\tau_i^{DLN} - L_i$) is the counter part of the *due date* of the VRPTW.

The time attributes given in figure 4.2 are more detailed than the time windows of the VRPTW. It is because the start time of each vehicle route is greater than or equal to e_0; and the end time of each vehicle route is less than or equal to l_0. This means that the operation duration between depot and any customers of each particular vehicle is fixed within the duration of e_0 and l_0. However, the order agents in MABLCS have some variable delivery time that increases the complexity of the problems.

The time windows are hard constraints for the modelers. These time windows of the orders are naturally emerging in the practical business environments like the 3PL companies. Specific examples of problems with hard time windows include bank deliveries, postal deliveries, industrial refuse collection and school bus routing and scheduling (Desrochers et al., 1992).

The VRPTW is also in the scope of the truck agents of the MABLCS. This is because during making the consolidation reasoning truck agents must solve the VRPTW concurrently in each acceptance/rejection decision and during its own routing and dispatching decisions. Each truck agent could change its route after running its own VRPTW problem. The change/exchange operations are also affected by the vehicle routing problem of the truck agents. Even if when the truck agents break down they consider the VRPTW.

The customer nodes which represent the physical location of them where the trucks have to visit within the defined time window are used in order to create order agents within the proposed MABLCS. All the customer nodes given in the test problem have their own X and Y coordinates, demand of capacity utilization, time of readiness and time of due date. Therefore, the time attributes of the order agents within MABLCS are inherited from the test set data. And the weight (or volume) attribute is also inherited from the test set data. While adapting the static VRPTW to the proposed MABLCS, only one of the capacity attribute of the truck agents are considered. In other words, any one of the volume or weight capacity type is used so as to check the capacity utilization of the truck agents while negotiating. The proposed system considers both volume and weight attributes of the orders during scheduling its operations. Checking both the volume and weight attributes while accepting or rejecting an order to the schedule of a truck increases the complexity of the solution generation.

As the MABLCS is proposed for the real time load consolidation problems, the orders in the problem set are released to the system consecutively within a simulation environment. When the first order enters to the MABLCS then it directly calls for proposal to negotiate all the trucks within the system and tries to be assigned (or consolidated) to one of them. The truck agents wait for the assignment/consolidation of the orders before making any operation.

7.3 Results Obtained

Table 7.1 represents the results of the MABLCS with the heuristics solutions of Li and Lim (2003). The solutions of Li and Lim (2003) are the ones that can approximate to the optimal solutions therefore their solutions are selected as comparison.

Table 7. 1. Distances and number of vehicles used in heuristics and MABLCS

Problem	LL NV*	LL TD**	MABLCS NV	MABLCS TD
R101	19	1650.80	26	2429
R102	17	1486.12	23	2160
R103	13	1292.85	17	2138
R104	9	1013.32	14	1926
R105	14	1377.11	19	2130
R106	12	1252.03	18	1971
R107	10	1113.69	16	1938
R108	9	964.38	14	1833
R109	11	1194.73	14	1966
R110	10	1124.40	16	1938
R111	10	1099.46	14	1843
R112	9	1003.73	14	1830

*Number of vehicles used, ** Total distance traveled, LL - (Li and Lim, 2003)

Figure 7.1 presents the vehicle routing decisions of the proposed MABLCS and its comparison with the results of Li and Lim (2003).

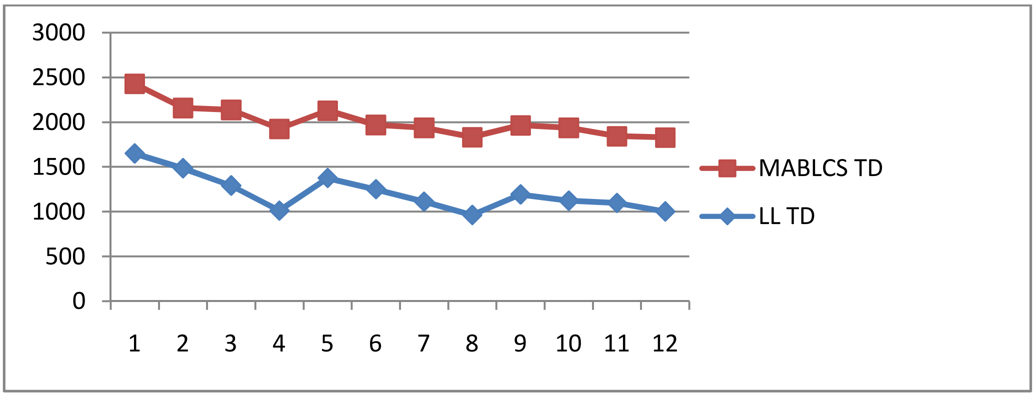


Figure 7. 1. Comparison of the MABLCS with the heuristics solutions for R101 to R112 according to distance traveled

Figure 7.2 presents the vehicle routing decisions of the proposed MABLCS and its comparison with the heuristics solutions according to number of trucks used during the transportation operations.

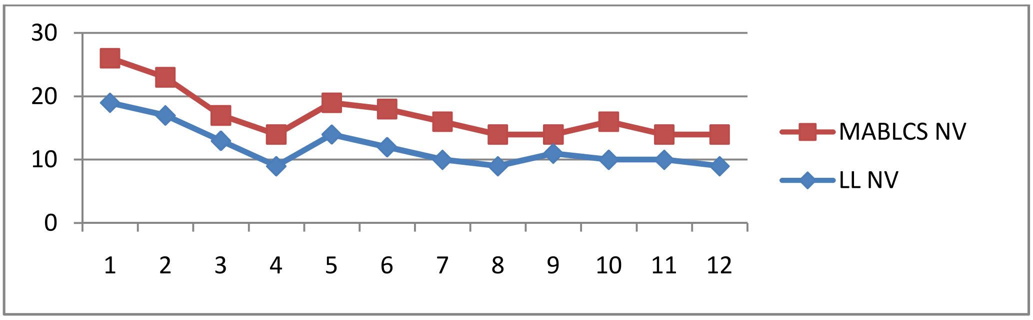


Figure 7. 2. Comparison of the MABLCS with the heuristics solutions for R101 to R112 according to number of trucks used

Figure 7.1 and figure 7.2 illustrate that the proposed multi-agent based load consolidation system can approximate to the solutions of the dedicated scheduling algorithms which are devised for solving static vehicle routing problems with time windows.

Table 7.2 presents the optimum results of the problem sets which are available in the literature with their counterpart MABLCS results and figure 7.3 and figure 7.4 illustrate the comparison of the MABLCS results with the optimum values which are available in the literature.

Table 7. 2. Distances and number of vehicles used for optimum results and MABLCS

Problem	**Optimum NV***	**Optimum TD****	**Author(s)**	**MABLCS NV**	**MABLCS TD**
R101	20	1637.7	KDMSS	26	2429
R102	18	1466.6	KDMSS	23	2160
R103	14	1208.7	CR+L	17	2138
R104	11	971.5	IV	14	1926
R105	15	1355.3	KDMSS	19	2130
R106	13	1234.6	CR	18	1971
R107	11	1064.6	CR	16	1938
R109	13	1146.9	CR	14	1966
R110	12	1068	CR	16	1938
R111	12	1048.7	CR	14	1843

KDMSS -(Kohl et al., 1999), CR -(Cook and Rich, 1999), L - (Larsen, 1999), VI - (Irnich and Villeneuve, 2003)

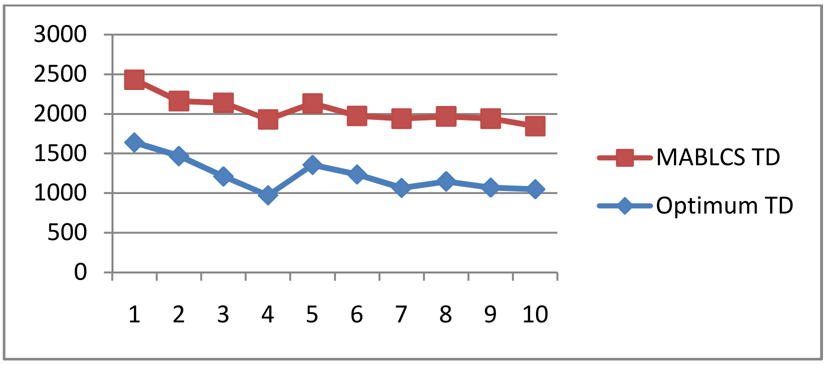


Figure 7. 3. Comparison of the MABLCS with the optimum solutions for R101 to R112 according to distance traveled

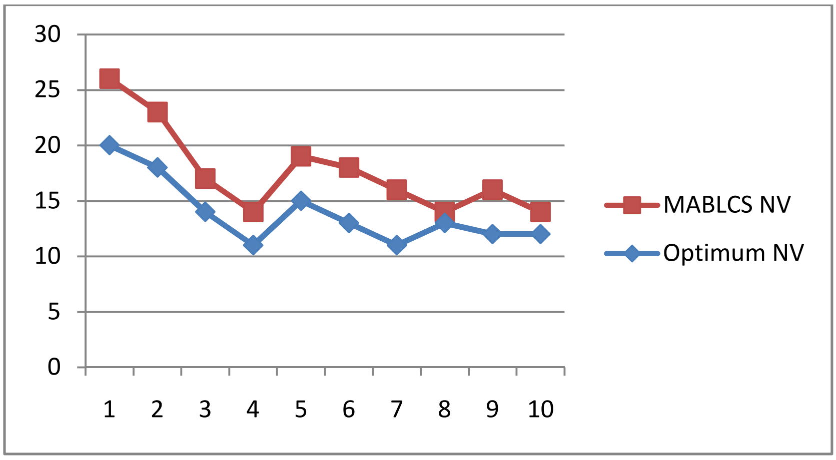


Figure 7. 4. Comparison of the MABLCS with the optimum solutions for R101 to R112 according to number of trucks used

The solutions obtained by the MABLCS shows that the average number of orders assigned/consolidated to a single truck is approximately 6. This average is important for the developed system. This is because the average order that can be consolidated to the schedule of a truck agent in 3PL operations is approximately 4 (this data is obtained by interviewing with the 3PL company managers). This shows that the scheduling capability of the truck agents is adequate to find a good solution of scheduling. In addition to the average number of orders assigned to a truck in a 3PL company, the container capacity of a standard truck agent is approximately 20 tones and 98 meter cube. This is another indicator for the truck agents that the scheduling capabilities of the truck agents are adequate. Because as the capacity of the container increases, the number of orders that might be picked up consecutively increases respectively. Of course, the time window intervals of the orders have a considerable effect on the solution. The vehicle capacities in the problem set are higher than the standard container capacities used in practical 3PL operations.

Another critical point that might be emphasized is the distributed nature of the problem solving approach of the proposed MABLCS. While getting the results of the problem set of R101 to R112 a computer with a single processor is used. However, the truck agents of the MABLCS can be employed on multiple computers on a network that have multiple processors. Therefore, the truck agents can use the brute force of their own computing. The best results obtained in the previous studies show that at most 13 orders are consolidated within a single container, which means that the truck agents can reach an optimum solution within a 13! trials in their own computing environment. As we can ignore the time of negotiation of the orders with the truck agents, the optimum solution can be obtained within a MABLCS when the agents are employed on different computers.

CHAPTER 8

EXPERIMENTAL CASE STUDY

In this chapter the MABLCS is tested with the real business data of a 3PL company with the title of *experimental case study*. The operational data such as average weight or volume attributes of the orders, distance between transportation network nodes, the direct and indirect costs are derived from the real business data. However the stochastic behavior is supplemented to the system in addition to the real business data. The simulation parameters are inserted to the system by using an integrated development environment (GUI) and output results are collected in some predefined data structures.

There are not so much simulation studies of any 3PL system operations in the literature. The framework proposed by Regan et al. (1998) is the one of the rare dynamic fleet management system frameworks (Regan et al., 1998). In their studies Regan et al. (1998) proposed some load acceptance and assignment strategies. They presented the application of the simulated framework to the investigation of the performance of a family of real-time fleet operational strategies, which include load acceptance, assignment, and reassignment. However in their studies the fleet management is performed for TL operations. The proposed MABLCS is designed and implemented for both TL and LTL operations.

The proposed MABLCS is developed in order to solve real-time load consolidation problems and support dispatch officers during their operations decisions. The proposed MABLCS can adapt to changing business environment by reacting to environmental changes that the orders and truck agents are in. Therefore the real performance of the proposed system can only be evaluated within the real business environment. However the performance of the MABLCS can be tested by comparing its performance with running the system with and without some of its functionalities. The proposed system's performance can be evaluated by using some various scenarios that include some different system parameters. Therefore, the MABLCS is evaluated by using some problem set with some agent reasoning capabilities rather than full implementation of the MABLCS in the real business.

Load consolidation studies in the literature generally estimate the orders which will be consolidated that has the same pickup and delivery points and same time windows. Therefore, the classical load consolidation methods cannot be compared directly with the proposed agent based load consolidation system. The proposed system considers both the time windows of the orders arriving to system and the capacity limitations of the containers. In addition to that, the proposed agent-based load consolidation system considers the business regulations and environmental changes. Therefore there is not a direct performance evaluation alternative. The proposed simulation setup is the abstraction of the real business operating system with MABLCS.

The MABLCS pursues the load consolidation objective while conducting the following operations which are distributed to agent elements by utilizing negotiation and BDI reasoning capabilities of the agent types.

- **Load acceptance/rejection**
- **Load assignment**
- **Re-assignment**
- **Routing**
- **Scheduling**

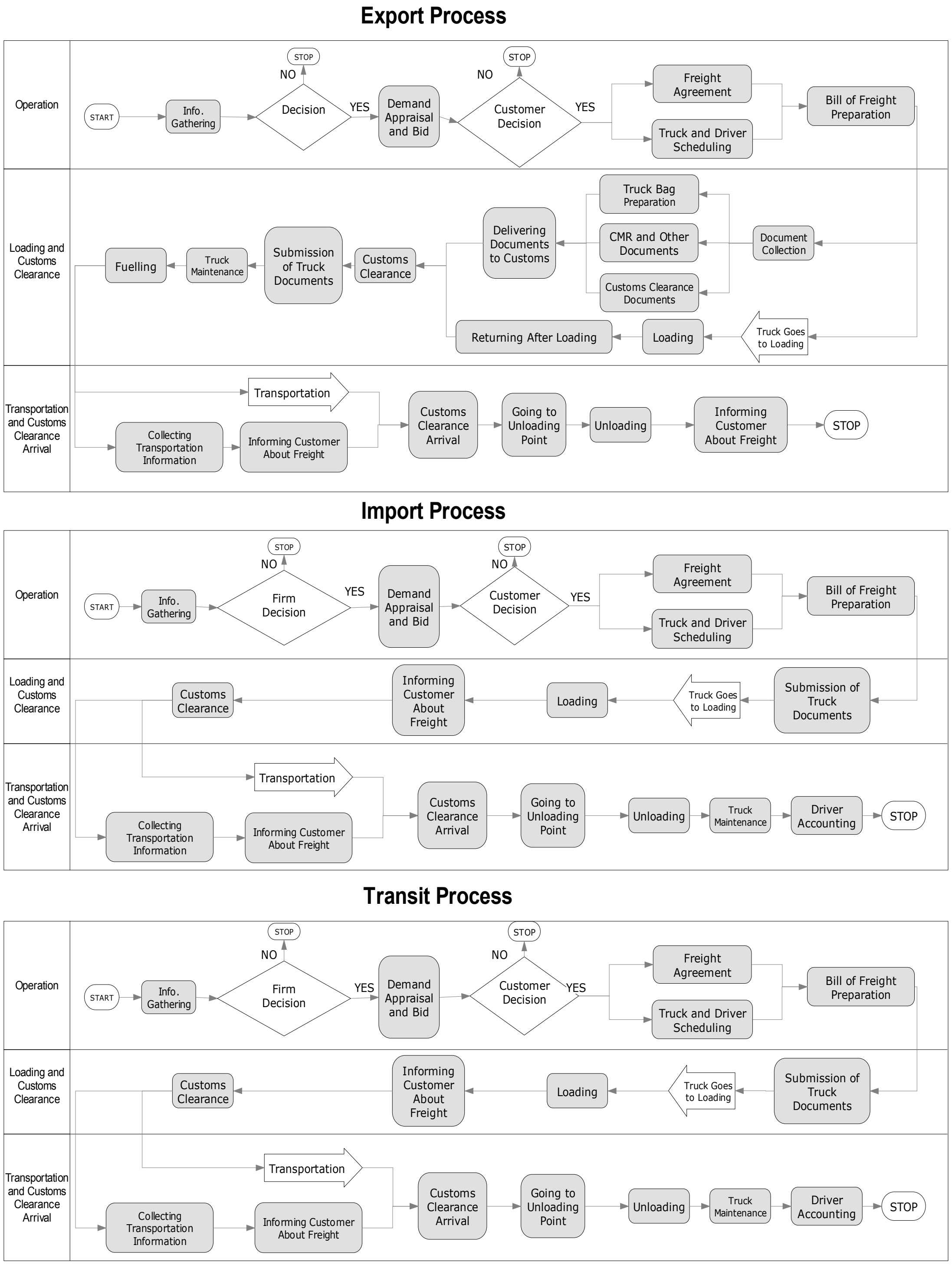
In order to show the load consolidation performance of the proposed MABLCS a set of experiments are conducted with the ***real business data*** of a 3PL company. The data of the 3PL company is used in the experiment to show the direct contribution of the proposed system to real 3PL companies. The contribution of the proposed MABLCS is tested with different experiment setups. The experiment setups are designed for the cases where some of the system functionalities of the MABLCS are included or not.

