The Pennsylvania State University

The Graduate School

Department of Industrial and Manufacturing Engineering

SIMULATION AND CONTROL OF DISTRIBUTED MULTI-AGENT BASED LOGISTICS SYSTEMS

A Thesis in

Industrial Engineering

by

Vincent William Slaugh

Submitted in Partial Fulfillment
of the Requirements
for the Degree of

Master of Science

August 2009

The thesis of Vincent W. Slaugh was reviewed and approved* by the following:

Soundar R. T. Kumara
Allen E. Pearce/Allen M. Pearce Professor of Industrial Engineering
Thesis Advisor

Christopher Byrne
Research Associate and Assistant Professor of Mathematics

M. Jeya Chandra
Professor of Industrial Engineering
Graduate Program Coordinator

*Signatures are on file in the Graduate School

ABSTRACT

This thesis investigates logistics systems in which distributed entities cooperate to serve demands by allocating supplies and transportation resources. In particular, a simulation framework, the Distributed Resource Allocation Planning and Execution Simulation (DRAPES), was designed and implemented as a distributed discrete-event simulation using agents. Using an agent-based approach, the simulation's purpose is to determine how local decision rules and processes followed by an agent affect the global performance of the system. Agents represent the distributed decision making entities, or units, of the logistics system, and have clearly defined information sensors, communication channels, local knowledge, and actions to change the state of the system. Also, computationally, the agent-based approach helped to design a successful simulation that has potential in the future to work with emulated software components of a real logistics system.

Simple rules from two different control methods were tested on the simulation. First, a decentralized approach follows a process of iterative bidding based on demands' priorities to determine which unit will fulfill which demand. Second, in a centralized approach, the distributed units send reports to a centralized controller, which in turn assembles the information, solves a mixed-integer program, and distributes the results as orders to the rest of the system. Simulation runs were made using scenarios that tested not only each approach's ability to generate an initial plan but also its ability to re-plan as the system state changed. The centralized control approach was found to consistently fulfill more demands, and achieve a better system utility than the decentralized approach.

Because of computational limitations for solving the mixed-integer program, future work is recommended to improve each of these control methods and consider how a hybrid approach could be implemented.

This work was designed and implemented to simulate a real-life logistics system. To protect the proprietary aspects of this system, we use generalized and fictitious names to represent the real-world scenarios.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

Deep gratitude is expressed to Dr. Soundar Kumara, who advised me on this thesis and made my first real foray into research an exciting and meaningful experience. His help and suggestions have taught me much about research methods and multi-agent systems, and his advice will surely have an impact on me personally and academically for years to come. Dr. Christopher Byrne served as a reader for this thesis, and I appreciate his time and effort.

I also wish to thank Dan Behringer, my supervisor on a project related to this thesis. I thoroughly enjoyed our discussions, and his expertise contributed greatly to shaping the simulation. Furthermore, colleagues Matt Rigdon and Shaun Lichter were a pleasure to work with on the mathematical model, and I thank them for their invaluable contributions. Finally, I thank my wife, Lindsey, whose support and love as I worked on this thesis have given me great joy.

Chapter 1

Introduction

We consider a system in which entities distributed across a network allocate supply and transportation resources to serve demands. Applications such as military mission logistics and humanitarian relief depend on such systems of distributed units to function. Because of communication restrictions or an abundance of data, these units must make decisions to change the system's state based on their local knowledge. Although this can make the system more robust and adaptable, a global view of how the system operates is hard to predict and understand.

Two research areas would be of importance to address this system. First, a simulation model is needed to understand the system's structure and test its performance. Each unit's sensed information, local knowledge, and available actions need to be clearly identified. From specifying the system's rules at a local level, its global performance can then be observed.

Second, different types of control logic need to specified and tested for how they can be implemented in a distributed environment. More centralized control methods supply a central source with information in an attempt to give it a global view of the system. From this data, it can issue orders to the system's distributed units, which exhibit little planning intelligence on their own. Alternately, more decentralized control methods allow for having no central control source, and units formulate their own plans based on their local knowledge and interaction with other units. By using different control

strategies in the simulation experimentation, observations can then be made about the strengths and weaknesses of each approach.

The system's purpose is to serve a set of demands, and those demands exhibit characteristics and requirements relevant to how simulation and control approaches are designed. In this system, each demand is represented by a requisition, which is circulated among units to find a source. Every supply shipped or consumed is tied to a specific requisition, not a demand aggregate or a forecast of future demand. This ensures that the system is responsive to demands with different priorities—a key characteristic of each demand—which are based solely on the priority of the task to which they belong. Two demands for equal quantities of the same commodity by the same unit can even have different priorities, if the two corresponding tasks have different priorities. The demand's priority is used as the primary determinant of how resources are allocated. The use of requisitions also allows for tracking the planned and actual shipments of supplies across the logistics distribution network.

Furthermore, demands are subject to physical constraints. The most basic physical constraint is the carrying capacity of the transportation asset assigned. The shipment for a demand requires a certain amount of cargo space based on the demand quantity and commodity. Also, demand attributes are subject to change over time or variation. Priority, quantity, and due date are all demand attributes that could potentially change over time. In this thesis, dynamic demand priorities and unforeseen consumption of supplies are examined. Finally, new demands may arise at random over time. These are immediately associated with a unit, which is then responsible for finding a source for that demand.

The objective of this research is to design and implement a distributed agent-based simulation for solving the distributed logistics problem with centralized and decentralized controllers. The problem is more formally defined in Chapter 2, with the structure of the system and the means available to change the state of the system also specified. Chapter 3 reviews relevant literature and categorizes different approaches, comparing the strengths and weaknesses of their methods and assumptions. Chapter 4 describes the simulation system and decision logic—decentralized and centralized—designed to address the problem. The simulation system is designed using the idea of multi-agent systems, and control approaches rely on simple decentralized rules and a mathematical programming model. Chapter 5 lists the experimentation used to evaluate control approaches. Results of experimentation are presented in Chapter 6. Finally, Chapter 7 offers conclusions and recommendations.

Chapter 2

Problem Description and Methodology

The relevant problem for this thesis is the allocation of supplies and transportation resources by distributed decision-making entities to satisfy a set of demands. Each demand in the system is characterized by a quantity of a single commodity, early and late required delivery dates, and a priority. The logistics system can be decomposed by its organization units that operate autonomously in a distributed environment. These units can have one of two similar roles. Operational units (OUs) have demands and communicate them to the rest of the system with the goal of finding a supplier. OUs keep and execute plans, and also have an on-hand inventory of supplies which they can allocate to demands. Another type of unit, the distribution center (DC), generates no demands of its own, but it keeps and execute plans to send shipments of supplies to OUs.

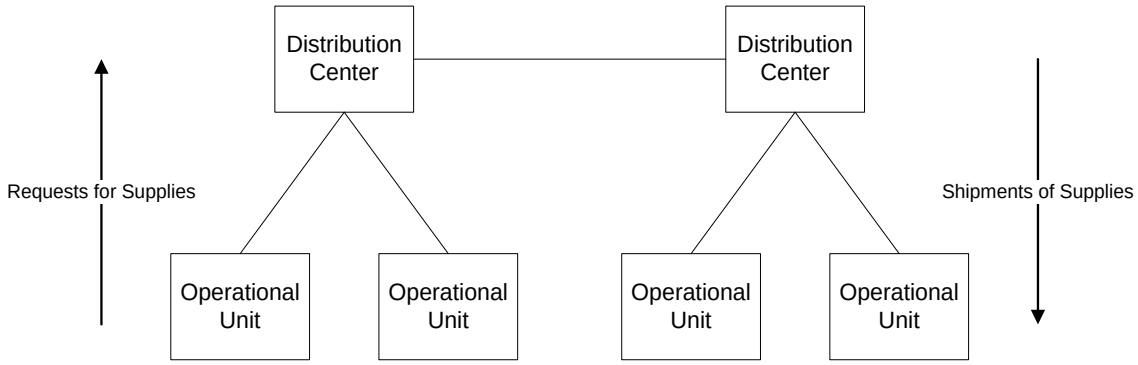

Figure **2-1**: System structure.

Even though the units exist in a hierarchy, as shown in Figure **2-1**, they exhibit autonomy in how they keep and change their state as distributed units. Because of the distributed nature of the problem, units must communicate through specific channels and messages. No unit may directly force another unit to allocate a resource. Instead, it must send a message, which the receiving unit processes and interprets. Also, the issue of real-time planning and re-planning is another important aspect of the system. Each unit in the system engages in its own planning and execution in real-time. It must re-plan based on changes to the system, newly sensed information, or deviations to the plan that occur as it is executed.

In addition to the system's restrictions as a distributed system, it is constrained by the scarcity of both supply and transportation resources. Its goal is to efficiently allocate resources to fulfill demands. The demand's priority and the timeliness of the receipt of supplies play a dominant role in determining the quality of the solution. It is also desired to minimize the amount of transportation provided, the reliance on lateral transshipments from distribution centers to operational units that do not have a direct superior-subordinate relationship, and the changeability of units' plans.

2.1 Assumptions

2.1.1 Network Topology

The network considered in this thesis is a multiechelon (serial) distribution system with a set U of locations—each known as a unit—that control the allocation of supply

and transportation resources. Except for suppliers in the top echelon, each unit has a primary supplier $i = \rho(j),\ i,j \in U$. Sites with no successors are referred to as *operational units*, and their immediate suppliers are *distribution centers*.