As stated previously, the operational data such as transportation durations and transportation costs that will be used in the proposed MABLCS are obtained from the real 3PL operations. The usage of the real business data provide us to take the real system parameters. This enables the truck agents make their own reasoning according to their real operations data. The container capacities and the transportation network that will be used in the experiment are also derived from the real business data. In the following sections the introduction of the 3PL company where the case study is performed and the process of obtaining true operations cost (with a well know costing method) are given.

8.1 Introduction to the Company and Case Study

The company where the experiment of the MABLCS will be conducted is located in south eastern part of Turkey. The main services of the company where the experimental case study is performed consist of export services from Gaziantep to European countries and import from European countries to Turkey. Transit services are also provided by the company. The company is established at 1936. The company presented a sharp growth rate after year 2000. The company owns over 200 trucks presently. The company is presently one of the biggest logistics company (land transportation service provider) in the south east of Turkey. Company’s main operations consist of planning (truck assignment and organization, route planning, load consolidation etc.), customer relations/marketing, land transportation (import, export and transit), warehousing, accounting, maintenance and support services (Baykasoğlu and Bartık, 2005; Baykasoğlu et al., 2007).

The company in consideration has a plan to start for LTL transportation services (during the time period of this thesis study LTL operations are being conducted rarely). The transportation operations are being conducted as the *export*, *import* and *transit* service types. The business processes are constructed according to these transportation types. Figure 8.1 represent the business process model of the company which is constructed according to the mentioned transportation service types. However within the proposed system each LTL operation is a unique operation. Therefore the business process map of the company given in the figure 8.1 might be updated when the proposed system is adapted to the company. The transportation network of the company might be divided to some regions and the processes might be organized according to their regional attributes.


Export Process

Import Process

Transit Process

Figure 8. 1. Process map of the logistics company (Baykasoğlu and Kaplanoğlu, 2007)

The company management has a strategy of dedicating some of their vehicles to LTL operations. This is because the company has some scheduled TL operations and it has some extra vehicles in its fleet for LTL operations.

In the proposed experimental study the direct and indirect costs that will be used in truck operations are obtained from the real business data. The direct cost and indirect cost expressed in the equation 5.1 uses the real cost data of the company. The accuracy of the approximation of the direct and indirect costs is tested by comparing the cost approximations with the historical true costs of the transportation services.

The true costing are tried to be obtained by using activity based costing (ABC) (Baykasoğlu and Kaplanoğlu, 2008). The transportation service costs of the 3PL company in consideration are also derived by using traditional costing method. The costs obtained by using ABC are also compared with the costs that are obtained by using the traditional costing method to see the accuracy of the costs used in the proposed system.

8.2 Cost of the Transportation Services with Activity Based Costing

Within the context of this study, the costs of the transportation services of the company are tried to be determined for a nine-month time period. 28 different services including both exports and imports that are performed within the time period of nine-months are identified. There are 28 different transportation operations 14 of which is export and 14 of which for import.

Some notes on activity based costing;

ABC has been revealed recently and used rarely by the service sectors especially by the logistics sector. ABC has appeared during the 1980s' with the studies of Cooper (Cooper, 1988a; Cooper, 1988b), Cooper and Kaplan (1988) and Johnson and Kaplan (1987) (Cooper and Kaplan, 1988; Johnson and Kaplan, 1987). Cost calculation of the products and/or services in traditional costing is based on the determination of direct costs and indirect costs and then summing them to find the individual cost of each element. Traditional costing involves collecting indirect costs from accounting departments and then allocates them to products or services (Tsai and Kuo, 2004). The overheads distribution to the products and/or services is performed by a single-volume cost driver and there is generally only one stage for allocation of the overheads to the cost objects. Using single-volume cost driver in order to allocate indirect costs to the cost object might not be a sufficient method for a detailed cost analysis in many circumstances. Direct labor or raw material usages are frequently considered as a cost driver in traditional costing and a single cost driver is generally used for the distribution of overheads. In addition to this, traditional cost accounting (TCA) may lead some cost distortions due to some lack of cost calculation. There is a consensus about distortion of product costs when the accounting is performed with TCA especially for the organizations where the proportion of overheads to total costs is fairly high (Gunasekaran and Sarhadi, 1998; Tsai and Kuo, 2004). On the other hand, the main premise behind ABC is to classify overheads or indirect costs and to allocate them to end products or services based upon the activities required to produce these products (Raz and Elnathan, 1999). The allocation of the indirect costs to product and/or services differs from the TCA. ABC assumes that cost objects (products, product lines, processes, customers, channels, markets and so on) create the need for activities, and activities create the need for the resources (Tsai and Kuo, 2004). The accuracy of ABC can vary according to its focus. The focus might be on the product, customer, or a combination of both. Resources include indirect costs of the organizations and they are allocated to the activity centers. Resource drivers are used during allocation of the resources to the activity centers (see figure 8.2).

Like many other traditional costing approaches ABC also makes backward-looking in order to support forward decision making. However, in some cases there might be disagreement over which costs to be included in an analysis, especially where fixed costs are involved. The usage of multiple cost drivers in ABC brings the advantage of detailed cost estimation; on the other hand selection of the proper cost driver is a challenge for a good ABC analysis. Another difficulty of ABC analysis is the decision which must be made about the costs types. Direct costs and indirect costs should be determined precisely.

Resource driver is an allocation rate of an individual resource and shows the resource consumption levels of the activities. This procedure comprises the first stage of ABC. Figure 8.2 presents the stages of ABC.

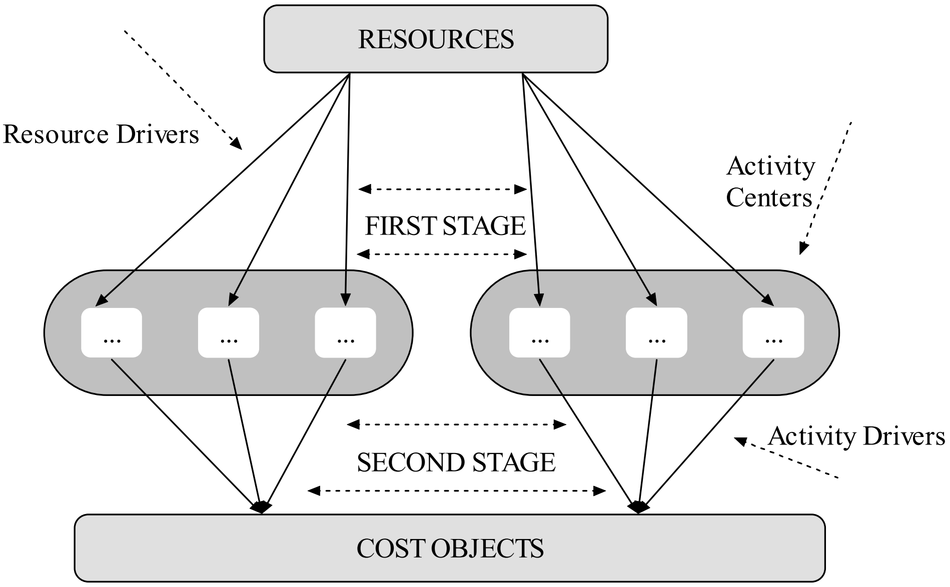


Figure 8. 2. Cost assignment procedure in ABC (Tsai and Kuo, 2004)

After finding the costs of the activities (cost pools), ABC plans to distribute them to cost objects. Some drivers are used for the allocation of costs of the activities to the cost objects in a similar manner of first stage cost allocation. The meaning of “driver” in the second stage of ABC has the same meaning with the first stage cost driver of ABC. Cost objects are loaded by the activity cost pools by the predetermined second-stage cost drivers. As a result of activity cost pool allocation to the cost objects, cost consumptions of each cost objects are found. The unit cost of each cost object is then found by dividing the total allocated cost by the product amount.

As the company in consideration was providing TL transportation services at the determined time period of cost calculation then the unit costs are calculated for each TL operations. As a result the cost calculation is made so as to find standard costs of each route (it is whether export or import). In other words the cost objects of this case study are the truckloads of the logistics company. Belgium Export, Belgium Import, Germany Export etc. are some examples of the cost objects. In order to find the cost of operations of the proposed 3PL company, direct cost of each transportation for the nine-month duration had been recorded.

The transportation services provided by the company and their respective data for the predetermined time period are given in table 8.1.

Table 8. 1. Transportation services and their data

No.	Transportation Services	Total Number of Transportations	Total Amount of Freight	Transportation Duration (days)	Direct Cost ($)
1	Belgium Exports	49	825	375	125,649
2	Belgium Imports	119	2,130	1,366	239,911
3	England Exports	19	283	195	52,490
4	England Imports	11	164	132	26,818
5	France Exports	107	2,253	1,072	194,906
6	France Imports	92	1,036	1,054	182,051
7	Germany Exports	73	1,208	644	178,966
8	Germany Imports	114	1,731	1,192	253,070
9	Greece Exports	36	639	178	40,273
10	Greece Imports	46	1,027	204	42,905
11	Holland Exports	7	118	64	15,012
12	Holland Imports	5	67	46	5,086
13	Iran Imports	10	147	63	9,811
14	Ireland Exports	2	28	19	2,646
15	Italy Exports	12	256	113	21,137
16	Italy Imports	11	177	113	11,392
17	Norway Exports	12	24	112	34,790
18	Norway Imports	35	690	819	101,593
19	Poland Exports	107	2,196	923	249,292
20	Poland Imports	23	382	164	48,078
21	Russia Exports	66	1,258	864	78,677
22	Russia Imports	31	628	441	35,345
23	Spain Exports	2	49	21	2,965
24	Spain Imports	1	17	11	1,473
25	Sweden Exports	4	43	49	8,232
26	Sweden Imports	38	582	700	95,017
27	Ukraine Exports	13	252	96	26,174
28	Ukraine Imports	3	56	31	2,953
	Totals	1,048	18,266	11,061	2,086,709

In TL operations of the 3PL company, transport services are grouped according to their direction such as export or import as the depot in Turkey is assumed as the central depot. However in the proposed MABLCS the points that the 3PL company serves are considered as both an origin and destination point. The pickup and delivery points of the company are considered as the node of the transportation network of the company. Therefore there is not a conceptual central depot in the proposed MABLCS.

In order to calculate the cost of accepting an order to the schedule, the truck agents calculate the cost of their operations which is stated in the equation 5.1. The unit indirect costs (*uic*) stated in the cost calculation section of the agent design uses the data of the indirect costs occurring during the transportation services of the 3PL company. In order to find the *uic* for the same amount of time, the overheads of the company is compiled from the accounting service of the company. Table 8.2 presents the overheads during the same time period of nine months.

Table 8. 2. Overheads

Overheads	Amount ($)	Overheads	Amount ($)
1- Vehicle Depreciation Costs	1,144,008	**13-** Warehouse Costs	18,092
2- Employees Insurance Costs	107,526	**14-** Building Electricity Costs	
3- Indirect Labor		Building Water Consumption	
Staff Training	121,203	Building Cleaning Expenses	9,538
4- Withholding Tax		**15-** Personnel Transportation Service Costs	
Return of Tax	50,523	Urban Transport of Staff Costs	
5- Motor Vehicle Tax		Urban Transport Fuel Consumption	
Vehicle Insurance		Other Fuel Consumptions	30,053
Vehicle License Costs		**16-** Aero plane Ticket Expenses	
Vehicle Traffic Control Costs		Foreign Travel Expenses	5,902
Vehicle Maintenance	311,866	**17-** Conveyance Lawyer Costs	
6- Tax of Building		Consultancy Costs	
Insurance of Building	10,007	Other Counseling Costs	
7- Truck Driver License Costs	72,516	Banking Costs	191,676
8- Replacement Part of Vehicles Costs		**18-** Advertising	
Tire Costs	63,213	Documents Expenses	
9- Customs Costs		Stationery Costs	
Tickets bought during transportations	17,863	Newspaper Expenses	
10- Telephone Bills	24,504	Computer Maintenance Costs	32,370
11- Refectory Expenses	24,150	**19-** Donations	
12- Representation Expenses		Other Costs	
Car Park Expenses		Motoring Fine Costs	80,257
Mailing Expenses			
Photocopy Costs	5,621	**Total:**	2,320,889

The overheads and direct costs given in the table 8.1 are used in order to find the cost of operations via ABC. Below in table 8.3 the ABC cost results of the transportation services are given.

Table 8. 3. ABC cost result

No.	Transportation Services	Total Overheads ($)	Direct Costs ($)	Total Costs ($)	Total Number of Transportation	Unit Costs ($)
1	Belgium Exports	80341	125649	205990	49	4204
2	Belgium Imports	416166	239911	656077	119	5513
3	England Exports	18668	52490	71158	19	3745
4	England Imports	9552	26818	36370	11	3306
5	France Exports	334996	194906	529902	107	4952
6	France Imports	176450	182051	358501	92	3897
7	Germany Exports	156172	178966	335137	73	4591
8	Germany Imports	311405	253070	564475	114	4952
9	Greece Exports	31713	40273	71986	36	2000
10	Greece Imports	50259	42905	93164	46	2025
11	Holland Exports	4998	15012	20010	7	2859
12	Holland Imports	3242	5086	8328	5	1666
13	Iran Imports	6869	9811	16680	10	1668
14	Ireland Exports	1277	2646	3923	2	1961
15	Italy Exports	9626	21137	30763	12	2564
16	Italy Imports	8037	11392	19429	11	1766
17	Norway Exports	7593	34790	42383	12	3532
18	Norway Imports	71489	101593	173081	35	4945
19	Poland Exports	337504	249292	586796	107	5484
20	Poland Imports	23331	48078	71409	23	3105
21	Russia Exports	138802	78677	217478	66	3295
22	Russia Imports	39802	35345	75147	31	2424
23	Spain Exports	1323	2965	4288	2	2144
24	Spain Imports	620	1473	2093	1	2093
25	Sweden Exports	2916	8232	11148	4	2787
26	Sweden Imports	65256	95017	160272	38	4218
27	Ukraine Exports	10548	26174	36721	13	2825
28	Ukraine Imports	1937	2953	4889	3	1630

8.3 Cost of the Transportation Services with Traditional Cost Accounting

In order to compare the costs obtained by using equation 5.1, the costs of the transportation services are also derived by using traditional costing method. In TCA the costs of the objects are calculated by calculating the direct and indirect costs separately.

Direct cost determination

In this case study, direct cost for each service is found by dividing the total direct cost of services to the number of service given during the nine-month time period. For example, total direct cost used for Belgium exports is $125,649. Unit based direct cost of this transportation services is found as; ($125,649) / (49 transportation) = $ 2,564 for each transportation. Direct costs of other services are found in a similar manner.

Indirect cost determination

With the standard and traditional costing methods indirect cost of each cost object is derived by a singe volume cost driver (Helberg et al., 1994; Nachtmann and Al-Rifai, 2004; Tsai, 1998). As Gupta and Galloway (2003) stated, traditional cost accounting uses single cost driver (direct labor or machine hours) as the basis for allocating overheads in manufacturing organizations but in service organizations like logistics, the traditional cost accounting drivers does not work properly (Gupta and Galloway, 2003). Logistic operations does not include a direct labor hours for its services or any type of raw materials. Therefore, cost driver for this case study is determined by the accounting staff as the "*number of transportation*". For the time interval of nine-months 1,048 transportation services are given for both export and import. By using TCA, the indirect costs are allocated to each 28 different services via the total number of transportation. Unit based indirect costs are calculated by the following equation;

Overhead allocated to each unit of service = Total Overhead / Selected Cost Driver (8. 1)

The total overheads of the company are \$ 2,320,889 for the predefined time period of 9 months. With the help of equation 8.1, the average overhead for each service can be calculated as; (\$2,320,889) / (1,048 transportations) = \$2,215 / transportation.

After determining the direct and indirect costs of each service, total costs are found by summing up the allocated indirect cost and average direct cost of each cost object. Table 8.4 presents the costs of each cost object with its corresponding cost data.

Table 8. 4. Traditional costing results of transportation services

No.	Transportation Services	Direct Costs ($)	Total Number of Transportation	Unit Direct Costs ($)	Indirect Cost ($)	Total Cost ($)
1	Belgium Exports	125649	49	2564	2215	4779
2	Belgium Imports	239911	119	2016	2215	4231
3	England Exports	52490	19	2763	2215	4977
4	England Imports	26818	11	2438	2215	4653
5	France Exports	194906	107	1822	2215	4036
6	France Imports	182051	92	1979	2215	4193
7	Germany Exports	178966	73	2452	2215	4666
8	Germany Imports	253070	114	2220	2215	4435
9	Greece Exports	40273	36	1119	2215	3333
10	Greece Imports	42905	46	933	2215	3147
11	Holland Exports	15012	7	2145	2215	4359
12	Holland Imports	5086	5	1017	2215	3232
13	Iran Imports	9811	10	981	2215	3196
14	Ireland Exports	2646	2	1323	2215	3538
15	Italy Exports	21137	12	1761	2215	3976
16	Italy Imports	11392	11	1036	2215	3250
17	Norway Exports	34790	12	2899	2215	5114
18	Norway Imports	101593	35	2903	2215	5117
19	Poland Exports	249292	107	2330	2215	4544
20	Poland Imports	48078	23	2090	2215	4305
21	Russia Exports	78677	66	1192	2215	3407
22	Russia Imports	35345	31	1140	2215	3355
23	Spain Exports	2965	2	1483	2215	3697
24	Spain Imports	1473	1	1473	2215	3687
25	Sweden Exports	8232	4	2058	2215	4273
26	Sweden Imports	95017	38	2500	2215	4715
27	Ukraine Exports	26174	13	2013	2215	4228
28	Ukraine Imports	2953	3	984	2215	3199

8.4 Cost Estimation and Cost Comparison

After going through all the activity based costing and traditional costing phases the true costs of the operations are obtained. Table 8.5 presents the costs of the transportation services both with activity based costing and traditional costing method. Table 8.5 also presents the difference between the costing methods and the tariff the company have for its transportation operations. As it can be seen two costing methods are somehow different in their results.

Table 8. 5. ABC and TCA comparison

No.	Transportation Services	ABC Costs ($)	Traditional Costs ($)	Difference Percent (%)	Transportation Price ($)	Profit / Loss (Traditional) ($)	Profit / Loss (ABC) ($)
1	Belgium Exports	4204	4779	13.7	4592	-187	388
2	Belgium Imports	5513	4231	-23.3	4373	143	-1140
3	England Exports	3745	4977	32.9	5895	918	2150
4	England Imports	3306	4653	40.7	4784	131	1478
5	France Exports	4952	4036	-18.5	3455	-582	-1498
6	France Imports	3897	4193	7.6	4623	429	726
7	Germany Exports	4591	4666	1.6	3678	-988	-913
8	Germany Imports	4952	4435	-10.4	4372	-62	-579
9	Greece Exports	2000	3333	66.7	2177	-1157	177
10	Greece Imports	2025	3147	55.4	1297	-1851	-729
11	Holland Exports	2859	4359	52.5	6429	2069	3570
12	Holland Imports	1666	3232	94.0	4514	1282	2848
13	Iran Imports	1668	3196	91.6	1165	-2031	-503
14	Ireland Exports	1961	3538	80.4	7800	4262	5839
15	Italy Exports	2564	3976	55.1	3289	-687	725
16	Italy Imports	1766	3250	84.0	5530	2280	3764
17	Norway Exports	3532	5114	44.8	4340	-774	808
18	Norway Imports	4945	5117	3.5	9805	4687	4859
19	Poland Exports	5484	4544	-17.1	3402	-1142	-2082
20	Poland Imports	3105	4305	38.7	2519	-1786	-586
21	Russia Exports	3295	3407	3.4	5843	2437	2548
22	Russia Imports	2424	3355	38.4	1698	-1657	-726
23	Spain Exports	2144	3697	72.4	4104	406	1959
24	Spain Imports	2093	3687	76.2	4224	537	2131
25	Sweden Exports	2787	4273	53.3	4526	253	1739
26	Sweden Imports	4218	4715	11.8	9382	4667	5165
27	Ukraine Exports	2825	4228	49.7	5231	1003	2406
28	Ukraine Imports	1630	3199	96.3	1605	-1594	-25

The truck agents that will provide both TL and LTL operations would use the direct costs and indirect costs which are given in the table 8.1 and table 8.2 respectively. As discussed in the section 5.3.2 the cost of the LTL operations are proposed to the order agent by calculating the unit transportation cost for each *distance x weight.* In their studies Regan et. al. (1998) also consider the loaded kilometers of the trucks however in their studies the cost of the transportation is proposed for the TL transportation (Regan et al., 1998).

In general transportation operations the revenue earned from the transportation business is proportional to the length of the distance of origin and destination of the respective order. The costing technique used in the proposed MABLCS is similar to the studies of Regan et. al. (1998). However there is not a fixed tariff for each transportation operation in the LTL operations as in the TL operations have. Therefore the tariff of the LTL operations will be the approximation to the TL operations. The approximation of the LTL tariff would be higher than the TL tariff. This is because there is not a standard way of calculating operation cost for the LTL operations. The profit model for TL operations applied in their study Regan et. al. (1998) assumes a lower value of $0.72 per loaded kilometer when it is compared with LTL operations because both revenues and costs, particularly fixed costs and overhead, in truckload trucking are lower than in less-than-truckload (LTL) or mixed operations (Regan et al., 1998). Therefore we can say that the tariff applied to the LTL operations is higher when we compare it with the TL operations.

The cost calculation function given in equation 5.1 could be applied for the real-time cost calculations. However the direct cost term in the equation is obtained by calculating the unit direct costs occurred in the system. As the total direct costs occurred and total distance traveled by the trucks are known from the 9-month historical data of the 3PL company then the unit direct cost for each *km* can be obtained for the 3PL company in consideration. In total, trucks have operated for 3,280,436 kilometers. During this period $2,086,709 cost has occurred directly. Therefore 2,086,709/3,280,436 = ***0.64 $*** occurs for each kilometers.

The unit direct cost of trucks is calculated as 0.64 $/km. This value would be used as an approximation for the direct cost of the truck operations. When the truck agents operate on transportation then a cost of 0.64 $/km occurs.

The unit indirect cost of the transportation is also approximated via same manner. The only difference is that unit indirect cost is calculated according to the distances and tones transported. When the total amount of overheads is divided by the total km x tones of the truck operations then the unit indirect cost is obtained.

In total 57,331,737 km x ton is carried by the 3PL company during the 9-month time period. During that period $ 2,320,889 overhead occurred. Therefore for each km x ton a cost of 2,320,889 / 57,331,737 = ***$ 0.04048*** occurs. In the APPENDIX B the parameter set of the direct and indirect costs to the truck agent are presented.

The transportation services within the 9-month time period are TL operations. The average weight utilization of the trucks are 20 tones. Therefore running each truck when it is fully loaded is 0.04048 x 20 = 0.81 $ for each container for each kilometer. By summing the direct cost and indirect cost in total $1.45 occurs to move each container for one kilometer. The unit cost found is used as an alternative costing method. The cost found for each transportation operation by using the approximated unit direct cost and unit indirect cost are given in table 8.6 together with the ABC and TCA results. In many of the transportation services the approximated cost is so close to the costs obtained by ABC and some of them the approximated cost are between the ABC and TCA cost values.

Table 8. 6. Cost approximation

Transportation Service	Trans. Dur.	ABC	Cost Approximation.	TCA	Price ($)
Belgium Exports	6	4204	4337	4779	4592
Belgium Imports	7	5513	5349	4231	4373
England Exports	7	3745	5060	4977	5895
England Imports	7	3306	5060	4653	4784
France Exports	8	4952	5851	4036	3455
France Imports	7	3897	5060	4193	4623
Germany Exports	6	4591	4626	4666	3678
Germany Imports	6	4952	4482	4434	4372
Greece Exports	5	2000	2918	3333	2177
Greece Imports	5	2025	2918	3147	1297
Holland Exports	6	2859	3759	4359	6429
Holland Imports	4	1666	2169	3232	4514
Iran Imports	4	1668	2169	3196	1165
Ireland Exports	7	1961	5494	3538	7800
Italy Exports	5	2564	3181	3976	3289
Italy Imports	4	1766	2169	3250	5530
Norway Exports	5	3532	2891	5114	4340
Norway Imports	8	4945	6672	5117	9805
Poland Exports	7	5484	4809	4544	3402
Poland Imports	6	3105	3788	4305	2519
Russia Exports	5	3295	3614	3407	5843
Russia Imports	5	2424	2602	3355	1698
Spain Exports	5	2144	2747	3697	4104
Spain Imports	5	2093	2602	3687	4224
Sweden Exports	5	2787	3614	4273	4526
Sweden Imports	6	4218	4337	4715	9382
Ukraine Exports	6	2825	3655	4228	5231
Ukraine Imports	4	1630	2169	3199	1605

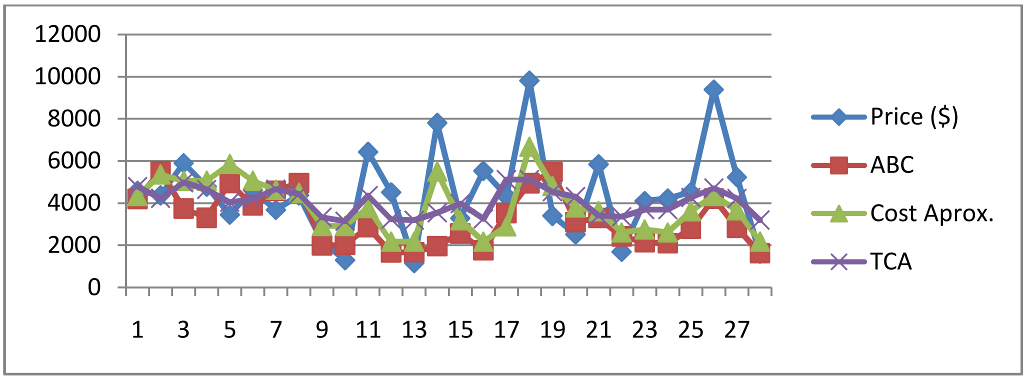


Figure 8. 3. Approximated costs with ABC, TCA and Prices

Table 8.6 and figure 8.3 illustrates that the usage of unit direct cost and unit indirect cost for LTL in MABLCS sounds reasonable.

8.5 Transportation Network

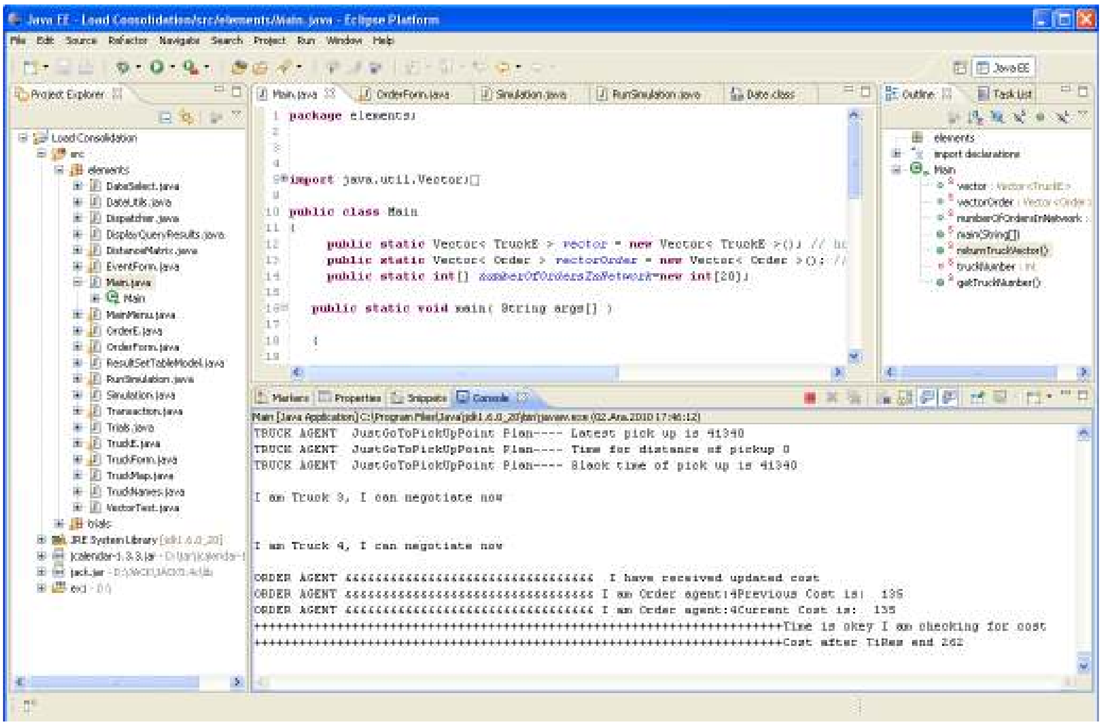
Although the number of nodes is extensible in the proposed MABLCS by just inserting the node data, the experiment is done for the 3PL transportation network where 16 different nodes exist (see table 8.1 for transportation service types). It is assumed that a transportation operation can be carried out between any two nodes, therefore there are 120 different arcs connecting these 16 nodes (which includes the main node located in Turkey). The distances between any nodes are assumed as symmetrical. The traveling times between these nodes are known exactly (the speed of the truck on each road is known). The 3PL operations are done on the international roads therefore it is assumed that the traveling durations between any two nodes is known exactly. The distance matrix representing the distances between each node could be found in the APPENDIX C. Figure 8.4 presents the transportation network and the nodes where the order agents emerges and truck agents operates.

Figure 8. 4. Transportation network and nodes of the network

The LTL operations are assumed to be carried out between any two nodes on the entire transportation network. This means that there might be an order request to be transported in each node on the transportation network.

8.6 Simulation Environment

The simulation study proposed in this chapter is directly coded in JACK^{TM} and JAVA SE 6 runtime environments. The reporting functionalities of the simulation environment are adapted to system with JAVA SE 6 runtime environment. The BDI reasoning capabilities are coded in JACK^{TM} environment therefore JACK^{TM} environment is run as a BDI reasoning kernel. Each agent type is defined in JACK^{TM} environment. *Eclipse Version 3.4.1* is used as an integrated development environment (IDE) to code the MABLCS. In the figure 8.5 below a sample view of simulation development environment can be seen.



Main [Java Application] C:\Program Files\Java\jdk1.6.0_20\bin\javaw.exe (02.Ara.2010 17:46:12)

```
package elements;



import java.util.Vector;

public class Main
{
    public static Vector< TruckE > vector = new Vector< TruckE >(); // hc
    public static Vector< Order > vectorOrder = new Vector< Order >(); //
    public static int[] numberOfOrdersInNetwork=new int[20];

  public static void main( String args[] )

  {

```

```
TRUCK AGENT  JustGoToPickUpPoint Plan---- Latest pick up is 41340
TRUCK AGENT  JustGoToPickUpPoint Plan---- Time for distance of pickup 0
TRUCK AGENT  JustGoToPickUpPoint Plan---- Slack time of pick up is 41340

I am Truck 3, I can negotiate now


I am Truck 4, I can negotiate now

ORDER AGENT ££££££££££££££££££££££££££££££££££  I have received updated cost
ORDER AGENT ££££££££££££££££££££££££££££££££££ I am Order agent:4Previous Cost is:  136
ORDER AGENT ££££££££££££££££££££££££££££££££££ I am Order agent:4Current Cost is:  135
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++Time is okey I am checking for cost
+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++Cost after TiRes and 262
```

Figure 8. 5. Simulation development environment view

The parameters of the simulation are defined to the system within the *RunSimulation.java* file (APPENDIX D). Therefore there is not a GUI for the definition of the simulation parameters. The parameters are directly typed in the *Eclipse Version 3.4.1* IDE.

When the simulation is run the main menu in figure 8.6 is obtained. The parameters defined to the system might be simulated by pressing the *Simulate* button. The simulation environment can present the current simulation results by pressing the *Simulation Results* button.

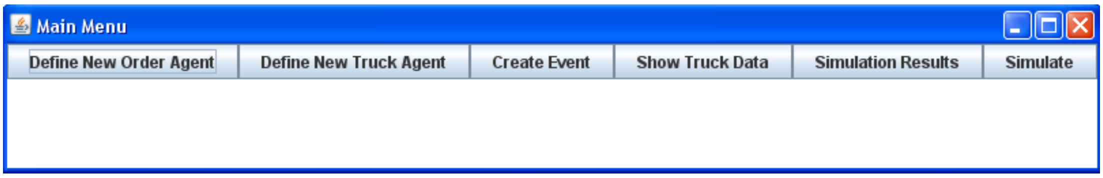


Figure 8. 6. Main menu of the simulation environment

The truck network positions, their operation status (moving, waiting idle etc.) could be seen by clicking *Show Truck Data* button. The menu given in figure 8.6 can also be used for the real time order and truck definition and event creation.