Through a policy of lateral transshipments, any distribution center can supply any operational unit, although its preferred supplier is still its immediate superior in the supply chain. This thesis is only concerned with two-level systems of inventory locations, but the principles exhibited here can be extended to higher echelon systems.

2.1.2 Tasks and Demands

The purpose of the supply chain is to serve a set N of tasks, each containing demands for all commodities of the set K. Each task is assigned to a single unit i, $i \in U$, such that unit i is an operational unit. Other characteristics of a task include the times e_j and l_j, which are the earliest and latest required delivery times for task $j \in N$, respectively. Also, task j is associated with a relative priority p_j, $0 \leq p_j \leq 1$, $j \in N$. Penalties are incurred for a demand's fulfillment outside of this interval and are weighted by priority. Each demand has a quantity q_{jk} associated with it, $j \in N$, $k \in K$.

The time at which a unit becomes aware of a task is the time w_j, $j \in N$. If $w_j = 0$, the task is known to units at time $t = 0$ and is part of the initial batch of demands. Otherwise, the demand is made known to its unit during execution.

Tasks arrive to each unit i during execution as a Poisson Process with rate λ_i, $i \in U$. The probability that task j contains a quantity $q_{jk} > 0$ is w_k. If $q_{jk} > 0$, demand

quantity q_{jk} is exponentially distributed with mean $\bar{q}_k$ for task j and supply type k, $j \in N$, $k \in K$. Task priorities p_j are uniformly distributed between 0 and 1.

2.1.3 Exogenous Resupply

Each unit i receives an exogenous resupply of a fixed quantity Q_{ik} for supply type k at fixed intervals of length R, $i \in U$, $k \in K$. Therefore, resupplies for each unit occur at times $t = \Delta R$, $\Delta = 1,2, \ldots$. When these events occur, the corresponding quantities of supplies are added to the receiving unit's on-hand inventory.

2.1.4 Transportation

Each unit i controls a fleet of k_i service units, $i \in U$. The space required for a unit of supply type k is j_k, $k \in K$. The travel time from unit i to unit j is T_{ij}, $i, j \in U$.

At any time during the simulation, a unit may allocate supplies from its inventory to send to another unit or consume internally. If sending to another unit, the unit must also allocate available and sufficient transportation resources for the transfer of supplies to take place. If both transportation and supply resources are allocated by unit i to be sent to unit j at time t, unit j adds the appropriate supplies to its inventory at time $t + T_{ij}$. The transportation resources are again available to unit i at time $t + T_{ij} + T_{ji}$.

In this system, operational units do not have transportation capabilities. Only one type of transportation resource exists, and it can only exist in integer values. Supplies for

multiple demands can be assigned to one shipment, but the fulfillment of a demand cannot be split into multiple shipments.

2.1.5 System Time and Simulation Duration

In this system, time is continuous and begins at t_0. The simulation's end time is t_f. Time is a continuous variable for the simulation. Agents plan only for the duration of the simulation.

2.1.6 Unit Plans

Each unit i maintains a list of all events P^{ik} for each supply type k in the future that will affect its inventory level, $i \in U, k \in K$. These events include exogenous resupplies, expected receipts, and expected allocations. P^{ik} contains L^{ik} elements with attributes for change q_l^{ik} in inventory level associated with that event, time r_l^{ik} at which the event occurs, and priority p_l^{ik} associated with that event, $l \in P^{ik}, j \in N, k \in K$.

2.1.7 Inventory Level over Time

A unit's state $x(t) = \{x_k(t) \forall k\}$ at time t is its inventory level for each supply type. Each unit begins with an initial inventory $x_k(t_s) = B_k$. The change of a unit's inventory over time is based on the plan events executed by units in the system. Assuming that the system executes according to a feasible plan, it follows the equation

$$x_k(t) = B_k + \sum_{m:\, t_s \le r_l^{ik} \le t}^{P^{ik}} q_l^{ik} \qquad \textbf{2.1}$$

2.1.8 Global Utility Function

System performance is evaluated based on three criteria to be minimized. First, each demand is evaluated based on whether it is not fulfilled, fulfilled early, fulfilled on time, or fulfilled late. If the demand for commodity k of task j was fulfilled within the delivery window, let $a_{jk} = 0, j \in N, k \in K$. It was fulfilled outside the delivery window, a_{jk} is time difference between time at which it was fulfilled and the nearest point of the delivery window. If it was not fulfilled, $a_{jk} = A$, a constant representing a time early or late that corresponds to an equivalent penalty for being unfulfilled.

Second, all transportation resources' times spent in transportation are summed. Let b_i be the total time spent delivering supplies for all transportation resources of unit $i, i \in U$. Indirectly, this will penalize fulfillments by distribution centers that are not an operational unit's direct superior in the unit hierarchy, as the travel times between units without a superior-subordinate relationship will be defined to be longer.

Third, the changeability of a plan over the duration of the simulation is assessed based on the number of times c_{jk} that the unit sourcing the demand for commodity k of task j changes, $j \in N, k \in K$. This intentionally penalizes changes in units' plans as they are executed.

Combining these three components, system's goal is to minimize its objective function z as the system operates over time. The three components are weighted by constants α, β, and γ, respectively. Mathematically, the objective function is

$$z = \alpha \sum_{j \in N} \sum_{k \in K} p_j \left(a_{jk}\right)^2 + \beta \sum_{i \in U} b_i + \gamma \sum_{j \in N} \sum_{k \in K} c_{jk} \qquad \textbf{2.2}$$

2.2 Solution Scheme

The solution to the problems posed by this system is to design a set of rules for allocating supply and transportation resources. Each unit—entities with clear decision-making roles classified as operational units and distribution centers—in the system is modeled as an agent with the capability to be an autonomous decision maker. That is, each unit possesses the power to change the state of the system by allocating resources from its inventory and dispatching transportation assets from its fleet. Units communicate with other units through messages, and follow a set of business rules for determining which unit should fulfill each task.

The various strategies for sharing information and allocating resources will be tested using a distributed discrete event simulation. Each rule is based on a snapshot of a current state of the system or sub-system, given global or local knowledge and knowledge about the future. The main aspect differentiating the approaches is the degree to which decision making is centralized in the system. In more centralized approaches, operational units report information to superior units in the hierarchy, and receive orders for allocating resources in return. In decentralized approaches, units demonstrate more

independence in making decisions, negotiating with other units and ultimately deciding who should send resources to whom at what time.

Chapter 3

Review of Literature

In this section, we review approaches to solving the problem of a multi-echelon supply chain in which transportation assets and supply resources are allocated to fulfill demands. In general, mathematical programming approaches have focused on centralized optimization and decentralized decision making in transportation networks and optimal inventory policies. Agent-based approaches, on the other hand, have focused primarily on system design for complex and dynamic transportation systems. Each approach has its own advantages and disadvantages for solving the problem described in this thesis.

3.1 Centralized Network Flow Problems

Three examples from literature on distribution logistics for disaster relief scenarios provide approaches to solving a related problem from the perspective of centralized control. Compared to other approaches, these approaches are characterized by more restrictive assumptions, an inability to replan in real time based on variations and perturbations, and a less well-defined path to implementation. However, they come much closer to an optimal solution from a global perspective.

Haghani & Oh (1996) formulate the problem as a multi-commodity, multi-modal network flow problem. Their model accounts for capacity constraints on nodes and arcs,

and allows for transfers between transportation modes during an operation. The problem is formulated as a single objective linear program on a spatio-temporal network. Vehicles are modeled based on the quantity of vehicles of a mode at a node at a certain time. Two solution algorithms were proposed and tested against the linear program for artificial data sets. For the first algorithm, the model was decomposed by constraints that linked vehicle and commodity flows. The second algorithm gradually fixes integer flow variables until all were fixed after a number of iterations.

This methodology provides a near-optimal solution to a problem faced by a decision maker with a global perspective and the ability to execute global decisions. It also helps efficiently allocate the resources of a multi-modal transportation network. Compared to the problem described in this thesis, this view focuses more on an operational than a tactical perspective in that flows of resources over a transportation network do not transfer directly into tactical plans for specific transportation resources. In addition, it is unclear how variations and perturbation dynamics would affect the quality of the plan. **As this formulation is the most directly related to the one used for centralized control in this thesis, it is reproduced here for completeness (Haghani & Oh, 1996).**

Sets

N	Set of nodes, $i, j \in N$ are indices ($U \cup UV \cup V \cup VW = N$)
U	Set of pure origin nodes, $u \in U$ is an index
UV	Set of origin nodes that also have a role of transshipment nodes, $uv \in UV$ is an index

V	Set of transshipment nodes, $v \in V$ is an index
VW	Set of destination nodes that also have a role as transshipment nodes, $vw \in VW$ is an index
W	Set of destination nodes, $w \in W$ is an index
A	Set of links in the physical network, $a \in A$ is an index
M	Set of modes, $m \in M$ is an index
G	Set of goods or commodities, $g \in G$ is an index

Parameters

EPT_{gi}	Earliest pick-up time period of commodity g at node i
SE_{git}	Amount of exogenous supply of commodity g at node i at time period t
EDT_{gi}	Earliest delivery time period of commodity g at node i
DE_{git}	Amount of exogenous demand of commodity g at node i at time period t
YE_{it}^{m}	Number of vehicles of mode m which are available at node i at time period t
YCA^{m}	Vehicle capacity of mode m
$ACA_{itj\,t'}^{m}$	Arc capacity between node i at time period t and node j at time period t' in number of vehicles for mode m
$CVR_{itj\,t'}^{m}$	Unit cost of moving vehicle of mode m from node i at time period t and node j at time period t'

$CGR_{itj\,t'}^{gm}$ Unit cost of shipping commodity g by mode m from node i at time period t and node j at time period t'

CSC_{git} Unit cost for carrying over the supply for commodity g at node i from time period t to time period $t+1$

CDC_{git} Unit cost for carrying over the demand for commodity g at node i from time period t to time period $t+1$

$CGT_{it(t+Kmm')}^{gmm'}$ Unit cost of transfer of commodity g from mode m to mode m' at node i at time period t. Kmm' represents the number of time periods required fro this transfer.