Before presenting the simulation results of the experimental case study the assumptions and system simulation environment is defined as follows;

The simulation assumptions and simulation environment

1. It is assumed that truck agents have time to make a decision within the time interval of consecutive orders
2. Transportation operations are non-preemptive
3. Loading and unloading times are assumed to be included in the transportation time
4. Transportation times are deterministic
5. Truck agents can transport more than one orders at a time and they do not have to return to their initial position
6. Trucks have a constant speed during their transportation operations
7. Truck schedules **are not** fixed. They might change in every time epoch during the simulation
8. Pick-up and delivery points are independent and uniformly distributed
9. Orders arrive to system dynamically
10. Because of the nature of the 3PL operations, there is not a constraint that specifies a trip that start and end at the depot
11. The 3PL company can find drivers on each network position so there is not a break constraint for the truck agents (in other words the truck driver scheduling is not considered, they are thought as attached to truck agents)
12. Pickup and delivery constraints of the orders are hard constraints
13. Truck agents can perform the pickup and delivery operations anytime within order agent pickup and delivery time windows
14. Truck agents can wait at any node of transportation network
15. Negotiation time between any two agents is neglected
16. The order agents can directly negotiate with the truck agents without waiting in a queue. However a truck agent can negotiate with only a single order

agent at a time to provide the atomicity of its operations (the errors due do the common share of beliefsets of the truck agents by multi-threads of order agents are prevented).

17. The response time of the truck agents to the call for proposal of the order agents are ignored.
18. Route diversion of the truck agents due to the new order arrival to the system is not considered therefore when a truck agents gets the next order from its schedule it picks up or delivers it.
19. Truck agents can not negotiate with more than one order concurrently
20. Pickup and delivery points of the order agents are known exactly and the shortest distances are known.
21. The customer orders might arrive to system any time during the operation of the system, this is reasonable for the operations of 3PL. Because 3PL companies operate on a widely geographic area where there is a huge time zone differences.
22. All the trucks in the system satisfies the regulations to move within the transportation network of the company

Under this assumptions and environment definitions the system can be simulated and tested for some different parameters. As the system is run the performance of the system can be measured.

The potential performance measures can be stated as follows;

- The average cost for each weight and volume unit
- Mean and standard deviation of the average length of loaded and of empty movements for each scenario
- Ratio of time spent empty to time spent loaded
- Operating profit generated per vehicle per time period
- Number of order agents permanently rejected due to profitability limitation
- Number of order agents permanently rejected due to feasibility limitation
- Total distance traveled by the truck agents
- Total cost of the system

8.7 Case and Parameter Definition

Because there are so many stochastic events in the proposed MABLCS, some intensive stochastic simulations should be carried out to evaluate its performance. The simulation experiments are conducted in order to prove the capabilities of the MABLCS in practical applications. Because there are so many heavy computational complexities for an individual simulation run, it is not practical to obtain a full factorial experimental design. Due to the abundance of computations, some scenarios are tested rather than analyzing the full factorial scenario occurrences of the proposed system.

Case definition

Cases are obtained by switching on and off some system capabilities of the proposed MABLCS. The **scenarios** are some different parameter sets (vectors) that would be used during the case tests. In order to show the capabilities of the MABLCS four different **cases** are tested under a selected scenario.

Case 1: Full functional (Benchmark)

Case 2: Profitability check capability is off

Case 3: Change/exchange capability is off

Case 1 comprises all the system capabilities that were explained in the previous sections. This case is used as a benchmark for the other cases. It shows the full capability of the proposed MABLCS. In **Case 2**, all the system capabilities are used except the *profitability check* mechanism of the truck agents that controls the contribution of the orders to the overall system profitability. By doing so the average profitability of the system seems to be reduced while increasing the number of orders transported. **Case 3** comprises all the system capabilities except the change/exchange of the order agents between truck schedules capability in order to show the contribution of this capability to overall system performance.

Parameter definition (scenario definition)

In this case study, the pickup time of the orders are assumed to be later than their response time ends ($\tau^{ARV} + T_i^{RES} \leq \tau_i^{AVL}$). This is reasonable, because the response time of any order request is not very long in practice of 3PL operations to cover the pickup time of the respective order. The pickup time of the order agents are presented as $\tau_i^{ARV} + T_i^{ADV}$ where T_i^{ADV} is a discrete random variable that has a uniform distribution. And the delivery of the order agents is presented as $\tau_i^{ARV} + T_i^{ADV} + L_i + T_i^{SLK}$ where L_i is known and T_i^{SLK} is a discrete random variable. The T_i^{RES} is also a discrete random variable. The values of T_i^{ADV}, T_i^{SLK} and T_i^{RES} are drawn from some independent uniform distributions with mean T_i^{ADV}, T_i^{SLK} and T_i^{RES} and ranges $[0,\ 2T_i^{ADV}]$, $[\,0,\ 2T_i^{SLK}]$ and $[0,\ 2T_i^{RES}]$ respectively. The origin and destination of the arriving orders are generated from a set where the 70 % of the orders have the pickup and delivery point attributes of "**14**" (Turkey). This is because the 3PL company where the case study is performed is located in Turkey and most of the transportation operations are from node "**14**". There are 15 remaining nodes. Other nodes of the transportation network have 2% each for pickup and delivery points in the population set. Therefore in total 2% x 15+ 70% =100% is obtained. The origin and destination nodes of any particular order agents can be assigned from this set with the exception of the value for origin drawn must be different than the value of the destination. Some of the orders might be cancelled after their assignment/consolidation to a truck agent. The stochastic behavior of the order cancelation is reflected to the MABLCS. On average 10% of the orders are cancelled after their assignment/consolidation.

The volume and weight attributes of the orders are also discrete random variable where $\mathbf{v}_i$ and $\mathbf{w}_i$ are drawn independently from uniform distributions with mean $\mathbf{v}_i$ and $\mathbf{w}_i$ and ranges $[1, 2\mathbf{v}_i]$ and $[1, 2\mathbf{w}_i]$ respectively. The lower bound for the $\mathbf{v}_i$ and $\mathbf{w}_i$ is 1 to generate orders which has a physical property. The threshold value that is used in the truck load consolidation reasoning is presented with μ. The tariff for the transportation of the LTL orders is presented with θ. Table 8.7 presents the list of parameters and some sample parameter values. In table 8.7, T represents the number of truck agents in the system; O denotes the number of order agents released to the system. **λ** represents the inter arrival time between the order agents.

The units of the parameters are as follows; T_i^{ADV} , T_i^{SLK} , T_i^{RES} and **λ** are in terms of *time units (in this case study time unit is considered as day)*, $\mathbf{v}_i$ is in terms of *meter cube*, $\mathbf{w}_i$ is in terms of *ton* and θ is in *$ / km x ton*.

Table 8. 7. Parameters and sample values

i	T	O	λ	T_i^{ADV}	T_i^{SLK}	T_i^{RES}	$\mathrm{v_i}$	$\mathrm{w_i}$	μ	θ
1	5	30	1	10	20	1.5	15	8	5	0.2
2	10	50	0.5	25	30	2	40	7	1	0.2
3	20	80	0.5	25	10	2	40	7	1	0.1

For each truck agent in the MABLCS a system variable records the total number of distance traveled in terms of kilometers. Each operation of the truck agents are also recorded during the simulation run. Therefore the occupied volume and occupied weight of each truck is known at the end of each simulation run. For instance if the truck agent travels from node i to node j with a volume utilization of 30 meter cube and weight utilization of 10 tons and if the traveling distance is 50 kilometers then the contribution of the transportation to the total system is calculated as 50x30 = 1500 meter cube x kilometers and 50x10 = 500 tons x kilometers. By doing so the average volume and weight utilizations of the truck agents are calculated at the end of each simulation run.

The order agents that live in the MABLCS leave the system whether being transported or rejected. If any order agent is being transported, then the data of time of being picked-up and delivered are recorded. And if the order agents are rejected from the system again the time of rejection is recorded. In addition to the time of rejection, the reason of rejection is also recorded by the order agents. There are two types of permanent rejection of the order agents which are; *feasibility* where the order is not feasible for the transportation with the truck agents within the system or *profitability* where the transportation of the order agent does not contribute to the profitability of the system.

The independency of the runs is provided to system by changing the seed values where the variables are being generated.

8.8 Simulation Results

Sample simulation output

When any setup is run in the proposed MABLCS three different outputs are obtained;

- Detailed order agents results
- Detailed truck agents operations results
- The summary output which summarizes the general system parameter values

When the MABLCS is run with a sample parameter vector the output report given in the following tables are obtained. Table 8.8 represents the output report for the order agents. Table 8.8 presents the order agent names, their pickup points, their destination points, their distance between pickup and delivery point, their volume and weight attributes, their response time (T_i^{RES}), their transportation status whether transported or not and the rejection reason if they are rejected from the system. This table is given in order to show how the simulation output is obtained by the MABLCS.

Table 8. 8. Sample order agent detailed report

Order	From	To	Distance	Volume	Weight	TiRES	TrStatus	Rejection Reason
1	15	14	2528	25	1	3	0	Cancellation
9	7	14	4967	16	7	3	1	Null
4	14	0	3842	12	4	3	1	Null
3	14	2	4047	4	2	3	1	Null
0	14	0	3842	5	11	3	1	Null
7	14	1	4317	13	13	3	1	Null
8	5	9	1358	23	4	3	1	Null
2	11	14	3500	23	7	3	1	Null
6	14	5	2600	4	13	3	1	Null
5	14	2	4047	29	7	3	1	Null

Second output of the system is about the truck agent operations. As it can be seen from the table 8.9, 3 different trucks are used in order to transport 10 different orders arriving to system. Although more than 3 trucks are defined to the system only 3 trucks are used. The operations of truck 1 are highlighted in the table 8.9. As it can be seen from the table, there are some consecutive pickup and delivery operations which represent the load consolidation capability of the truck 1 agent.

Table 8. 9. Sample truck agent detailed report

Truck	Order No	From	To	Final Volume	Final Weight	Distance	Type
Truck 0	0	0	14	5	11	3842	pick-up
Truck 0	3	14	14	9	13	0	pick-up
Truck 0	4	14	14	21	17	0	pick-up
Truck 0	4	14	0	9	13	3842	delivery
Truck 0	3	0	2	5	11	655	delivery
Truck 0	0	2	0	0	0	655	delivery
Truck 1	**2**	**0**	**11**	**23**	**7**	**2300**	**pick-up**
Truck 1	**5**	**11**	**14**	**52**	**14**	**3500**	**pick-up**
Truck 1	**2**	**14**	**14**	**29**	**7**	**0**	**delivery**
Truck 1	**6**	**14**	**14**	**33**	**20**	**0**	**pick-up**
Truck 1	**6**	**14**	**5**	**29**	**7**	**2600**	**delivery**
Truck 1	**5**	**5**	**2**	**0**	**0**	**852**	**delivery**
Truck 2	9	0	7	16	7	1164	pick-up
Truck 2	9	7	14	0	0	4967	delivery
Truck 2	7	14	14	13	13	0	pick-up
Truck 2	8	14	5	36	17	2600	pick-up
Truck 2	7	5	1	23	4	615	delivery
Truck 2	8	1	9	0	0	1955	delivery

The schedule representation given in chapter 5 is used in figure 8.7 to represent the truck 1 agent operations. Figure 8.7 and 8.8 represents the physical operations of truck 1 agent.

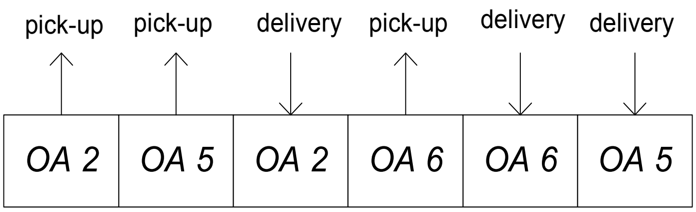


Figure 8. 7. Truck 1 agent schedule representation

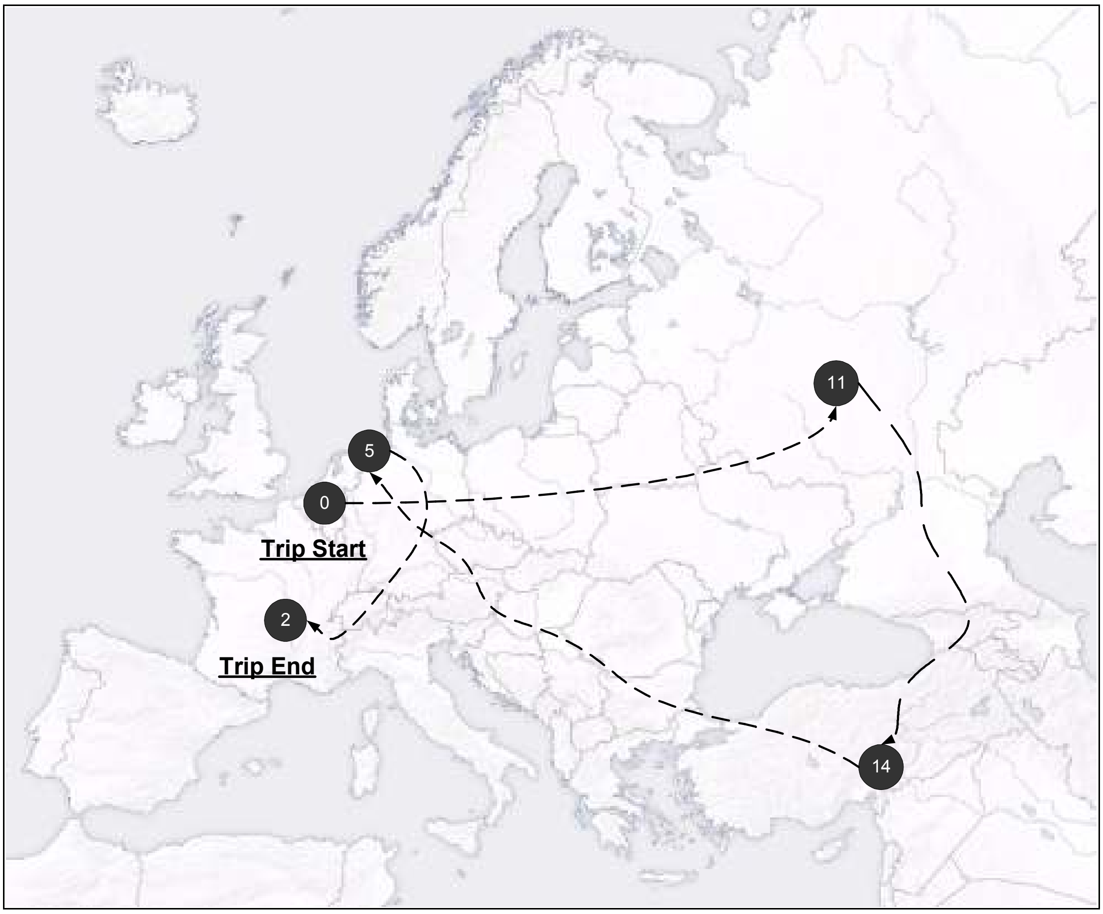


Figure 8. 8. Operation of truck 1 agent

The operations of truck 1 comply with all the time windows specified while the creation of the order agents.

The third output report of the proposed MABLCS is the summary of the parameter values as stated previously. Below in table 8.10 a sample summary output can be seen where; **TDV:** Total distance x volume (km x meter cube), **TDW:** Total distance x weight (km x ton), **TVT:** Total volume transported (meter cube), **TWT :** Total weight transported (ton), **RR:** Raw revenue ($), **TR:** Total revenue ($), **TD:** Total distance (km), **TCO:** Total cost of operation ($), **P:** Profit ($), **ACPV:** Average cost per volume ($/meter cube), **ACPW:** Average cost per weight ($/ton), **APPV:** Average profit per volume ($/meter cube), **APPW:** Average profit per weight ($/ton), **ORF:** Number of orders rejected due to feasibility, **ORP:** Number of orders rejected due to profitability, **ORC:** Number of orders rejected due to cancellation **OT:** Number of orders transported.

Table 8. 10. Sample summary output

TDV	TDW	TVT	TWT	RR	TR	TD	TCO
713619	397229	113	77	312707	156354	51318	48923

P	ACPV	ACPW	APPV	APPW	ORF	ORP	ORC	OT
107430	433	635	951	1395	0	0	1	9

This table show a sample output where in total 9 orders are transported by the truck agents and 1 order is cancelled after its assignment/consolidation.

8.8.1 Case 1 outputs

When the MABLCS is run for the **case 1** with the first parameter vector in table 8.7 (i=1) then the output reports below are obtained. Table 8.11 presents the detailed output for the order agents processed in the system. Some of the orders are rejected due to *feasibility*, *profitability* or *cancellation* reasons and most of them are transported to their destinations within the time interval they propose.