Variables

$Y_{itj\,t'}^{m}$ Flow of vehicles of mode m from node i at time period t and node j at time period t'

YC_{it}^{m} Number of vehicles of mode m which is carried over from time period t to time period $t+1$ at node i

$X_{itj\,t'}^{gm}$ Flow of commodity g by mode m from node i at time period t and node j at time period t'

SC_{git} Amount of supply of commodity g which is carried over from time period t to time period $t+1$ at node i

DC_{git} Amount of demand of commodity g which is carried over from time period t to time period $t+1$ at node i

$XT_{it(t+Kmm')}^{gmm'}$ Amount of commodity g which is transferred from mode m to mode

m' at node i at time period t

SE_{git}^{m} Amount of exogenous supply of commodity g assigned to mode m at node i at time period t

SC_{git}^{m} Amount of supply of commodity g which is carried over by mode m from time period t to time period $t+1$ at node i

DE_{git}^{m} Amount of exogenous demand of commodity g delivered by mode m at node i at time period t

DC_{git}^{m} Amount of demand of commodity g which is carried over by mode m at time period t to time period $t+1$ at node i

Objective

$$\begin{aligned}\text{Minimize } & \sum_{i}\sum_{j}\sum_{t}\sum_{t'}\sum_{m} CVR_{itj\,t'}^{m} \times Y_{itj\,t'}^{m} \\ & + \sum_{i}\sum_{j}\sum_{t}\sum_{t'}\sum_{g}\sum_{m} CGR_{itj\,t'}^{m} \times X_{itj\,t'}^{m} \\ & + \sum_{i\in U,UV,V}\sum_{t}\sum_{g} CSC_{git} \times \left(\sum_{m} SC_{git}^{m}\right) \\ & + \sum_{i\in VW,W}\sum_{t}\sum_{g} CDC_{git} \times \left(\sum_{m} DC_{git}^{m}\right) \\ & + \sum_{i\in UV,V}\sum_{t}\sum_{g}\sum_{m}\sum_{m'} CGT_{it(t+Kmm')}^{gmm'} \times XT_{it(t+Kmm')}^{gmm'}\end{aligned} \quad \textbf{3.1}$$

Constraints

$$\begin{aligned}\sum_{j}\sum_{t'} X_{jt'it}^{gm} + \sum_{m} XT_{i(t-Kmm')t}^{gm'm} + SC_{gi(t-1)}^{m} + SE_{git}^{m} \\ = \sum_{j}\sum_{t'} X_{itjt'}^{gm} + \sum_{m'} XT_{it(t+Kmm')}^{gmm'} + SC_{git}^{m}\end{aligned} \quad \textbf{3.2}$$

$$\text{for all } i \in U \cup UV \cup V, m \in M, g \in G, t$$

$$\sum_{j}\sum_{t'} X^{gm}_{jt'it} + \sum_{m'} XT^{gmm'}_{i(t-Kmm')t} - DC^{m}_{gi(t-1)} = \sum_{j}\sum_{t'} X^{gm}_{itjt'} + \sum_{m'} XT^{gm'm}_{it(t+Kmm')} - DC^{m}_{git} + DE^{m}_{git} \quad \textbf{3.3}$$

$$\text{for all } i \in VW \cup W, m \in M, g \in G, t$$

$$SE_{git} = \sum_{m} SE^{m}_{git} \text{ for all } i \in U \cup UV, g \in G, t \quad \textbf{3.4}$$

$$DE_{git} = \sum_{m} DE^{m}_{git} \text{ for all } i \in U \cup UV, g \in G, t \quad \textbf{3.5}$$

$$DC^{m}_{git} = 0 \text{ for all } i \in N, m \in M, g \in G, t < EDT_{gi} \quad \textbf{3.6}$$

$$X^{gm}_{itjt'} \geq 0, XT^{gmm'}_{it(t+Kmm')} \geq 0, SE^{m}_{git} \geq 0, SC^{m}_{git} \geq 0, DE^{m}_{git} \geq 0, DC^{m}_{git} \geq 0 \text{ for all } i, j \in N, m \in M, m' \in M, g \in G, t, t' \quad \textbf{3.7}$$

$$YE^{m}_{it} + \sum_{j}\sum_{t'} Y^{m}_{jtit'} + YC^{m}_{i(t-1)} = \sum_{j}\sum_{t'} Y^{m}_{itjt'} + YC^{m}_{it} \quad \textbf{3.8}$$

$$\text{for all } i \in N, m \in M, t$$

$$Y^{m}_{itjt'} \leq ACA^{m}_{itjt'} \text{ for all } i, j \in N, m \in M, t, t' \quad \textbf{3.9}$$

$$Y^{m}_{itjt'} \geq 0 \text{ and integer, } YC^{m}_{it} \geq 0 \text{ and integer} \quad \textbf{3.10}$$

$$\text{for all } i, j \in N, m \in M, t, t'$$

$$YCA^{m} \times Y^{m}_{itjt'} - \sum_{g} X^{gm}_{itjt'} \geq 0 \quad \textbf{3.11}$$

$$\text{for all } i, j \in N, m \in M, t, t'$$

Ozdamar et al. (2004) also applied the multi-commodity, multi-modal network flow problem to logistics planning in natural disaster scenarios. Commodities are dispatched from supply points to distribution centers using different types of transportation resources, which are modeled similar to commodities. A logistics plan is formulated for multiple time periods with varying demand and supply by taking a snapshot of the current state of the system and forecasting future demand. The problem

is solved using Lagrangean relaxation by decomposing the problem in a manner similar to Haghani & Oh's first algorithm. The model was implemented using both artificial data sets and real data from a 1999 earthquake in Turkey.

The ability of this approach to replan based on changing circumstances makes it appealing, although it is limited by deterministic forecasting. For example, if demand for an item is exponentially distributed, planning based on mean demand could result in plans that perform poorly in execution. The basic idea of planning based on a current snapshot of the system's state as well as knowledge about future demands—or demand distributions—will be utilized in this thesis.

Barbarosoglu & Arda (2004) extend the multi-commodity, multi-modal network flow problem to a stochastic program. Their driving application was that of vital first-aid supplies to areas affected by disasters, especially earthquakes. Random variables modeled include resource requirements, arc capacities, and supply amounts. The problem is divided into two stages of response based on the amount of information available about supplies and demands. In the pre-event phase, supplies are allocated from supply nodes to other nodes before the demand is fully known. In the second phase, additional external supplies are not allowed, and a second transportation problem is solved given real demands and arc capacities. This model was also implemented for the 1999 Turkey earthquake scenario.

Accounting for stochastic variables in the distribution network gives this model great value for decision makers, especially as the possible conditions can be tied to specific earthquake scenarios. However, computational difficulty can limit the

applicability of stochastic programs, especially as the number of scenarios and time periods increases.

3.2 Decentralized Network Flow Problems

From the field of fleet management, two papers discuss decision making methods relevant to the current work. Adelman (2007), and Powell & Carvalho (1998) have used the concept of queueing networks to look at the problem of fleet management—a resource management problem encountered in freight transportation. Adelman views the problem from the perspective of an intermodal freight transportation provider's revenue management problem on a queueing network.

In this problem, a firm has a fleet of service units that move about a closed queueing network with nodes representing destinations on the transportation network. By serving requests and moving service units, varying rewards are earned. Ideally, a move generates a high reward and positions the service unit at a node with high demand from which the service unit can quickly find another profitable move to make. As tasks arrive according to a Poisson process, the controller's task is to accept/reject the tasks so as to maximize profit over the long run.

Adelman formulates the problem as a semi-Markov decision process with the state of the system given by the set of arrival times and destinations for all resources in the fleet. He then defines the control problem as optimizing the selection of tasks over time using long-run averages. An upper bound is found using a simple linear program. The model is then extended to consider queues and stockouts and approximate their

effect on the network. Furthermore, Adelman shows how the decision process can be distributed among agents that solve linear programs. Information is then used to make a dynamic programming value function approximation decomposed by unit that uses information about each unit's state to estimate the value of different system states. Numerical results show the value of these techniques. Future extensions to this model could include repositioning empty resources, better demand forecasting with customer classes, and variable fleet sizes, and incorporating demand information into pricing.

The decentralized fleet management approach provides the advantages of a clear mathematical framework for making decisions about which tasks to accept and reject over time. Tasks with different rewards for completion correspond well to tasks differentiated by priority. However, the strict focus on the transportation network leaves out important aspects about decisions that require both allocating supplies and transportation resources. Also, it focuses on satisfying demands as soon as possible and does not provide a way to manipulate complex schedules. In the logistics system considered in this thesis, demands may become known with varying lengths of time before their due dates.

3.3 Supply Chain Inventory Problem

Literature on inventory theory for multiechelon systems is also relevant to this thesis. For example, Graves [1996] studies a multiechelon inventory distribution system with stochastic demands and fixed replenishment intervals. Using a technique called virtual allocation and knowledge of Poisson arrivals, the virtual allocation rule is shown

to be near optimal. Under this policy, units in higher echelons of the system observe downstream demands, and always commit resources to the oldest outstanding orders. The policy is demonstrated for a two-echelon system.

The inventory approach contributes to the strategy of modeling incoming demands as Poisson processes. The virtual allocation rule highlights an important principle—that planning based on knowledge of these arrivals can improve system performance. However, this system also follows a fulfill as-soon-as-possible strategy for responding to demands. Also, the model does not account for the availability of transportation resources as a stress on the system.

3.4 Agent-Based Approaches

Agent-based approaches have provided an alternate perspective on solving logistics problems. The initial plea for the application of multi-agent systems (MAS) to transport logistics appears in Fischer et al (1996). Many reasons were offered for the suitability of MAS, including the complexity of the scheduling problem, the importance of common-sense and local knowledge, and the highly dynamic nature of the planning process. The article also proposed a multi-agent approach to designing the transportation problem with distributed artificial intelligence techniques of task allocation, decentralized planning, and negotiation. The exact problem examined was the dispatching of trucks performed by dispatchers. Companies and trucks are represented by agents, which interact cooperatively within a company and competitively between companies through

negotiation protocols. The ideas expressed in this paper set the stage for investigation into different strategies for decision making and implementation.

Jennings (2000) synthesized ideas about agent-based systems to articulate the basic concepts of MAS, their relevance, and their strengths and weaknesses. He argues, first, that "agent-oriented approaches can significantly enhance our ability to model, design and build complex, distributed software systems." Second, he predicts that "the agent-oriented approach will succeed as a mainstream software paradigm" as a "natural and logical evolution" of contemporary software approaches. Jennings identifies five characteristics of agents, beginning with their definition as problem-solving entities with well-defined boundaries and interfaces. Second, they sense information about their environment and can change the environment through effectors. Third, they have goals or objectives. Fourth, they exhibit autonomy through controlling their internal state and behaviors. Finally, agents are flexible, reactive, and able to anticipate the future. Jennings also notes that agents almost always exist in multi-agent systems.

From a software design perspective, Jennings argues that agents and the agent-paradigm provide advantages through effectively decomposing complex systems, naturally modeling complex systems, and appropriately modeling organizational relationships. He also names two disadvantages: difficulty predicting patterns and outcomes of interactions in a MAS and predicting the overall system behavior based on the knowledge about individual agents. In spite of these drawbacks, other projects and this thesis demonstrate the advantages of an agent-oriented approach conceptualizing, designing controls, and simulating a logistics system.

Drogoul et al. (2003) argue that most research supposedly performed using agent approaches, particular multi-agent based simulation, fails to have a design trajectory that ultimately leads to computational agents. They note widespread confusion about the definition of an agent often caused by the lack of a common set of semantics to discuss agent systems. They also find that many methodologies for multi-agent simulation underestimate the difficulties surrounding the computational model. To help resolve this, the authors propose a framework for developing multi-agent based simulations consisting of three levels of models: domain, design, and operational. By understanding each of these three roles, different communities who have their own definitions of agents can communicate across domains and minimize difficulties when translating from one domain to another.

Davidsson et al., (2005) and Shen et al., (2006) provide two reviews of applications of agent-oriented approaches to transport logistics and intelligent manufacturing systems, respectively. Davidsson et al conclude that agent-based approaches suit transport logistics problems well because of how they can represent distributed decision making entities with complex requirements and schedules. However, they desire to see more implementations and qualitative and quantitative assessments against existing systems. Similarly, Shen et al., note the scarcity of actual implementations and express concern that the field has progressed slowly since the late 1990s. They advocate a renewed focus on implementing agent technologies with existing enterprise systems.

Perugini (2006) provides one detailed multi-agent approach to the domain of transport logistics, especially military logistics planning. Selfish agents interact using

auctions under the Provisional Agreement Protocol, which is characterized by a deliberation process that leads up contracts, to form a logistics plan that satisfies global military requirements. The approach was applied to new military logistics scenarios, as well as scenarios from more traditional operations research approaches. Perugini concluded that the system's decentralized bidding produced results of similar quality to both centralized optimization and manual planning. As with most other approaches, all tasks in this system were to be fulfilled as soon as possible. Also, information about how the plan was evaluated—the design of a multi-agent based simulation—was not clear.

Like Perugini, Aknine et al (2004) propose a task allocation protocol for multi-agent systems, and its attributes were most relied upon for this thesis. It is an extension of the Contract Net Protocol designed for multi-agent systems, and uses the communication primitives of Announce, PreBid, PreAccept, PreReject, DefinitiveBid, DefinitiveAccept, and DefinitiveBid. This allows for concurrently bidding multiple tasks and the system to be flexible in dynamic environments by resubmitting bids and having multiple levels of commitment to a task.

An application of a multi-agent system to a military logistics system, is discussed by Gnanasambandam et al., (2004), who design a system in which a set of resources—discussed as computational resources, although able to be generalized to any resource—serve a set of tasks. The focus is maintaining the system's performance during execution. They propose control methods for the system by predicting system performance using an open Jackson network model. Solution quality and timeliness are the key measures of performance for the system, which works to allocate CPU resources that, in turn, serve the purpose of allocating supplies to military units in a distributed environment. The

queueing model is used to tune performance of the system, identify instabilities and bottlenecks, and identify aspects of the system for the hierarchical controller to give attention. Stresses from the physical world and the user act as sensed inputs for the agent, and various system components such as a repository of system technical specifications contribute important knowledge for decision making. Interesting extensions to this problem would include more detailed evaluation through a multi-agent based simulation and more complex requirements for fulfilling tasks, such as keeping a calendar-based plan instead of operating a service queue.

Lee et al (2008) further discusses agent strategies for both resource allocation and algorithm selection for a network similar to Gnanasambandam et al. Tasks are broadcast and distributed through an auction in this system, which was tested for 16 components distributed across three machines. Lee et al highlight trade-offs between quality of solution and solution time as an important aspect of decision making.

Chapter 4

System Description and Methodology

This chapter addresses the modeling of the problem as a multi-agent system, the simulation of the problem, and different control strategies for resource allocation. It also describes the simulation developed for this thesis, the Distributed Resource Allocation Planning and Execution Simulation (DRAPES).

4.1 The Agent-Based Approach

4.1.1 Domain Model

The logistics system is modeled as a multi-agent system primarily because that system is distributed and naturally decomposed by organizational entities. Those entities—units known as distribution centers (DCs) and operational units (OUs)—cooperate to solve a logistics problem of providing demands. Although a hierarchy of units exists, the units operate independently and interact with other units only through sending messages and shipping supplies. Each unit has control over its own state—its internal inventory of on-hand supplies—and maintains of a list of actions that it will take in the future to change its state. Demands originate at operations units and can be fulfilled internally by that OU or fulfilled externally by its primary DC or other secondary DCs. This process is illustrated in Figure **4-1**.

To solve the logistics problem, agents can interact in two ways: sending messages and sending supplies. Various types of messages, specified later in this chapter for different control methods, are passed between units to determine which unit will take responsibility for fulfilling a demand. Also, DCs can interact with OUs transferring supplies from the DC's on-hand inventory to an OU's on-hand inventory. This is done by subtracting supplies from the DC's on-hand inventory, loading them on transportation assets, and adding those supplies to the OU's on-hand inventory after a required transportation time.

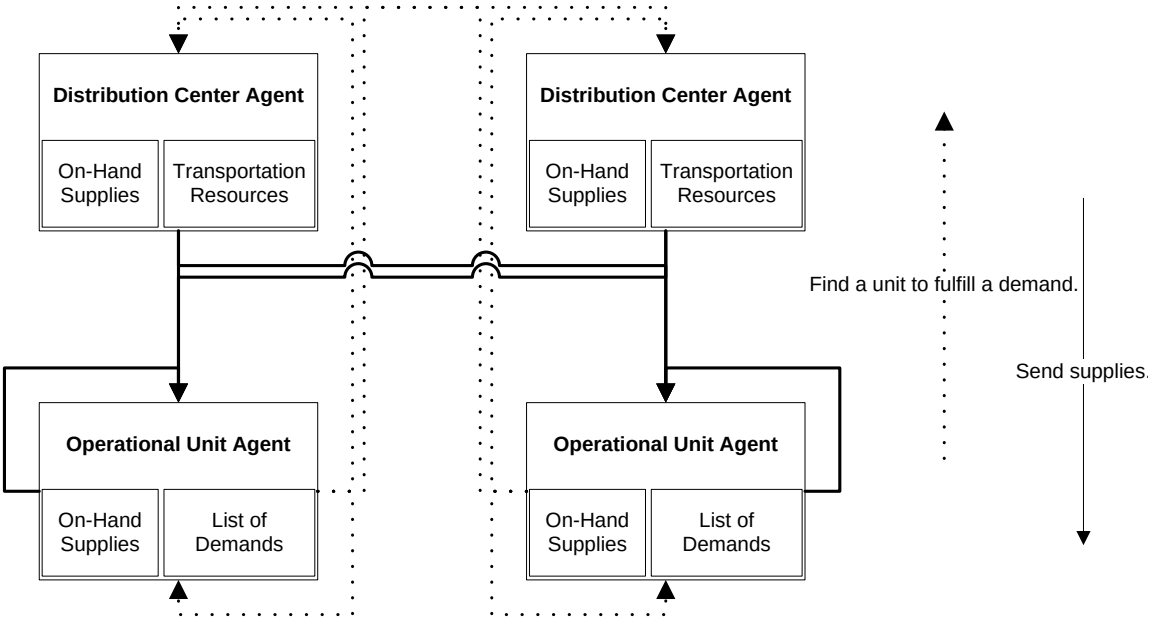

Figure **4-1**: Domain model.