Table 8. 11. Detailed output of the order agents for case 1

Order	From	To	Distance	Volume	Weight	TiRES	TrStatus	Rejection Reason
1	14	8	2871	9	9	1	0	Feasibility
5	11	14	3500	1	4	2	0	Profitability
7	13	14	4565	6	1	1	0	Profitability
8	13	15	1989	16	6	2	0	Profitability
11	15	14	2528	8	12	2	0	Cancellation
16	14	1	4317	28	4	2	0	Profitability
19	2	0	655	10	7	0	0	Profitability
3	14	0	3842	23	12	1	1	Null
23	14	6	2200	8	16	2	0	Feasibility
24	14	2	4047	20	16	1	0	Feasibility
21	14	1	4317	30	12	2	0	Cancellation
25	14	5	2600	13	16	2	0	Feasibility
2	3	0	540	2	13	0	1	Null
9	9	14	4615	28	2	2	1	Null
6	14	1	4317	29	7	2	1	Null
13	14	2	4047	17	9	0	1	Null
14	14	1	4317	17	10	1	1	Null
17	14	0	3842	7	7	1	1	Null
12	14	12	4841	3	4	1	1	Null
0	14	0	3842	2	13	1	1	Null
29	14	9	4615	14	8	2	1	Null
4	15	14	2528	24	12	2	1	Null
28	12	14	4841	14	12	1	1	Null
22	2	14	4047	11	7	0	1	Null
26	14	8	2871	23	11	1	1	Null
18	0	3	540	10	10	1	1	Null
10	14	0	3842	13	10	2	1	Null
15	4	14	2018	18	6	1	1	Null
27	14	2	4047	30	9	1	1	Null
20	14	2	4047	24	10	1	1	Null

Table 8.12 presents the truck operations where all the truck agents are used during the simulation run. The order agents which have been reported as ***1*** in the *TrStatus* column in table 8.11 are being transported by the truck agents defined to the system. Table 8.12 presents the truck agent operations by grouping the operations of the same truck agent.

Table 8. 12. Detailed output presenting the operations of truck agents for case 1

Truck	Order No	From	To	Final Volume	Final Weight	Distance	Type
Truck 0	2	0	3	2	13	540	pick-up
Truck 0	2	3	0	0	0	540	delivery
Truck 0	0	0	14	2	13	3842	pick-up
Truck 0	0	14	0	0	0	3842	delivery
Truck 0	4	0	15	24	12	2234	pick-up
Truck 0	4	15	14	0	0	2528	delivery
Truck 1	28	0	12	14	12	1546	pick-up
Truck 1	22	12	2	25	19	1084	pick-up
Truck 1	28	2	14	11	7	4047	delivery
Truck 1	26	14	14	34	18	0	pick-up
Truck 1	22	14	14	23	11	0	delivery
Truck 1	26	14	8	0	0	2871	delivery
Truck 1	20	8	14	24	10	2871	pick-up
Truck 1	20	14	2	0	0	4047	delivery
Truck 2	9	0	9	28	2	1552	pick-up
Truck 2	9	9	14	0	0	4615	delivery
Truck 2	10	14	14	13	10	0	pick-up
Truck 2	18	14	0	23	20	3842	pick-up
Truck 2	18	0	3	13	10	540	delivery
Truck 2	10	3	0	0	0	540	delivery
Truck 3	3	0	14	23	12	3842	pick-up
Truck 3	6	14	14	52	19	0	pick-up
Truck 3	3	14	0	29	7	3842	delivery
Truck 3	13	0	14	46	16	3842	pick-up
Truck 3	6	14	1	17	9	4317	delivery
Truck 3	13	1	2	0	0	933	delivery
Truck 3	29	2	14	14	8	4047	pick-up
Truck 3	29	14	9	0	0	4615	delivery
Truck 3	15	9	4	18	6	3658	pick-up
Truck 3	15	4	14	0	0	2018	delivery
Truck 3	27	14	14	30	9	0	pick-up
Truck 3	27	14	2	0	0	4047	delivery
Truck 4	14	0	14	17	10	3842	pick-up
Truck 4	12	14	14	20	14	0	pick-up
Truck 4	17	14	14	27	21	0	pick-up
Truck 4	14	14	1	10	11	4317	delivery
Truck 4	17	1	0	3	4	518	delivery
Truck 4	12	0	12	0	0	1546	delivery

The summary output (table 8.13) reveals that the total operating cost (TCO) is $ 82,801 for the operation of the 19 LTL operations. The total revenue earned during the same simulation run is $117,290 therefore in total $34,488 profit is earned during these operations.

Table 8. 13. Summary output for case 1

TDV	TDW	TVT	TWT	RR	TR	TD	TCO
1421920	678455	309	172	586449	117290	86465	82801

P	ACPV	ACPW	APPV	APPW	ORF	ORP	ORC	OT
34488	268	481	112	201	4	5	2	19

8.8.2 Case 2 outputs

When the MABLCS is run for the **case 2** with the first parameter vector in table 8.7 (*i*=1) then the output reports below are obtained. Table 8.14 presents the detailed output for the order agents processed in the system. As it is presented in the table the order agents are rejected from the system only due to the *feasibility* reason. This is because the profitability check of the agent types is switched-off in this case study.

Table 8. 14. Detailed output of the order agents for case 2

Order	From	To	Distance	Volume	Weight	TiRES	TrStatus	Rejection Reason
1	14	8	2871	9	9	1	0	Feasibility
14	14	1	4317	17	10	1	0	Feasibility
18	0	3	540	10	10	1	0	Feasibility
3	14	0	3842	23	12	1	1	Null
19	2	0	655	10	7	0	0	Feasibility
21	14	1	4317	30	12	2	0	Feasibility
22	2	14	4047	11	7	0	0	Feasibility
23	14	6	2200	8	16	2	0	Feasibility
24	14	2	4047	20	16	1	0	Feasibility
2	3	0	540	2	13	0	1	Null
11	15	14	2528	8	12	2	1	Null
9	9	14	4615	28	2	2	1	Null
7	13	14	4565	6	1	1	1	Null
28	12	14	4841	14	12	1	0	Feasibility
6	14	1	4317	29	7	2	1	Null
16	14	1	4317	28	4	2	1	Null
0	14	0	3842	2	13	1	1	Null
8	13	15	1989	16	6	2	1	Null
13	14	2	4047	17	9	0	1	Null
29	14	9	4615	14	8	2	1	Null
4	15	14	2528	24	12	2	1	Null
17	14	0	3842	7	7	1	1	Null
27	14	2	4047	30	9	1	1	Null
26	14	8	2871	23	11	1	1	Null
12	14	12	4841	3	4	1	1	Null
15	4	14	2018	18	6	1	1	Null
25	14	5	2600	13	16	2	1	Null
5	11	14	3500	1	4	2	1	Null
10	14	0	3842	13	10	2	1	Null
20	14	2	4047	24	10	1	1	Null

Table 8.15 presents the truck operations where all the truck agents are used during the simulation run. The order agents which have been reported as ***1*** in the *TrStatus* column in table 8.14 are being transported by the truck agents defined to the system. Table 8.15 presents the truck agent operations by grouping the operations of the same truck agent.

Table 8. 15. Detailed output presenting the operations of truck agents for case 2

Truck	Order No	From	To	Final Volume	Final Weight	Distance	Type
Truck 0	2	0	3	2	13	540	pick-up
Truck 0	2	3	0	0	0	540	delivery
Truck 0	0	0	14	2	13	3842	pick-up
Truck 0	0	14	0	0	0	3842	delivery
Truck 0	4	0	15	24	12	2234	pick-up
Truck 0	27	15	14	54	21	2528	pick-up
Truck 0	4	14	14	30	9	0	delivery
Truck 0	27	14	2	0	0	4047	delivery
Truck 1	3	0	14	23	12	3842	pick-up
Truck 1	3	14	0	0	0	3842	delivery
Truck 1	6	0	14	29	7	3842	pick-up
Truck 1	6	14	1	0	0	4317	delivery
Truck 1	29	1	14	14	8	4317	pick-up
Truck 1	17	14	14	21	15	0	pick-up
Truck 1	29	14	9	7	7	4615	delivery
Truck 1	17	9	0	0	0	1552	delivery
Truck 1	5	0	11	1	4	2300	pick-up
Truck 1	5	11	14	0	0	3500	delivery
Truck 2	7	0	13	6	1	1713	pick-up
Truck 2	9	13	9	34	3	791	pick-up
Truck 2	9	9	14	6	1	4615	delivery
Truck 2	7	14	14	0	0	0	delivery
Truck 2	25	14	14	13	16	0	pick-up
Truck 2	25	14	5	0	0	2600	delivery
Truck 2	20	5	14	24	10	2600	pick-up
Truck 2	20	14	2	0	0	4047	delivery
Truck 3	8	0	13	16	6	1713	pick-up
Truck 3	13	13	14	33	15	4565	pick-up
Truck 3	15	14	4	51	21	2018	pick-up
Truck 3	8	4	15	35	15	1780	delivery
Truck 3	13	15	2	18	6	2681	delivery
Truck 3	15	2	14	0	0	4047	delivery
Truck 3	10	14	14	13	10	0	pick-up
Truck 3	10	14	0	0	0	3842	delivery
Truck 4	11	0	15	8	12	2234	pick-up
Truck 4	11	15	14	0	0	2528	delivery
Truck 4	12	14	14	3	4	0	pick-up
Truck 4	16	14	14	31	8	0	pick-up
Truck 4	16	14	1	3	4	4317	delivery
Truck 4	26	1	14	26	15	4317	pick-up
Truck 4	26	14	8	3	4	2871	delivery
Truck 4	12	8	12	0	0	1974	delivery

Summary output report is given in table 8.16.

Table 8. 16. Summary output for case 2 parameter

TDV	TDW	TVT	TWT	RR	TR	TD	TCO
1502879	721901	329	176	571161	114232	104953	96392

P	ACPV	ACPW	APPV	APPW	ORF	ORP	ORC	OT
17840	293	548	54	101	9	0	0	21

8.8.3 Case 3 outputs

When the MABLCS is run for the **case 3** with the first parameter vector in table 8.7 (*i*=1) then the output reports below are obtained. Table 8.17 below presents the detailed output for the order agents processed in the system. Some of the orders are rejected due to *feasibility*, *profitability* or *cancellation* reasons and most of them are transported to their destinations within the time interval they propose.

Table 8. 17. Detailed output of the order agents for case 3

Order	From	To	Distance	Volume	Weight	TiRES	TrStatus	Rejection Reason
1	14	8	2871	9	9	1	0	Feasibility
7	13	14	4565	6	1	1	0	Profitability
6	14	1	4317	29	7	2	0	Profitability
8	13	15	1989	16	6	2	0	Profitability
9	9	14	4615	28	2	2	0	Profitability
12	14	12	4841	3	4	1	0	Profitability
11	15	14	2528	8	12	2	0	Cancellation
16	14	1	4317	28	4	2	0	Profitability
19	2	0	655	10	7	0	0	Profitability
3	14	0	3842	23	12	1	1	Null
24	14	2	4047	20	16	1	0	Feasibility
21	14	1	4317	30	12	2	0	Cancellation
25	14	5	2600	13	16	2	0	Feasibility
2	3	0	540	2	13	0	1	Null
26	14	8	2871	23	11	1	0	Profitability
29	14	9	4615	14	8	2	0	Feasibility
14	14	1	4317	17	10	1	1	Null
13	14	2	4047	17	9	0	1	Null
0	14	0	3842	2	13	1	1	Null
28	12	14	4841	14	12	1	1	Null
4	15	14	2528	24	12	2	1	Null
27	14	2	4047	30	9	1	1	Null
22	2	14	4047	11	7	0	1	Null
17	14	0	3842	7	7	1	1	Null
18	0	3	540	10	10	1	1	Null
23	14	6	2200	8	16	2	1	Null
15	4	14	2018	18	6	1	1	Null
5	11	14	3500	1	4	2	1	Null
10	14	0	3842	13	10	2	1	Null
20	14	2	4047	24	10	1	1	Null

Table 8.18 presents the truck operations where all the truck agents are used during the simulation run. The order agents which have been reported as ***1*** in the *TrStatus* column in table 8.17 are being transported by the truck agents defined to the system. Table 8.18 presents the truck agent operations by grouping the operations of the same truck agent.

Table 8. 18. Detailed output presenting the operations of truck agents for case 3

Truck	Order No	From	To	Final Volume	Final Weight	Distance	Type
Truck 0	2	0	3	2	13	540	pick-up
Truck 0	2	3	0	0	0	540	delivery
Truck 0	0	0	14	2	13	3842	pick-up
Truck 0	0	14	0	0	0	3842	delivery
Truck 0	4	0	15	24	12	2234	pick-up
Truck 0	27	15	14	54	21	2528	pick-up
Truck 0	4	14	14	30	9	0	delivery
Truck 0	27	14	2	0	0	4047	delivery
Truck 1	3	0	14	23	12	3842	pick-up
Truck 1	3	14	0	0	0	3842	delivery
Truck 1	22	0	2	11	7	655	pick-up
Truck 1	22	2	14	0	0	4047	delivery
Truck 1	5	14	11	1	4	3500	pick-up
Truck 1	5	11	14	0	0	3500	delivery
Truck 2	15	0	4	18	6	2485	pick-up
Truck 2	10	4	14	31	16	2018	pick-up
Truck 2	15	14	14	13	10	0	delivery
Truck 2	20	14	14	37	20	0	pick-up
Truck 2	10	14	0	24	10	3842	delivery
Truck 2	20	0	2	0	0	655	delivery
Truck 3	14	0	14	17	10	3842	pick-up
Truck 3	13	14	14	34	19	0	pick-up
Truck 3	14	14	1	17	9	4317	delivery
Truck 3	13	1	2	0	0	933	delivery
Truck 3	23	2	14	8	16	4047	pick-up
Truck 3	23	14	6	0	0	2200	delivery
Truck 4	28	0	12	14	12	1546	pick-up
Truck 4	18	12	0	24	22	1546	pick-up
Truck 4	28	0	14	10	10	3842	delivery
Truck 4	17	14	14	17	17	0	pick-up
Truck 4	17	14	0	10	10	3842	delivery
Truck 4	18	0	3	0	0	540	delivery

Summary output report is given in table 8.19.

Table 8. 19. Summary output for case 3

TDV	TDW	TVT	TWT	RR	TR	TD	TCO
886232	607066	221	160	508335	101667	72614	71047

P	ACPV	ACPW	APPV	APPW	ORF	ORP	ORC	OT
30620	321	444	139	191	4	8	2	16

8.8.4 Statistical analysis

Each case defined in the previous sections is run for 21 times. The summary output reports are obtained for each simulation run. The following figures present the **TWT/TD** and **TVT/TD** in order to represent the vehicle utilizations in terms of weight and volume capacities respectively. The figures 8.9 and 8.10 show that vehicle utilization for **case 1** where all the system capabilities are active is highest when we compare it with **case 2** and **3.**

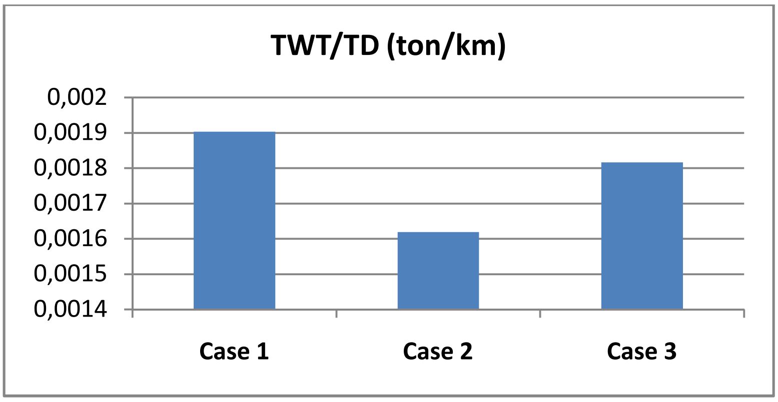


Figure 8. 9. Truck weight utilizations

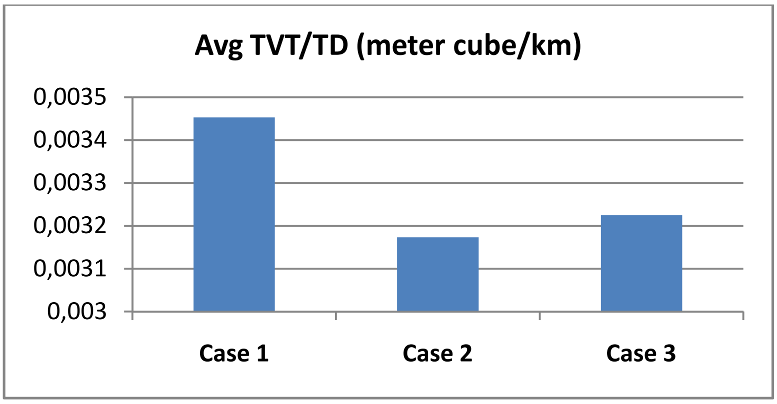


Figure 8. 10. Truck volume utilizations

In figure 8.11 the average profit earned from each case are illustrated. In figure 8.12 the numbers of transportations of the cases are presented. Although the number of transportations of case 1 is less than the number of transportations of case 2, the profit earned is higher than case 2. It is because case 1 utilizes all the agent-based system capabilities.