Additionally, a third type of unit, the Central Controller (CC), can be added to the system for the purpose of controlling the system from a global perspective. The CC receives status updates from DCs and OUs, and disseminates orders to those units about when and where to allocate resources.

4.1.2 Design Model

Each unit in the system, whether a distribution center or an operational unit, is represented by a unit agent. A unit agent is an autonomous agent comprised of semi-autonomous activities that are executed concurrently. The unit agent is functionally decomposed into planning, execution, and simulation activities, as shown in Figure **4-2**.

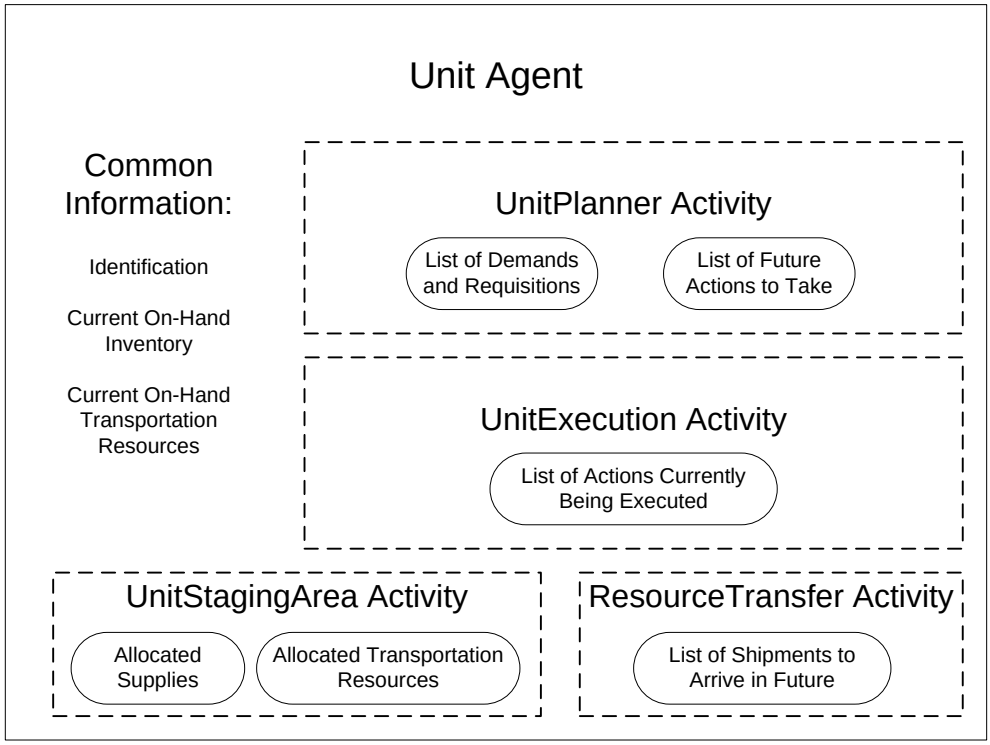

Figure **4-2**: Unit agent.

The *UnitPlanner* activity maintains most of the data and business logic of the agent. It keeps track of all of the unit's knowledge about the future and makes decisions based on that knowledge. The activity maintains three collections of information relevant decision making, including:

1. List of mission tasks that the unit has been assigned to execute. Each mission task has demands associated with it. At the time of execution, a demand will either have been sourced internally, externally, or not at all.
2. Expected additions/subtractions of supplies and transportation resources. Using the unit's current inventory and projected demands from mission tasks, this allows a unit to predict supply and transportation resource levels in the future.
3. A plan of actions to take in the future. The plan is comprised of plan elements, which represent specific actions to take at specific moments in time.

Using this information and a set of business rules, the unit can conduct internal planning and communicate with other units about plans. Also, through a system of timers and lists, the planner can send plan elements to the execution activity to execute.

The *UnitExecution* activity manages the real-time state changes of the agent. It receives actions to execute immediately from the planner activity, and provides feedback to the planner activity about the state of the agent. Most notably, it keeps track of a task that the unit is performing, and consumes on-hand resources from the inventory based on that task. When sending items to another agent, it transfers resources from on-hand inventory to the *UnitStagingArea* activity.

Two other minor activities—the *UnitStagingArea* activity and the *ResourceTransfer* activity—work with the UnitExecution activity to simulate the proper transfer of supplies between units. The UnitStagingArea activity matches supplies with transportation entities after they have been allocated to be sent to another unit. The resource transfer activity keeps knowledge about deliveries in progress, and notifies

units' execution activities once supplies have been delivered and transportation resources return to their sender.

4.1.3 Operational Model

The simulation was implemented as a distributed object-oriented discrete-event simulation in Java utilizing the OpenCybele agent framework (OpenCybele™ Agent Infrastructure Users Guide, 2002). OpenCybele provides the capability to distribute the system across multiple machines, support publish/subscribe communication between and within agents, and maintain a global discrete-event clock using timers. For future use of the simulation, a Java-based simulation provides for relatively easy interface with external software systems that could serve as emulators for inventory data storage or task generation.

In DRAPES, agents and activities are Java classes that implement classes from OpenCybele. They subscribe and can publish to communication channels defined by unique names, and available within agents or across the whole system. All communication between agents is done through a messaging protocol that enforces distributed computing through serializable Java objects. Agents and activities also possess the ability to set timers for events to occur in the future.

Timers, received external messages, and received internal messages all trigger a CybeleEvent. When setting a timer or subscribing to a communication channel, agents and agents' activities must specify a Java method with an argument of type CybeleEvent

that will be called when that event takes place. Examples are discussed later in this chapter.

4.2 Simulation Design

4.2.1 Assumptions

A set of assumptions help understand the operating logic of the simulation. First, barring any perturbations to the system specified in the experimental design, the only way resources are subtracted from the system is by allocating resources to send to another unit or by consuming resources as a mission task is executed. Demands are deterministic for the tests run in this thesis, although the capability to vary demands exists for the simulation. Furthermore, global information is available to each unit with so that the unit knows travel times, unit hierarchy information, and inventory parameters such as the truckloads-per-unit of inventory ratio. Finally, it is assumed that all communications between agents in the simulation are 100% successful.

4.2.2 Knowledge Classes

The basic forms of knowledge held by units are presented in Figure **4-3** as a web of knowledge classes. Each box in the figure is a Java class in DRAPES.

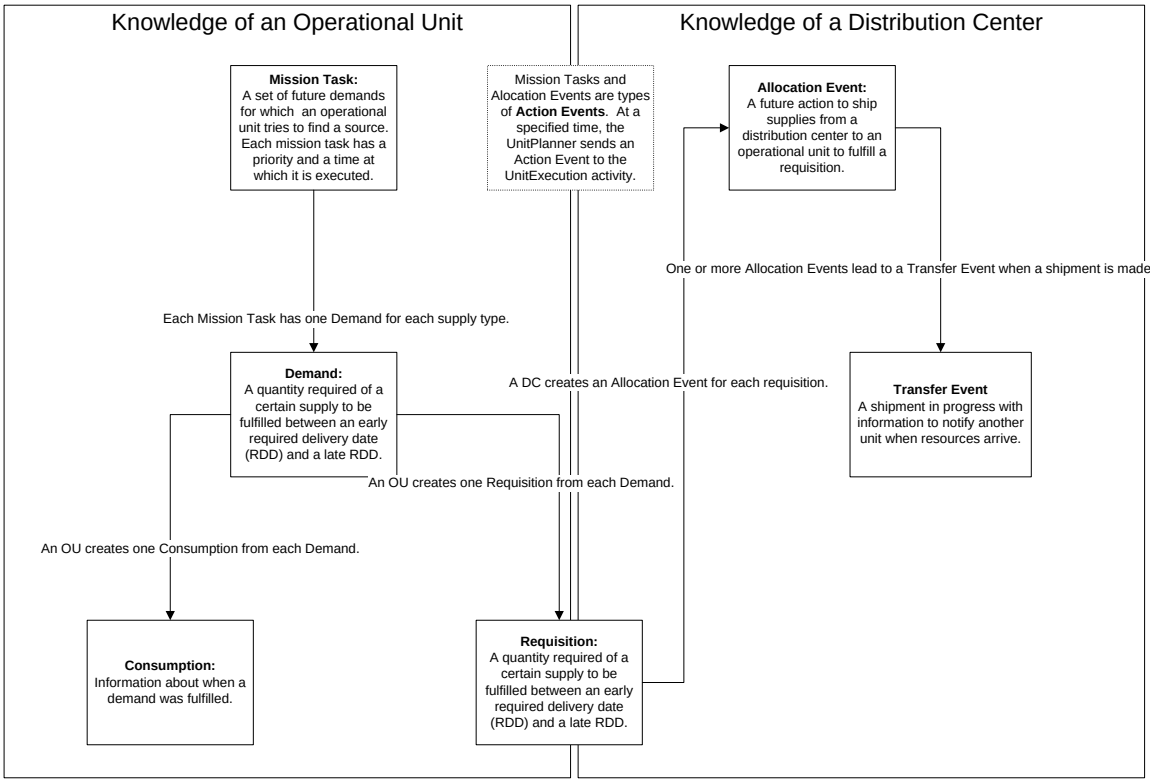

Figure **4-3**: Knowledge classes.