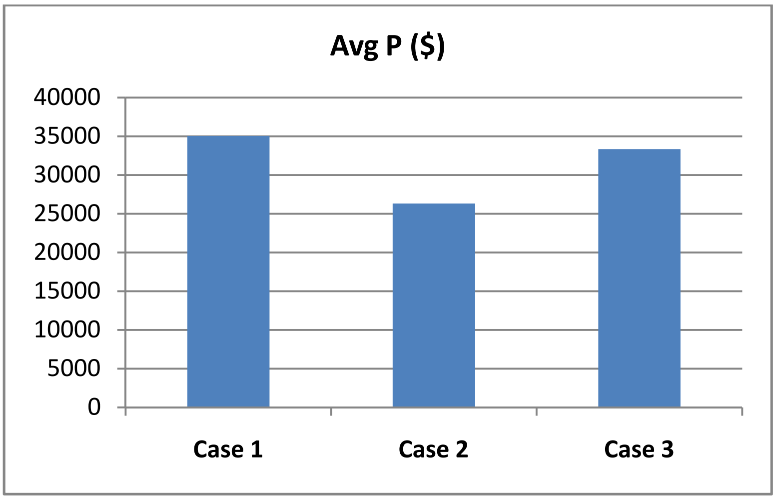


Figure 8. 11. Profitability comparison

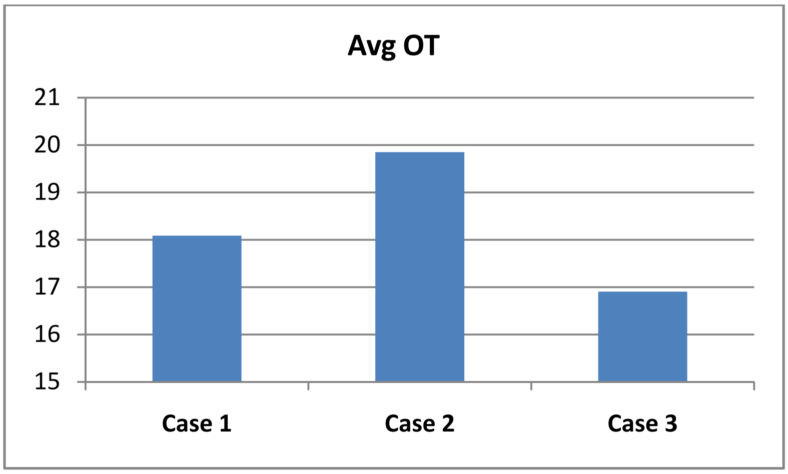


Figure 8. 12. Number of transported orders comparison

8.8.4.1 Case 1 and case 2 comparison

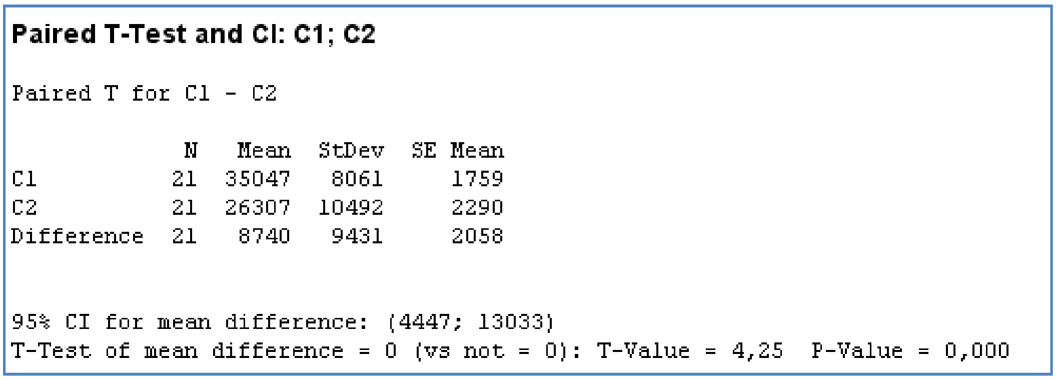
In order to support the average values obtained from the simulation output the question below is asked;

"Is there any evidence of difference in measured performance parameters between **case 1** and other cases?" In order to answer to this question the *repeated measures* or *paired t test* is conducted. The results obtained in section **8.8.1** to **8.8.3** are run for the same input parameter. The input parameter is held fix for the *t* test analysis. However the seed values for the simulation run is changed for each run. 21 different seed values are used for every case. Therefore using *paired t test* to test the evidence of difference between cases seems reasonable. The *paired t-test* is conducted with a confidence level 95%.

The null hypothesis is ;

$$H_0 : \mu_D = 0$$

t-test analysis is conducted in MINITAB. t-test result obtained for the **case 1** and **case 2** from is presented below. As the $p < (\alpha = 0.05)$ the null hypothesis is rejected (see figure 8.13 and 8.14). Therefore we can conclude that there is evidence of difference between case 1 and case 2 with respect to **Profit** parameter.



```
Paired T-Test and CI: C1; C2

Paired T for C1 - C2

             N   Mean  StDev  SE Mean
C1          21  35047   8061     1759
C2          21  26307  10492     2290
Difference  21   8740   9431     2058


95% CI for mean difference: (4447; 13033)
T-Test of mean difference = 0 (vs not = 0): T-Value = 4,25  P-Value = 0,000
```

Figure 8. 13. MINITAB paired t test output for the samples of case 1 and case 2

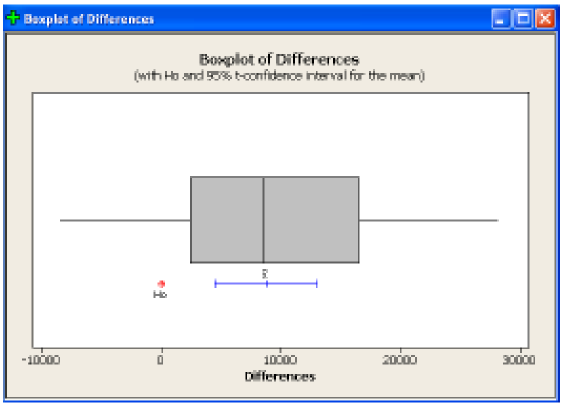
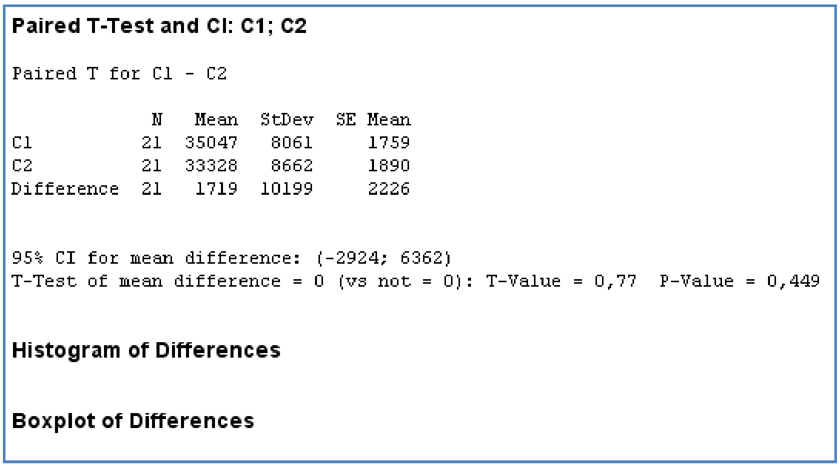


Figure 8. 14. Boxplot of mean differences of case 1 and case 2

8.8.4.2 Case 1 and case 3 comparison

Case 1 and **case 3** are also compared with *paired t test*. As the p > (α = 0.05) the null hypothesis is accepted (see figure 8.15 and 8.16). This output indicates that the change/exchange reasoning mechanism of the agent types does not contribute to the profitability as much as the contribution of *profitability check* mechanism of the agent types. However the average profits of these two cases are $35,047 and $33,328 respectively (see figure 8.15). Therefore although paired t-test accepts the null hypothesis there is $ 1719 difference in their profit averages of **case 1** and **case 2.**



Paired T-Test and CI: C1; C2

```
Paired T for C1 - C2

             N   Mean  StDev  SE Mean
C1          21  35047   8061     1759
C2          21  33328   8662     1890
Difference  21   1719  10199     2226


95% CI for mean difference: (-2924; 6362)
T-Test of mean difference = 0 (vs not = 0): T-Value = 0,77  P-Value = 0,449
```

Histogram of Differences

Boxplot of Differences

Figure 8. 15. MINITAB paired t test output for the samples of case 1 and case 3

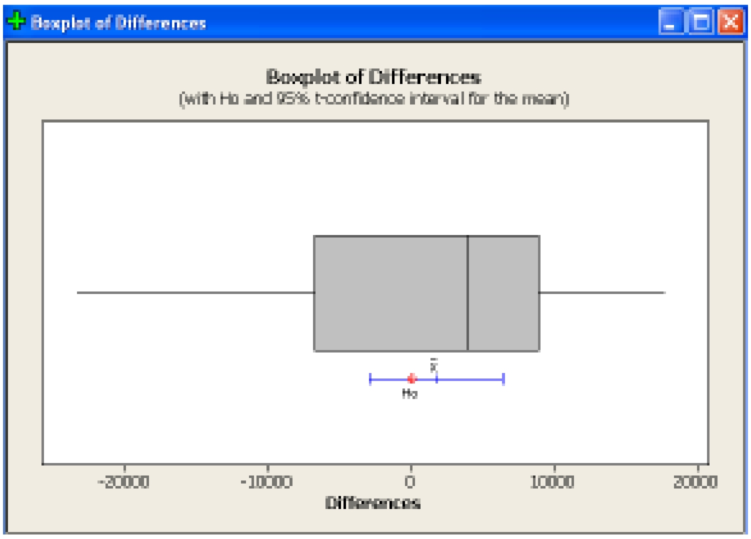


Figure 8. 16. Boxplot of mean differences of case 1 and case 3

The case comparisons reveal that;

(1) Profitability check mechanism contributes on the profitability and truck utilization levels

(2) The change/exchange order agents between truck mechanism provides an improvement (not as much as the profitability check mechanism contributes) on the total transportation system

CHAPTER 9

CONCLUSION

In this thesis, a multi-agent based load consolidation system is proposed. The inspiration behind this study is to support the decisions made during the 3PL company operations. Logistics organizations mainly the ones providing land transportation services are facing with difficulties while making effective operational decisions. This is especially the case in making load/capacity/route planning and load consolidation decisions where customer orders are generally unpredictable and subject to sudden changes.

Throughout this thesis, first of all an introduction is given about why such a system is required for the third-party logistics organizations. Then, in chapter 2, the literature about the classical load consolidation subject is reviewed. While giving the literature review the aspects of necessity of such a system is also emphasized. In chapter 3, multi-agent systems and their general properties are specified. The inspiration behind the usage of multi-agent paradigms to logistics domain is discussed in this chapter. In chapter 4, the definition of the proposed multi-agent based load consolidation problem is given. The notations that is used in the further sections of the thesis are defined. In chapter 5, the conceptual design is made by using a well-known design methodology (Prometheus design methodology). The details of the design phases are presented in this chapter. The interactions and interaction protocols between agent types are designed in this chapter. In chapter 6, the proposed MABLCS is implemented by using a popular agent development environment (JACK^{TM} development environment), this chapter presents the implementation phase. The implementation of reasoning mechanisms of the agent types is illustrated. In chapter 7, the performance of the proposed MABLCS was tested by using a well-known problem domain named as *vehicle routing problem with time windows.* A set of problem is run on the proposed system. The results obtained in this chapter reveal that the proposed system can follow the solution patterns of the dedicated vehicle routing algorithms. In chapter 8, the performance of the proposed MABLCS is tested under stochastic environment by simulating some test bed examples. In order to show the load consolidation performance of the proposed MABLCS a set of experiments are conducted with the ***real business data*** of a 3PL company in this chapter. The data of the 3PL company is used in the experimental case study to show the direct contribution of the proposed system to real 3PL companies. The contribution of the proposed MABLCS is tested with different experiment setups. The experiment setups are designed for the cases where some of the system functionalities of the MABLCS are included or not. The cost data used in the proposed system is derived by analyzing the historical operations costs of the company. The true costs of operations are derived by applying activity based costing to the company. The cost approximations derived by using the function given in equation 5.1 is compared with the cost results of activity based costing.

In summary, the proposed MABLCS pursues the load consolidation objective while conducting the following operations which are distributed to agent elements by utilizing negotiation and BDI reasoning capabilities of the agent types.

- **Load acceptance/rejection**
- **Load assignment**
- **Re-assignment**
- **Routing**
- **Scheduling**

The proposed system inherently include these operation decisions because making a good load consolidation decision in a dynamic business environment requires the decision of load acceptance/rejection, load assignment, load re-assignment, vehicle routing and vehicle scheduling.

The proposed MABLCS is open for improvement; this is because the reasoning methods the agent types use and the depth of search for the VRPTW of each truck agents could be improved. The reasoning methods given in chapter 5 could be improved in order to obtain more intelligent agent types who make better operational decisions. Due do its distribute nature, in the proposed system there is not a central dispatching of the trucks, the trucks agents path their own way by reasoning their plans. The agent types defined in the system use their own distributed artificial intelligence to make their own decision. However, their artificial intelligence might be improved by adding more capabilities. The vehicle routing problem algorithm that the truck agents use also has a considerable effect on the performance of the overall 3PL operations. Some solutions which are dominating the plan on hand could be skipped by the truck agents. However when the depth of the searching increases the load of computation increases proportionally. Although the proposed system does not claim to find the optimum transportation alternatives, the model would be improved to obtain optimum decisions in each business case.

9.1 Further Studies

In the further studies, the order agents which are rejected by the truck agents due to the schedule limitations (or capacity limitations) might change their schedule limitations (with a relaxed pickup and delivery time) and they can re-negotiate with the truck agents. This depends on the decision of the shipper as a third-party. Because, the relaxation of the pickup and delivery times depends on the shipper's decision.

The current system can also be improved or extended in a number of ways:

1. En-route diverting a truck away from its current pickup or delivery operation (while it is on the way of next operation) when a new order request result in a more advantageous consolidation.

2. Learning capabilities can be gained by the truck agents that let them to reason about the seasonal order requests on some transportation network nodes.

3. The truck agents can be run on different platforms to use the brute force of computing of running truck agents on different processors in spite of running them on different threads of a single processor.

4. Vehicle routing algorithms which run in distributed nature which result in near-optimum solutions can be adapted to the truck order acceptance/rejection plan in order to find better consolidation alternatives.

5. In accepting/rejection and change/exchange decisions of the truck agents the bin-packing model might be integrated. This is because in some physical conditions the orders might not be packed into the container although it is permanently accepted by the truck agents.

6. Driver scheduling can be articulated to the scheduling of the truck agents. This might be conducted by integrating a new type of agent which is called as driver and building the interaction diagram between the truck and driver agents. While the truck agents are making the acceptance/rejection decision they first negotiate with the driver agents to find a feasible driver.

7. The proposed system would be integrated with some mapping software such as Google Map etc. to provide real time positions of the truck agents.

REFERENCES

AOS (2009) JACK Intelligent Agents® Agent Manual. Carlton South, Victoria, AUSTRALIA, Agent Oriented Software Pty. Ltd.

Attanasio, A., Fuduli, A., Ghiani, G. and Triki, C. (2007). Integrated shipment dispatching and packing problems: a case study. *Journal of Mathematical Modelling and Algorithms,* **6** (1), 77-85

Baykasoğlu, A. (2003). Kitlesel özel üretim (in Turkish). *Kalder Forum,* **3** (9), 47-55

Baykasoğlu, A. and Bartık, E. (2005) Modelling work processes of a logistic company for performance improvement and training. *Paper Presented at the 3rd International Logistics & Supply Chain Congress. November 23-24, 2005.* Istanbul, Turkey.

Baykasoğlu, A. and Kaplanoğlu, V. (2006). Developing a service costing system and an application for logistics companies. *International Journal of Agile Manufacturing,* **9** (2), 13-18

Baykasoğlu, A. and Kaplanoğlu, V. (2007). A service-costing framework for logistics companies and a case study. *Management Research News,* **30** (9), 621-633

Baykasoğlu, A. and Kaplanoğlu, V. (2008). Application of activity-based costing to a land transportation company: A case study. *International Journal of Production Economics,* **116** (2), 308-324

Baykasoğlu, A., Kaplanoğlu, V. and Unutmaz, Z. D. (2007) Lojistik firmalarında yük konsolidasyonu için bir model önerisi. *Paper Presented at the YA/EM 2007: Yöneylem Araştırması ve Endüstri Mühendisliği 27. Ulusal Kongresi. July 02-04, 2007.* İzmir, Turkey.

Bellifemine, F., Caire, G. and Greenwood, D. (2007). *Developing Multi-Agent Systems with JADE,* John Wiley & Sons Ltd.

Bodin, L., Golden, B., Assad, A. and Ball, M. (1983). Routing and scheduling of vehicles and crews: The state of the art. *Comp. & Oper. Res.,* **10** (2), 63-211

Bräysy, O. (2001). Genetic algorithms for the vehicle routing problem with time windows. *Arpakannus,* **1** ( ), 33-38

Brennan, J. J. (1981) *Models and analysis of temporal consolidation.* PhD thesis, The University of California

Brintrup, A. (2010). Behaviour adaptation in the multi-agent, multi-objective and multi-role supply chain. *Computers in Industry,* **61** 636-645

Chen, F. Y., Wang, T. and Xu, T. Z. (2005). Integrated inventory replenishment and temporal shipment consolidation: A comparison of quantity-based and time-based models. *Annals of Operations Research,* **135** (1), 197-210

Cil, I. and Mala, M. (2010). A multi-agent architecture for modelling and simulation of small military unit combat in asymmetric warfare. *Expert Systems with Applications,* **37** (2), 1331-1343

Cohen, M. and Stathis, K. (2001). Strategic change stemming from e-commerce: implications of multi-agent systems on the supply chain *Strategic Change,* **10** (3), 139-149

Cook, W. and Rich, J. L. (1999). A parallel cutting-plane algorithm for the vehicle routing problem with time windows. *Computational and Applied Mathematics Rice University, Houston EUA,* 5

Cooper, R. (1988a). The rise of activity based costing-Part One: What is an activity based cost system? *Journal of Cost Management,* **2** (2), 45-54

Cooper, R. (1988b). The rise of activity based costing-Part Two: When do I need an activity-based cost system? *Journal of Cost Management,* **2** (3), 41-48

Cooper, R. and Kaplan, R. S. (1988). How cost accounting distorts product costs. *Management Accounting,* **69** (10), 20-27

Crainic, T. G. (2003). *Long-Haul Freight Transportation,* Massachusetts, Kluwer Academic Publishers.

Crainic, T. G. and Laporte, G. (1997). Planning models for freight transportation. *European Journal of Operational Research,* **97** (3), 409-438

Çetinkaya, S., Mutlu, F. and Lee, C.-Y. (2006). A comparison of outbound dispatch policies for integrated inventory and transportation decisions. *European Journal of Operational Research,* **171** 1094–1112

Dastani, M., Dix, J. and Seghrouchni, A. E. F. (2004). *Programming multi-agent systems,* Springer

Davidsson, P., Henesey, L., Ramstedt, L., Törnquist, J. and Wernstedt, F. (2005). An analysis of agent-based approaches to transport logistics. *Transportation Research Part C,* **13** (4), 255-271

Desrochers, M., Desrosier, J. and Solomon, M. (1992). A new optimization algorithm for the vehicle routing problem with time windows. *Operations Research,* **40** (2), 342-354

Evertsz, R., Fletcher, M., Jones, R., Jarvis, J., Brusey, J. and Dance, S. (2003) Implementing Industrial Multi-agent Systems Using JACK™. *Paper Presented at the First International Workshop, ProMAS.* Melbourne, Australia.

Ferber, J. (1999). *Multi-agent systems: an introduction to distributed artificial intelligence,* Addison-Wesley Harlow.

Fischer, K., Müller, J. P. and Pischel, M. (1996). Cooperative Transportation Scheduling: an Application Domain for DAI. *Applied Artificial Intelligence,* **10** (1), 1-34

Forget, P., D'Amours, S., Frayret, J.-M. and Gaudreault, J. (2009). Study of the performance of multi-behaviour agents for supply chain planning. *Computers in Industry,* **60** (9), 698-708

Fox, M. S., Barbuceanu, M. and Teigen, R. (2000). Agent-Oriented Supply-Chain Management. *International Journal of Flexible Manufacturing Systems,* **12** (2-3), 165-188

Gaudreault, J., Frayret, J. M. and Pesant, G. (2009). Distributed search for supply chain coordination. *Computers in Industry,* **60** (6), 441-451

Gendreau, M., Guertin, F., Potvin, J. Y. and Taillard, E. (1999). Parallel tabu search for real-time vehicle routing and dispatching. *Transportation Science,* **33** (4), 381-390

Georgeff, M. and Lansky, A. (1987) Reactive Reasoning and Planning: an Experiment with a Mobile Robot. *Paper Presented at the 7th National Conference on Artificial Intelligence.* Seattle, WA.

Georgeff, M. and Rao, A. (1998). *Rational software agents: from theory to practice,* Springer.

Gunasekaran, A. and Sarhadi, M. (1998). Implementation of activity-based costing in manufacturing. *International Journal of Production Economics,* **56** 231-242

Gupta, M. and Galloway, K. (2003). Activity-based costing/management and its implications for operations management. *Technovation,* **23** (2), 131-138

Hahn, C., Madrigal-Mora, C. and Fischer, K. (2009). A platform-independent metamodel for multiagent systems. *Autonomous Agents and Multi-Agent Systems,* **18** (2), 239-266

Hall, R. W. (1987). Consolidation Strategy: Inventory, Vehicles And Terminals. *Journal of Business Logistics,* **8** (2), 57-71

Helberg, C., Galletly, J. E. and Bicheno, J. R. (1994). Simulating activity-based costing. *Industrial Management & Data Systems,* **94** (9), 3-8

Higginson, J. K. and Bookbinder, J. H. (1994). Policy recommendations for a shipment-consolidation program. *Journal of Business Logistics,* **15** (1), 87-112

Howden, N., Ronnquist, R., Hodgson, A. and Lucas, A. (2001) JACK Intelligent Agents - Summary of an Agent Infrastructure. *Paper Presented at the 5th International Conference on Autonomous Agents*

Huber, M. (1999) JAM: A BDI-Theoretic Mobile Agent Architecture. *Paper Presented at the 3rd International Conference on Autonomous Agents.* New York, NY.

Irnich, S. and Villeneuve, D. (2003). *The Shortest Path Problem with Resource Constraints and k-cycle Elimination for K>3,* Groupe d'études et de recherche en analyse des décisions, HEC Montréal.

Jackson, G. C. (1981). Evaluating order consolidation strategies using simulation. *Journal of Business Logistics,* **2** (2), 110-138

Johnson, H. T. and Kaplan, R. S. (1987). *Relevance Lost: The rise and fall of management accounting,* Boston, Harvard Business School Press.

Karageorgos, A., Mehandjiev, N., Weichhart, G. and Hämmerle, A. (2003). Agent-based optimisation of logistics and production planning. *Engineering Applications of Artificial Intelligence,* **16** (4), 335-348

Kohl, N., Desrosiers, J., Madsen, O. B. G., Solomon, M. M. and Soumis, F. (1999). 2-path cuts for the vehicle routing problem with time windows. *Transportation Science,* **33** (1), 101

Laporte, G. and Nobert, Y. (1987). Exact algorithms for the vehicle routing problem. *Surveys in Combinatorial Optimization,* **31** 147-184

Larsen, J. (1999) *Parallelization of the vehicle routing problem with time windows.* Ph.D. Thesis, Department of Mathematical Modelling, Technical University of Denmark

Lau, P. K. H. and Lau, H. C. W. (2008). Development of an optimised transport logistics system for supporting distribution operations. *International Journal of Computer Applications in Technology,* **31** (3/4), 187-196

Li, H. and Lim, A. (2003). Local search with annealing-like restarts to solve the VRPTW. *European Journal of Operational Research,* **150** (1), 115-127

Lin, F. R. and Lin, Y. Y. (2006). Integrating multi-agent negotiation to resolve constraints in fulfilling supply chain orders. *Electronic Commerce Research and Applications,* **5** (4), 313-322

Lu, L. and Wang, G. (2007). A study on multi-agent supply chain framework based on network economy. *Computers and Industrial Engineering,* **54** (2), 288-300

Magnanti, T. L. (1981). Combinatorial optimization and vehicle fleet planning: Perspectives and prospects. *Networks,* **11** (2), 179-213

Masters, J. M. (1980). The effects of freight consolidation on consumer service. *Journal of Business Logistics,* **2** (1), 55-74

Min, H. and Cooper, M. (1990). A comparative review of analytical studies on freight consolidation and backhauling. *Logistics and Transportation Review,* **26** (2), 149-169

Monteiro, T., Roy, D. and Anciaux, D. (2007). Multi-site coordination using a multi-agent system. *Computers in Industry,* **58** (4), 367-377

Nachtmann, H. and Al-Rifai, M. H. (2004). An application of activity based costing in the air conditioner manufacturing industry. *The Engineering Economist,* **49** (3), 221-236

Nwana, H., Lee, L. and Jennings, N. (1997). Co-ordination in multi-agent systems. *Software Agents and Soft Computing Towards Enhancing Machine Intelligence,* 42-58

Padgham, L. and Winikoff, M. (2004). *Developing intelligent agent systems a practical guide,* Melbourne, Australia, John Wiley & Sons, Ltd.

Parunak, H. V. D. (1999). *Industrial and practical applications of DAI,* MIT Press, Cambridge, MA.

Rao, A. S. and Georgeff, M. (1995) BDI Agents: from Theory to Practice. *Paper Presented at the 1st International Conference on Multi-Agent Systems.* San Francisco, CA.

Raz, T. and Elnathan, D. (1999). Activity based costing for projects. *International Journal of Project Management,* **17** (1), 61-67

Regan, A. C., Mahmassani, H. S. and Jaillet, P. (1998). Evaluation of dynamic fleet management systems - Simulation framework. *Forecasting, Travel Behavior, and Network Modeling,* (1645), 176-184

Savelsbergh, M. W. P. (1995). The General Pickup and Delivery Problem. *Transportation Science,* **29** (1), 17-29

Sheffi, Y. (1986). Carrier/shipper interaction in the transportation market: An analytic framework. *Journal of Business Logistics,* **7** (1), 1-27

Solomon, M. M. (1987). Algorithms for the vehicle routing and scheduling problems with time window constraints. *Operations Research,* **35** (2), 254-265

Sycara, K. P. (1998). Multiagent systems. *AI magazine,* **19** (2), 79

Tsai, W. H. (1998). Quality cost measurement under activity-based costing. *International Journal of Quality & Reliability Management,* **15** (7), 719-752

Tsai, W. H. and Kuo, L. (2004). Operating costs and capacity in the airline industry. *Journal of Air Transport Management* **10** (4), 269-275

Tyan, J. C., Wang, F. K. and Du, T. C. (2003). An evaluation of freight consolidation policies in global third party logistics. *The International Journal of Management Science,* **31** (1), 55-62

Wang, M., Wang, H., Vogel, D., Kumar, K. and Chiu, D. K. W. (2009). Agent-based negotiation and decision making for dynamic supply chain formation. *Engineering Applications of Artificial Intelligence,* **22** (7), 1046-1055

Wong, T. N. and Fang, F. (2008). A multi-agent protocol for multilateral negotiations in supply chain management. *International Journal of Production Research,* **48** (1), 271-299

Wooldridge, M. (2002). *An introduction to multiagent systems,* England, John Wiley & Sons, Ltd.

Wooldridge, M. and Jennings, N. R. (1995). Intelligent agents: theory and practice. *The knowledge engineering review,* **10** (2), 115-152

Yang, J., Jaillet, P. and Mahmassani, H. (2004). Real-time multivehicle truckload pickup and delivery problems. *Transportation Science,* **38** (2), 135-148

Ying, W. and Dayong, S. (2005). Multi-agent framework for third party logistics in E-commerce. *Expert Systems with Applications,* **29** (2), 431-436

http://www.whitestein.com/library/company-and-product-resources

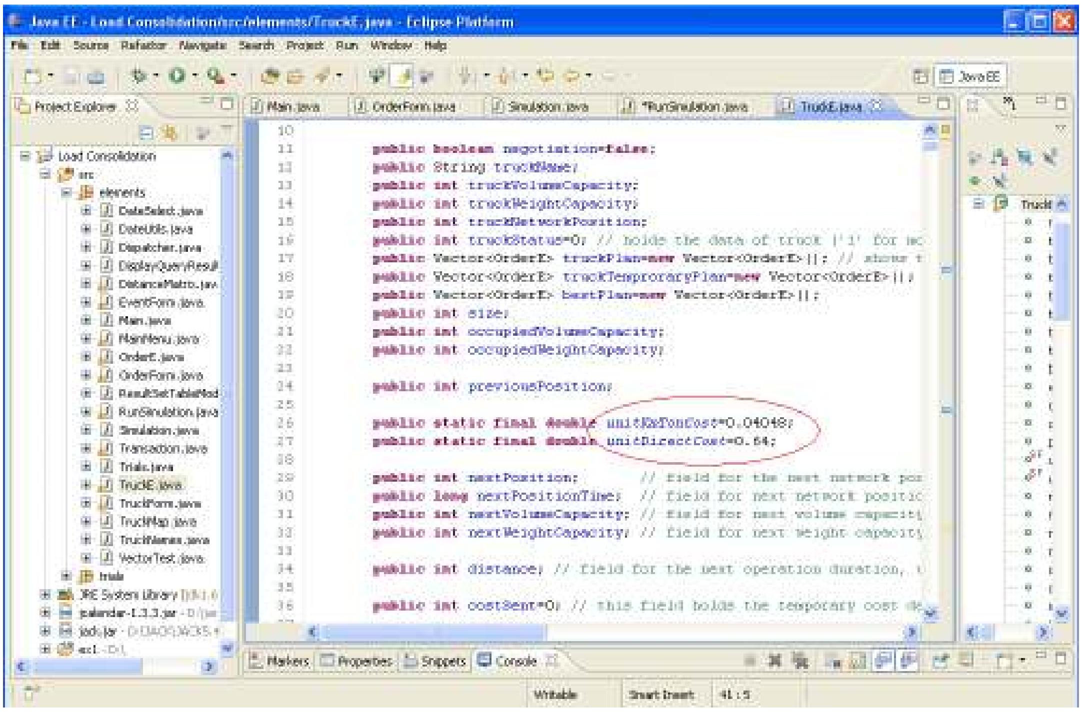
APPENDIX A. SAMPLE TEST PROBLEM

Table A. 1. Solomon test problem R103

CUST NO.	XCOORD.	YCOORD.	DEMAND	READY TIME	DUE DATE	SERVICE TIME
1	35.00	35.00	0.00	0.00	230.00	0.00
2	41.00	49.00	10.00	0.00	204.00	10.00
3	35.00	17.00	7.00	0.00	202.00	10.00
4	55.00	45.00	13.00	0.00	197.00	10.00
5	55.00	20.00	19.00	149.00	159.00	10.00
6	15.00	30.00	26.00	0.00	199.00	10.00
7	25.00	30.00	3.00	99.00	109.00	10.00
8	20.00	50.00	5.00	0.00	198.00	10.00
9	10.00	43.00	9.00	95.00	105.00	10.00
10	55.00	60.00	16.00	97.00	107.00	10.00
11	30.00	60.00	16.00	124.00	134.00	10.00
12	20.00	65.00	12.00	67.00	77.00	10.00
13	50.00	35.00	19.00	0.00	205.00	10.00
14	30.00	25.00	23.00	159.00	169.00	10.00
15	15.00	10.00	20.00	0.00	187.00	10.00
16	30.00	5.00	8.00	61.00	71.00	10.00
17	10.00	20.00	19.00	0.00	190.00	10.00
18	5.00	30.00	2.00	157.00	167.00	10.00
19	20.00	40.00	12.00	0.00	204.00	10.00
20	15.00	60.00	17.00	0.00	187.00	10.00
21	45.00	65.00	9.00	0.00	188.00	10.00
22	45.00	20.00	11.00	0.00	201.00	10.00
23	45.00	10.00	18.00	97.00	107.00	10.00
24	55.00	5.00	29.00	68.00	78.00	10.00
25	65.00	35.00	3.00	0.00	190.00	10.00
26	65.00	20.00	6.00	172.00	182.00	10.00
27	45.00	30.00	17.00	0.00	208.00	10.00
28	35.00	40.00	16.00	37.00	47.00	10.00
29	41.00	37.00	16.00	0.00	213.00	10.00
30	64.00	42.00	9.00	0.00	190.00	10.00
31	40.00	60.00	21.00	71.00	81.00	10.00
32	31.00	52.00	27.00	0.00	202.00	10.00
33	35.00	69.00	23.00	0.00	186.00	10.00
34	53.00	52.00	11.00	37.00	47.00	10.00
35	65.00	55.00	14.00	0.00	183.00	10.00
36	63.00	65.00	8.00	143.00	153.00	10.00
37	2.00	60.00	5.00	41.00	51.00	10.00
38	20.00	20.00	8.00	0.00	198.00	10.00
39	5.00	5.00	16.00	83.00	93.00	10.00
40	60.00	12.00	31.00	44.00	54.00	10.00
41	40.00	25.00	9.00	85.00	95.00	10.00
42	42.00	7.00	5.00	97.00	107.00	10.00
43	24.00	12.00	5.00	31.00	41.00	10.00