Knowledge of demands begins with the *mission task*. Mission tasks are distributed to operational units, which are responsible for executing the task at a future point in time. Each mission task is also assigned a priority between 0 and 1, with 1 being the highest priority.

Each mission task contains a *demand* for each commodity in the system. A demand is defined by the commodity, the quantity required of that commodity, and a required delivery window with an early required delivery date (RDD) and a late RDD.

To circulate knowledge of a demand and find a source for it, the OU creates a *requisition* for each demand. A requisition is the only knowledge class used by both OUs and DCs. The copy of a requisition held by the original requesting OU also keeps record

of planning information and status changes for the requisitions. Requisitions can take on different states while they are in the system, which are shown in Figure **4-4.** As a requisition state's changes, the source unit sends updates to the recipient unit so that it can update its knowledge of the future. Also, knowledge of requisition states helps units determine how much of its inventory is available to be allocated and how much should be reserved for requisitions in the fulfillment process. Some states are specific to decision making methods, such as the "bid selected" state.

When an OU executes the mission task, it examines its list of demands and requisitions associated with each demand. If the OU's copy of a requisition indicates that it a source was successfully found, it creates a *consumption*. Otherwise, that demand of the mission task is left unfulfilled. A consumption records information about the actual consumption so that each demand can be evaluated as fulfilled or unfulfilled and early or late upon the end of the simulation.

If a requisition is to be fulfilled by a DC, that DC creates an *allocation event* to be executed when it is time to ship the information. As the DC removes supplies from its inventory and transportation resources from its fleet, it creates a *transfer event* which keeps information about resources to be transferred from a DC to an OU in the future.

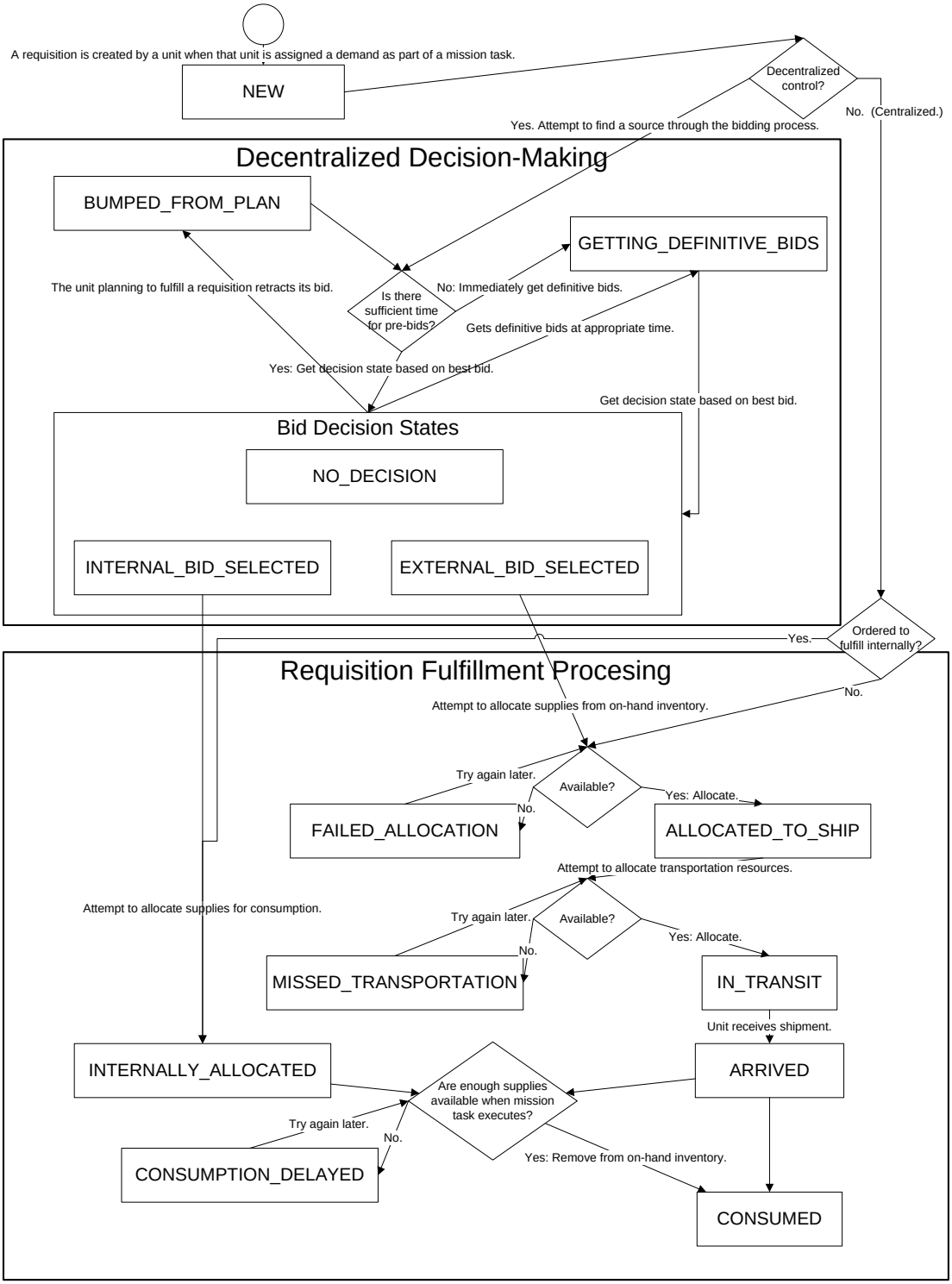

Figure **4-4**: Requisition state diagram.

4.2.3 The Simulation Agent

For basic simulation functionality in DRAPES, another agent—the Simulation Agent—is introduced. The simulation agent (SimAgent) has responsibilities to make the simulation work in a distributed fashion before, during, and after the simulation run.

At simulation initialization, the SimAgent creates each unit and initializes its state according to a configuration XML file. It also distributes the initial batch of mission tasks to each OU. During the simulation run, it may distribute more mission tasks as OUs are to become aware of them.

At the end of the simulation, the SimAgent also notifies each unit to end the simulation and prepare a report. The SimAgent also keeps track of statistics about mission tasks and units' resource usage. At the end of the simulation, it analyzes results and presents them as an XML file.

The SimAgent also has responsibility for notifying units about events that require re-planning. When an event occurs, the SimAgent sends a message to the relevant units with the appropriate information using predefined communication channels. Situations that cause re-planning events include:

1. The assignment of new tasks.
2. The unexpected loss of resources.
3. A mission task's priority change during the simulation run.

4.2.4 Simulation Communication Protocols

Regardless of the decision-making method employed in a simulation run, some basic processes remain the same. These can be defined by the event timers, internal agent communications, and agent-to-agent communications—the basic events in the system. These basic interactions are depicted in Figure **4-5**. Event timers are set by an activity of an agent for a future time when that same activity is notified of the event's occurrence. Internal agent communications are messages passed between different activities of a single agent. Agent-to-agent communications are messages passed between two agents using the communication channels to which agents publish and subscribe. Horizontal lines are messages passed within and between agents. Vertical lines are timers set by agents.