44	23.00	3.00	7.00	0.00	185.00	10.00
45	11.00	14.00	18.00	69.00	79.00	10.00
46	6.00	38.00	16.00	32.00	42.00	10.00
47	2.00	48.00	1.00	0.00	184.00	10.00
48	8.00	56.00	27.00	0.00	185.00	10.00
49	13.00	52.00	36.00	0.00	192.00	10.00
50	6.00	68.00	30.00	108.00	118.00	10.00
51	47.00	47.00	13.00	0.00	203.00	10.00
52	49.00	58.00	10.00	0.00	193.00	10.00
53	27.00	43.00	9.00	0.00	208.00	10.00
54	37.00	31.00	14.00	95.00	105.00	10.00
55	57.00	29.00	18.00	0.00	197.00	10.00
56	63.00	23.00	2.00	136.00	146.00	10.00
57	53.00	12.00	6.00	130.00	140.00	10.00
58	32.00	12.00	7.00	0.00	196.00	10.00
59	36.00	26.00	18.00	200.00	210.00	10.00
60	21.00	24.00	28.00	0.00	202.00	10.00
61	17.00	34.00	3.00	0.00	201.00	10.00
62	12.00	24.00	13.00	0.00	194.00	10.00
63	24.00	58.00	19.00	58.00	68.00	10.00
64	27.00	69.00	10.00	0.00	185.00	10.00
65	15.00	77.00	9.00	73.00	83.00	10.00
66	62.00	77.00	20.00	51.00	61.00	10.00
67	49.00	73.00	25.00	127.00	137.00	10.00
68	67.00	5.00	25.00	83.00	93.00	10.00
69	56.00	39.00	36.00	142.00	152.00	10.00
70	37.00	47.00	6.00	50.00	60.00	10.00
71	37.00	56.00	5.00	182.00	192.00	10.00
72	57.00	68.00	15.00	0.00	180.00	10.00
73	47.00	16.00	25.00	0.00	197.00	10.00
74	44.00	17.00	9.00	0.00	199.00	10.00
75	46.00	13.00	8.00	149.00	159.00	10.00
76	49.00	11.00	18.00	0.00	192.00	10.00
77	49.00	42.00	13.00	73.00	83.00	10.00
78	53.00	43.00	14.00	179.00	189.00	10.00
79	61.00	52.00	3.00	96.00	106.00	10.00
80	57.00	48.00	23.00	92.00	102.00	10.00
81	56.00	37.00	6.00	182.00	192.00	10.00
82	55.00	54.00	26.00	0.00	192.00	10.00
83	15.00	47.00	16.00	0.00	196.00	10.00
84	14.00	37.00	11.00	0.00	198.00	10.00
85	11.00	31.00	7.00	101.00	111.00	10.00
86	16.00	22.00	41.00	0.00	196.00	10.00
87	4.00	18.00	35.00	0.00	184.00	10.00
88	28.00	18.00	26.00	93.00	103.00	10.00
89	26.00	52.00	9.00	74.00	84.00	10.00
90	26.00	35.00	15.00	0.00	211.00	10.00
91	31.00	67.00	3.00	0.00	187.00	10.00
92	15.00	19.00	1.00	0.00	194.00	10.00
93	22.00	22.00	2.00	18.00	28.00	10.00
94	18.00	24.00	22.00	188.00	198.00	10.00
95	26.00	27.00	27.00	0.00	207.00	10.00
96	25.00	24.00	20.00	0.00	205.00	10.00
97	22.00	27.00	11.00	0.00	204.00	10.00
98	25.00	21.00	12.00	0.00	202.00	10.00
99	19.00	21.00	10.00	0.00	198.00	10.00
100	20.00	26.00	9.00	83.00	93.00	10.00
101	18.00	18.00	17.00	185.00	195.00	10.00

APPENDIX B. COST PARAMETER SETTING



```
        public boolean negotiation=false;
        public String truckName;
        public int truckVolumeCapacity;
        public int truckWeightCapacity;
        public int truckNetworkPosition;
        public int truckStatus=0; // holds the data of truck |'1' for mo
        public Vector<OrderE> truckPlan=new Vector<OrderE>(); // shows t
        public Vector<OrderE> truckTemproraryPlan=new Vector<OrderE>();
        public Vector<OrderE> bestPlan=new Vector<OrderE>();
        public int size;
        public int occupiedVolumeCapacity;
        public int occupiedWeightCapacity;

        public int previousPosition;

        public static final double unitKmTonCost=0.04048;
        public static final double unitDirectCost=0.64;

        public int nextPosition;        // field for the next network pos
        public long nextPositionTime;  // field for next network positic
        public int nextVolumeCapacity; // field for next volume capacit
        public int nextWeightCapacity; // field for next weight capacit

        public int distance; // field for the next operation duration, u

        public int costSent=0; // this field holds the temporary cost de
```

Figure B. 1. Direct and indirect cost of truck agent parameters set view

APPENDIX C. TRANSPORTATION DISTANCE MATRIX

Table C. 1. Distance matrix of the transportation network

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	Belgium	England	France	Germany	Greece	Holland	Iran	Ireland	Italy	Norway	Poland	Russia	Spain	Sweden	Turkey	Ukraine
Belgium	0	518	655	540	2485	222	6042	1164	1468	1552	1185	2300	1546	1713	3842	2234
England		0	933	990	2965	615	6517	660	1948	1955	1757	2735	1825	2116	4317	2679
France			0	1031	2337	852	6247	1578	1321	2244	1833	3681	1084	2362	4047	2681
Germany				0	2397	482	5656	1633	1381	1285	765	2746	2040	1436	3456	1746
Greece					0	2644	4218	3608	1161	3658	2082	2780	3129	3691	2018	1780
Holland						0	4800	1257	1616	1358	1032	3119	1742	1519	2600	2119
Iran							0	7167	5071	6815	5526	5700	7041	6765	2200	4728
Ireland								0	2599	2605	2260	4329	2474	2766	4967	3329
Italy									0	2642	1925	3381	1974	2674	2871	2381
Norway										0	1690	3782	3091	791	4615	2782
Poland											0	2129	2724	1637	3326	1129
Russia												0	2100	2900	3500	1100
Spain													0	3256	4841	3689
Sweden														0	4565	1989
Turkey															0	2528
Ukraine																0

APPENDIX D. SIMULATION PARAMETER SET

```
package elements;

import java.util.Date;
import java.util.Random;

import order.Order;

class RunSimulation implements Runnable
 {
    int runTime=30;
    int sleepTime=1000;
    public long tiRes; // is is used as second in the Jack kernel
    private static Random generator      = new Random(20);
    private String threadName; // name of thread
    public int orderNo;
    public int fromPoint;
    public int toPoint;
    public int volume;
    public int weight;
    public Date pickUpDate;
    public Date deliveryDate;
    public int truckAmount=5;
    public String truckName;
    public int truckVolumeCapacity=98;
    public int truckWeightCapacity=22;
    public int truckNetworkPosition;


     public RunSimulation( String name )
     {
        threadName = name;
     }



      public void run()
     {

        try // put thread to sleep for sleepTime amount of time
        {
           System.out.printf( "%s going to sleep for %d
milliseconds.\n", threadName, sleepTime );

           for (int i=0; i<truckAmount; i++)
           {
             truckName="Truck "+i;
             TruckE t=new TruckE(truckName, truckVolumeCapacity,
truckWeightCapacity,
             truckNetworkPosition);
             truck.Truck a=new truck.Truck(t.getTruckName(),t);
                   MainMenu.numberOfTruck++;
                   Main.vector.add(t);
```

```
                  }


           for (int i=0; i<runTime; i++)
           {

               int fromNodes[]=
{0,1,2,3,4,5,6,7,8,9,10,11,12,13,15,14,14,

14,14,14,14,14,14,14,14,14,14,14,14,14,14,14,14,14,14,

14,14,14,14,14,14,14,14,14,14,14,14,14,14,14};
               int toNodes[]=   {0,1,2,3,4,5,6,7,8,9,10,11,12,13,15

,14,14,14,14,14,14,14,14,14,14,14,14,14,14,14,14,14

,14,14,14,14,14,14,14,14,14,14,14,14,14,14,14,14,14,14};

               tiRes=generator.nextInt( 3 );
               fromPoint=fromNodes[generator.nextInt( 50 )];
               toPoint=toNodes[generator.nextInt( 50 )];
               orderNo=i;

                  while (fromPoint==toPoint)
                  {
                    toPoint=generator.nextInt( 3 );
                  }

                        volume=1+generator.nextInt( 30 );
                        weight=1+generator.nextInt( 16 );;


                       int tiADV=generator.nextInt( 20000 );
                       int tiSLK =generator.nextInt( 40000 );

                  long
pickUp=tiRes*1000+System.currentTimeMillis()+tiADV;
                  long
deliver=pickUp+DistanceMatrix.distance(fromPoint, toPoint)+tiSLK;
                  Date pickUpDate=new Date(pickUp);
                  deliveryDate=new Date(deliver);

                  OrderE order=new OrderE(orderNo,fromPoint,
toPoint, volume, weight,tiRes, pickUpDate, deliveryDate);

                        System.out.printf("\n%s\n\n%s %s\n%s %s\n%s
%s\n%s %s\n%s %s\n%s " +
                              "%s\n%s %s \n\n","New Order is created
with attributes: ","No:",

      order.getNo(),"From:",order.getFrom(),"To:",order.getTo(),"Vol
ume:",order.getVolume(),
                              "Weight:",order.getWeight(), "Pick up
date: ",order.getPickUpDate(),"Delivery date: ",
                              order.getDeliveryDate());

                        Order o=new Order(""+order.getNo());  //
Order agent creation

                        o.submitTransportRequest(order);
                 Thread.sleep( sleepTime ); // put thread to sleep
```

```
            }
         } // end try

       catch ( InterruptedException exception )
       {
          exception.printStackTrace();
       } // end catch

       System.out.printf( "%s done sleeping\n", threadName );
    } // end method run
} // end class PrintTask
```

CURRICULUM VITAE

PERSONAL INFORMATION

Surname, Name: Kaplanoğlu, Vahit

Nationality: Turkish (TC)

Date and Place of Birth: 1 June 1982 , Gaziantep

Marital Status: Married with two children

Phone: +90 533 629 97 03

Fax: +90 342 360 43 83

email: kaplanoglu@gantep.edu.tr

EDUCATION

Degree	Institution	Year of Graduation
MS	University of Gaziantep (Ind. Eng.)	2007
BS	University of Marmara (Ind. Eng.)	2004
High School	Sunguroğlu Erkek Lisesi	1999

WORK EXPERIENCE

Year	Place	Enrollment
2004- Present	University of Gaziantep	Research Assistant

FOREIGN LANGUAGES

English

German

PUBLICATIONS

International Journals

Baykasoğlu, A., Kaplanoğlu, V., Erol, R., Şahin, C. (2010) A multi-agent framework for load consolidation in logistics, *Transport Journal*, Accepted for publication.

Baykasoğlu, A., Kaplanoğlu, V. and Unutmaz, Z. D. (2010) 'Utilizing Prometheus design tool for truck load consolidation decisions', *International. Journal of Information Systems and Supply Chain Management* , Accepted for publication.

Baykasoğlu, A. and Kaplanoğlu, V. (2010) 'Evaluating the basic load consolidation strategies for a transportation company through logistics process modeling and simulation', *International Journal of Data Analysis Techniques and Strategies*, Accepted for publication.

Baykasoglu, A., Durmusoglu, Z.D.U Kaplanoglu, V., (2009), Training Fuzzy Cognitive Maps via Extended Great Deluge Algorithm with Applications, *Computers in Industry*, Accepted for publication.

Baykasoğlu, A., Kaplanoğlu, V., 2008, Application of activity based costing to a land transportation company: a case study, *International Journal of Production Economics*, Vol. 116, Issue 2, 308-324.

Baykasoğlu, A., Kaplanoğlu, V.,2007, A service-costing framework for logistics companies and a case study, *Management Research News*, Vol. 30, Issue 9, pp. 621-633.

Baykasoğlu, A, Kaplanoğlu, V., 2006, Developing a Service Costing System and an Application for Logistics Companies, *International Journal of Agile Manufacturing*, Vol. 9, Issue 2, pp.13-18.

National Conferences

Baykasoğlu, A., Kaplanoğlu, V., Logistics process modeling for evaluating basic load consolitation strategies: a case study, YA/EM 2009: Yöneylem Araştırması / Endüstri Mühendisliği Kongresi 29. Ulusal Kongresi, Bilkent Üniversitesi, 22-24 Haziran 2009, Ankara, Bildiriler Kitabı CD'sinde, 2009.

Baykasoğlu, A., Kaplanoğlu, V. and Unutmaz, Z. D., An agent based framework for truck load consolidation, YA/EM'2008: Yöneylem Araştırması / Endüstri Mühendisliği Kongresi XXVIII, June 30, July 01-02, 2008, Galatasaray University, Istanbul, Turkey, pp. in CD

Baykasoğlu, A. and Kaplanoğlu, V., Süreç modelleme benzetimi ile faaliyet tabanlı maliyetlendirme ve lojistik hizmeti üreten bir işletmeye uygulanması, ÜAS'2007: Ulusal Üretim Araştırmaları Sempozyumu, November 15-17, 2007, Gazi University, Ankara, Turkey, pp.??-??

Baykasoğlu, A., Kaplanoğlu, V. and Unutmaz, Z. D., Lojistik firmalarında yük konsolidasyonu için bir model önerisi, YA/EM 2007: Yöneylem Araştırması ve Endüstri Mühendisliği 27. Ulusal Kongresi, July 02-04, 2007, İzmir, Turkey, pp. in CD

Baykasoğlu, A. and Kaplanoğlu, V., Kalite maliyet modeli ve bir lojistik işletmesindeki analizi, MARKATEK-2007: Marka, Kalite ve Teknoloji Yönetimi Sempozyumu, May 18-19, 2007, MMO Gaziantep Şubesi, Gaziantep, Turkey, pp. 367-373

Baykasoğlu, A. and Kaplanoğlu, V., Faaliyet tabanlı maliyetlendirmenin bir lojistik işletmesine uygulanması, GAP V. Mühendislik Kongresi, April 26-28, 2006, Şanlıurfa, Turkey, pp. 182-189

International Conferences

Baykasoğlu, A., Göçken, T., Kaplanoğlu, V., A fuzzy approach to prioritize project activities, FUZZYSS'09: 1st International Fuzzy Systems Symposium, (ed., Baykasoğlu A., Dereli, T., Turksen, IB) TOBB Economy and Technology University, October 1-2, 2009, Ankara, Turkey, pp. 118-124.

Baykasoğlu, A., Durmuşuğlu, UZD., Kaplanoğlu, V., Training fuzzy cognitive maps by extended great deluge algorithm, FUZZYSS'09: 1st International Fuzzy Systems Symposium, (ed., Baykasoğlu A., Dereli, T., Turksen, IB) TOBB Economy and Technology University, October 1-2, 2009, Ankara, Turkey, pp. 110-117.

Baykasoğlu, A. and Kaplanoğlu, V., Logistics Process Modeling for Effective Decision Making: A Case Study, 6th International Logistics and Supply Chain Congress, November 06-07, 2008, Istanbul, Turkey, pp. 379-384

Baykasoğlu, A. and Kaplanoğlu, V., Utilizing business process modeling & simulation to analyze effect of capacity on transportation performance for a logistics company, IMS'2008: 6th International Symposium on Intelligent and Manufacturing Systems, October 14-17, 2008, Sakarya, Turkey, pp.??-??

Baykasoğlu, A., Unutmaz, Z. D. and Kaplanoğlu, V., A truck load consolidation approach for a logistics company, 5th International Logistics and Supply Chain Congress, November 08-09, 2007, Istanbul, Turkey, pp. 606-613

Baykasoğlu, A. and Kaplanoğlu, V., Application of the balanced scorecard concept to a logistics company, 4th International Logistics and Supply Chain Congress, November 29-30, December 01, 2006, Izmir, Turkey, pp. 773-777

Baykasoğlu, A. and Kaplanoğlu, V., Application of business process modeling and simulation to a logistics company, AMSE'06: International Conference on Modelling and Simulation, August 28-30, 2006, Konya, Turkey, pp. 977-982

Baykasoğlu, A. and Kaplanoğlu, V., Developing service costing system for logistic companies and an application, ICAM 2006: International Conference on Agile Manufacturing, July 19-20, 2006, Norfolk, Virginia, USA, pp.??-??

RESEARCH PROJECTS

Developing a Simulation and Activity Based Costing System for SME's and an Application for a Logistics Company. "Kobi' ler İçin Benzetim ve Faaliyet Tabanlı Maliyetlendirme Sistemi Geliştirilmesi ve Bir Lojistik İşletmesine Uygulanması", (Gaziantep Üniversitesi Araştırma Fonu 01-06-2007, MF-07-04)

MEMBERSHIPS

ISAM- International Society of Agile Manufacturing

SCHOLARSHIPS and FELLOWSHIPS

TÜBİTAK BİDEB 2211 - *National Ph.D Scholarship Programme* - During the Ph.D studies

TÜBİTAK BİDEB 2214 - *International Research Fellowship Programme* -For visiting RMIT University Computer Science and IT Department Agent Research Group in *Australia* in context of Ph.D studies

HOBBIES

Swimming, playing guitar and table tennis