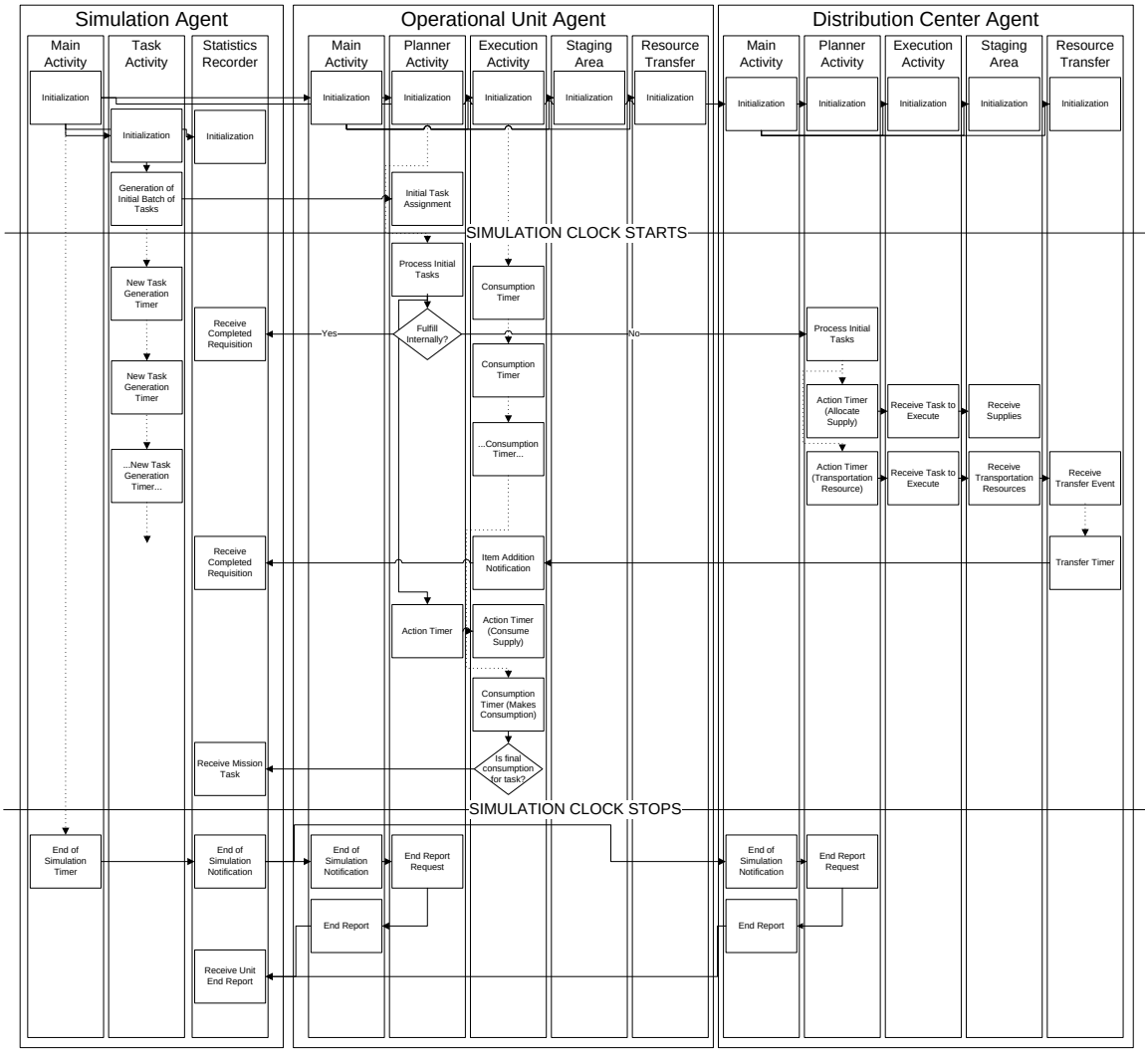

Figure **4-5**: Simulation interactions.

4.2.4.1 Internal Simulation Agent Timers

Name	New mission task generation timer
Message Data	(None)
Processing	• Create and send out a new mission task.

Name	End of simulation timer
Message Data	(None)
Processing	• Notify all units of simulation run end.

4.2.4.2 Unit Agent Events from Simulation Agent Messages

Name	Assignment of initial batch of mission tasks
Message Data	List of mission tasks
Processing	• Record mission tasks. • Create requisitions for each demand

Name	Assignment of mission tasks during simulation run
Message Data	Mission task
Processing	• Record mission tasks. • Create requisitions. • Decide on actions and adjust plan.

Name	Exogenous resupply notification.
Message Data	Commodity, Quantity
Processing	• Add items to on-hand inventory.

Name	End of simulation notification.
Message Data	(None)
Processing	• Assemble necessary information. • Send report to Simulation Agent.

4.2.4.3 Simulation Agent Events from Unit Agent Messages

Name	Receipt of completed requisitions (from OUs)
Message Data	(None)
Processing	• Store for processing at end of simulation.

Name	Receipt of completed mission tasks (from OUs)
Message Data	(None)
Processing	• Store for processing at end of simulation.

Name	Receipt of Unit Agent end report.
Message Data	Unit agent ID, list of unfulfilled requisitions, resource usage reports.
Processing	• Store for processing. • If all unit agent end reports have been received, prepare end of simulation report.

4.2.4.4 Messages Passed Between Agents

Name	Receipt of Unit Agent end report.
Message Data	Unit agent ID, list of unfulfilled requisitions, resource usage reports.
Processing	• Store for processing. • If all unit agent end reports have been received, prepare end of simulation report.

Name	Requisition status update.
Message Data	Requisition ID, new requisition status.

Processing	• Record status update. • Take actions if necessary.

Name	Receipt of resources by a shipment.
Message Data	Transfer Event.
Processing	• Add items to inventory.

It should also be noted that different control methods will contain additional message protocols for determining which unit will fulfill a requisition.

4.2.4.5 Internal Unit Agent Messages and Timers

Name	Notification to execute an Action Event (from the UnitPlanner to the UnitExecution activity).
Message Data	Action Event (a mission task or an Allocation Event)
Processing	• Create Consumption or take steps to send resources to other units.

Name	Consumption Timer (by the UnitExecution activity of an OU)
Message Data	(None)
Processing	• Given the current list of Consumptions, remove resources from inventory to fulfill requisitions.

Name	Allocation of supplies to ship (by UnitStagingArea activity of a DC).
Message Data	Requisition
Processing	• Record information. • Check if transportation resources are available to ship.

	• If transportation resources are available, ship supplies for the requisition. • Otherwise, add to a queue for transportation resources sorted by a combination of requisition priority and lateness.

Name	Receipt of Transfer Event (by ResourceTransfer activity of a DC).
Message Data	Transfer Event
Processing	• Record event. • Set timer to transfer resources.

Name	Transfer Event timer.
Message Data	(None)
Processing	• Retrieve Transfer Event from list. • Notifying unit receiving supplies.

4.3 Decentralized Control

The first method for making decisions about which units allocate resources to which requisitions is a decentralized approach. Operational units and distribution centers communicate their ability to fulfill a requisition through a bidding process. Each operational unit—the ultimate consumer of supplies for a requisition—decides based on those bids who will fulfill a requisition. A decentralized approach is advantageous in that the data sensed and known by each individual agent can be large while the data passed between agents is relatively small.

4.3.1 Communication Protocols

The communication protocol for bidding is implemented with communication performatives from Aknine et al (2004), as shown in Figure **4-6**. The performatives used in this thesis are defined as follows:

- *Announce*. An Operational Unit initially sends an Announce message to all Distribution Centers that are possible sources for fulfilling a requisition. Included in this message is the information about the requisition. Later, a second Announce message is sent soliciting a final bid—the DefinitiveBid discussed below.
- *PreBid*. In response to an Announce message, each Distribution Center calculates its bid and sends it to the requesting Operational Unit. DCs may also send updated PreBids as its circumstances change, depending upon the experimental scenario.
- *PreAccept*. After an Operational Unit selects a bid, it sends a PreAccept message to the lowest bidding unit, which could be itself or a DC. That unit adds fulfilling that requisition to its plan. If the unit's bid would change based on circumstances to become lower, it could submit a revised PreBid. Also, if accepting this requisition causes the unit's plan to be infeasible, low-priority requisitions are removed from the plan and the requesting units for those removed are notified.
- *PreReject*. Units not selected to fulfill a requisition are notified with PreReject messages. For the specified requisition, they will take no

further action unless their bid becomes lower, which would cause them to submit a revised PreBid. If the unit had previously planned to fulfill this requisition, it removes the requisition from its plan.

- *DefintiveBid*. Each potential supplier sends a DefinitiveBid in response to a second Announce message. That bid is the unit's final bid.
- *DefinitiveAccept*. When a unit receives a DefinitiveAccept message, it both adjusts its plan and creates Action Events necessary to ship the requisition or fulfill it internally. Again, if accepting this requisition causes the unit's plan to be infeasible, low-priority requisitions are removed from the plan and the requesting units for those removed are notified.
- *DefinitiveReject*. A unit receiving a DefinitiveReject message permanently removes that requisition from its knowledge of possible requisitions to fulfill. If it had previously planned to fulfill the requisition based on an earlier bid, it removes those events from its plan.

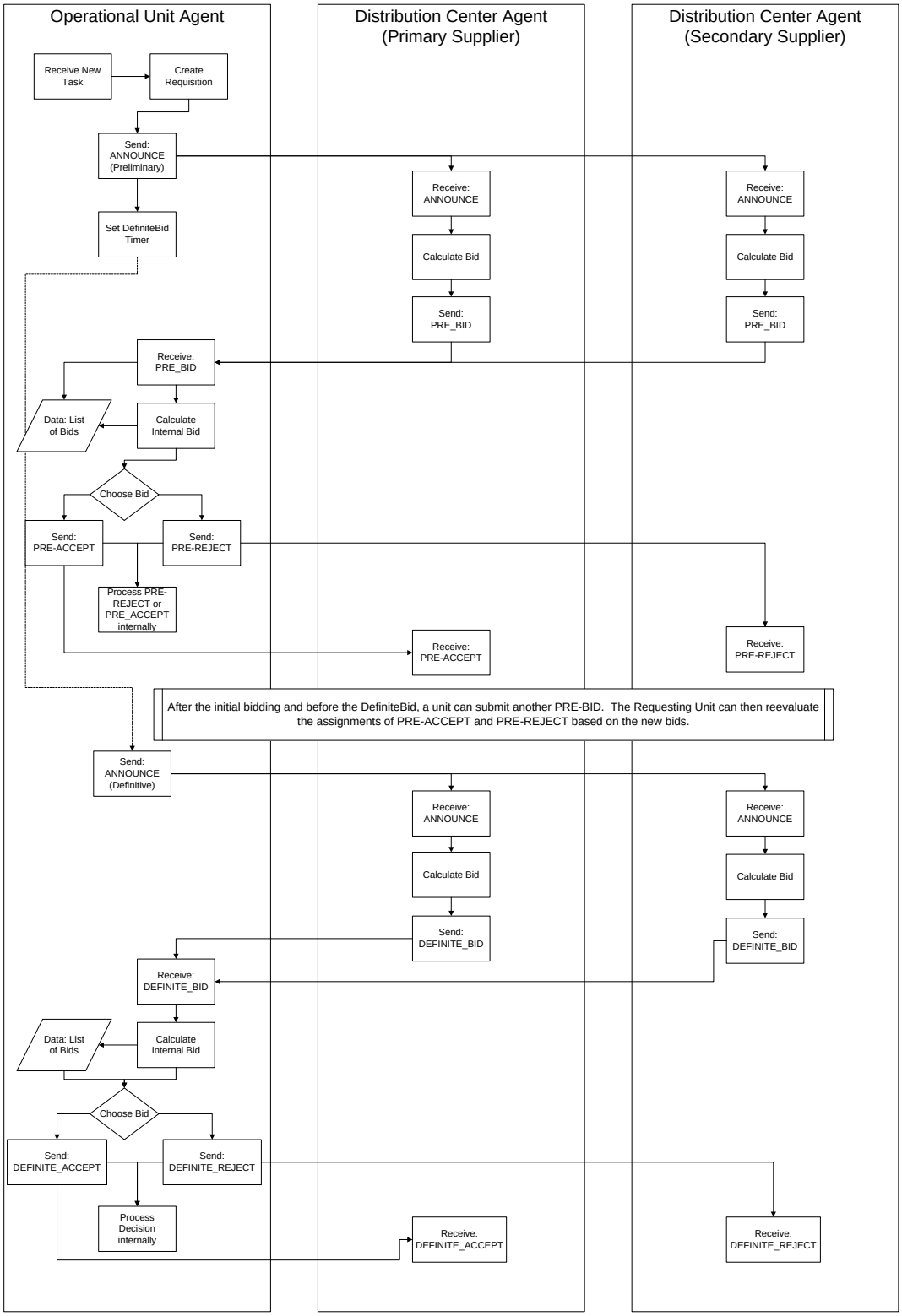

Figure **4-6**: Bidding process.

To support decentralized decision making, the following message type is used for bidding-related communications between unit agents:

Name	Bidding message-
Message Data	Depends upon performative, but could including bidding and requisition information.
Processing	• Process based on performative, as described above.

After the simulation is initialized, an initial plan is developed using these bidding protocols. The first hour of the simulation time frame is devoted to bidding and assignment of requisitions. Requisitions are bid iteratively, starting with the highest priority requisition at the beginning of the hour, and the lowest priority requisition at the end of the hour. This eliminates problems arising from bidding on multiple requisitions at once.

4.3.2 Bid Calculation

When a unit $i \in U$ is solicited to bid on requisition, it must calculate a bid based on its local view of the system. The calculation of the bid is expressed using the following sets, parameters, and variables. This applies to both distribution centers and operational units.

Sets

U Set of all units in the system

K Set of commodities

P^k Set of plan events (receipts and issuances) occurring in the future for commodity k, sorted by time from earliest to latest, $k \in K$

Parameters

t_0 Simulation start time

t_f Simulation end time

t_d Current time of a decision

T_{ij} Travel Time from unit $i \in U$ to unit $j \in U$

r_m^k Time at which the m^{th} plan event occurs, $m \in P^k$

B_k Initial inventory level of commodity $k \in K$

q_m^k Change in inventory level due to the m^{th} plan event, $m \in P^{i,k}$

p_m^k Priority associated with the m^{th} plan event, $m \in P^{i,k}$

α Supportability penalty

β Travel time penalty

γ Plan changeability penalty

π Ratio for comparing probability of success vs. earliness/lateness when evaluating bids

λ_j Arrival rate of new tasks to unit $j \in U$

z^k Acceptance rate of requisitions of commodity $k \in K$

m_k Mean quantity for exponentially distributed commdity $k \in S$

Variables

$x^k(t, v)$ Priority-filtered Inventory level at time t, considering only plan events $m \in P^k: p_m^k \geq v$

$u^k(m, v)$ Inventory level after the occurrence of plan event m, considering only plan events $m \in P^k: p_m^k \geq v$

$a^k(t, v)$ Number of supplies available at time t, considering only priority-filtered plan events $m \in P^k: p_m^k \geq v$

$y^k(T, \lambda, v)$ Sum of supplies of new tasks arriving over a time interval of duration T with rate λ and priority greater than or equal to v

Bid Components

b_1 "Supportability" estimate

$l(t)$ Earliness/Lateness of a requisition delivered at time t

$s(t)$ Probability of successfully fulfilling a requisition to be delivered at time t

b_2 Transportation time

b_3 "Changeability" estimate

Calculating the bid for a requisition is a six-step process followed by each unit that submits a bid. The steps are as follows:

1. Consider a requisition with the following information:

The following information is disseminated by the requesting unit, the Operational Unit that creates a requisition for a demand:

RT Requisition's Requested Delivery Time (mean of early and late RDDs)

RP Requisition Priority

RQ Requisition Quantity

RU Requesting Unit

RS Requested Commodity ($k = RS$)

2. Calculate the number of supplies available at the possible times when the requisition would be allocated. Also consider transportation.

The goal of this step is to ensure that allocating supplies to the proposed requisition would not affect higher priority requisitions that the unit will fulfill at any time. The unit considers fulfilling requisition at various times, including two days early, one day early, on time, one day late, and two days late. A set of times is considered for allocating supplies to a requisition, with each time in that set being $d \in \{RT - T_{i(RU)} \pm (0,1,2 \text{ days})\}$. The first step is to find the available inventory at time d so that no backorders will be created after at or after time d. This is expressed as

$$a^k(d, RP) = \min(x^k(t, RP)), \qquad d \le t \le t_f \tag{4.1}$$

This depends on the inventory level on a continuous time interval, which follows the equation

$$x^k(t, v) = x^k(d, 0) + \sum_{m:\{t_d \le r_m^k \le t,\, p_m^k \ge v\}}^{P^k} q_m^k \tag{4.2}$$

If $a^k(d, RP) < RQ$, there are not enough supplies available at $t = d$. Therefore, $l(d) = \infty$, and the total bid value is set to ∞. No other component of the bid need to be calculated. Otherwise, the earliness/lateness component is defined as

$$l(d) = d - (T_{i(RU)} - RT) \quad \textbf{4.3}$$

If the unit calculating the bid is a distribution center, it needs to revise its bid based on the availability of transportation resources. Distribution centers maintain plans for transportation resources in the same format as commodities, so similar process can be used. The main difference is that the concern is only that transportation resources are available for the duration of round-trip travel. That is, allocating available transportation resources now has no effect creating backorders on transportation allocations taking place after the necessary travel time. Equation **4.1** is used for the transportation plan but revised to have the time constraints $d \leq t \leq T_{i(RU)} + T_{(RU)i}$. If the required transportation is not available, then the total bid value is revised to be ∞. Otherwise, transportation resource availability does not affect the plan.

3. Calculate the probability of the allocation for the proposed requisition being successfully made.

The goal of this step is to determine the probability that, if the unit takes on this proposed requisition, it will successfully fulfill it. That is, resources will not be taken away to serve other new requisitions with higher priority that will arrive in the future. Since the other new requisitions are currently unknown, the probability of fulfilling the proposed requisition must be found. If the arrival rate of new requisitions is zero, then

the probability of successfully fulfilling the proposed requisition is 100%. Otherwise, it is calculated using the following process:

$$s(d) = 1 - \left[P\left(y^k\left(t_f - t_d, z_j, RP\right) > \left(a^k(d, RP) - RQ\right)\right)\right] \quad \textbf{4.4}$$

Equation **4.4** asserts that the probability of successful fulfillment is the probability that the sum of demands from future requisitions is less than the difference of available supplies and supplies needed for the proposed requisition. For each unit, z_j is meant to be an estimate of the fraction of requisitions announced by unit *j* that are fulfilled by the bidding unit. This fraction is calculated by counting the number of bids for requisitions from unit *j* that are won and lost for the entire simulation.

Equation **4.4** is expanded using conditional probability to become

$$\begin{aligned} s(d) = 1 - \sum_{i=1}^{\infty} & P\left(Poisson\left(z_k(1 - RP)\right) = i\right) \\ & * P\left(Erlang\left(i, \left(z_k(1 - RP)\right)\right) > a^k(d, RP) - RQ\right) \end{aligned} \quad \textbf{4.5}$$

Further,

$$\begin{aligned} & s(d) \\ & = \sum_{i=1}^{\infty}\left[\left(\frac{\left(z_k(1 - RP)\right)^i e^{-\left(z_k(1-RP)\right)}}{i!}\right)\right. \\ & \left.*\left(\sum_{n=0}^{i-1}\frac{e^{-\left(z_k(1-RP)\right)\left(a^k(d,RP)-RQ\right)}\left[\left(z_k(1 - RP)\right)\left(a^k(d, RP) - RQ\right)\right]^n}{n!}\right)\right] \end{aligned} \quad \textbf{4.6}$$

4. Calculate the "supportability" component of the bid.

With the results from steps 2 and 3, the supportability component b_1 of the bid can be calculated. It should be noted that a tradeoff could exist between on-time delivery

and probability of failed allocation. This is accounted for in following expression, which also finds the best delivery time to use for the bid:

$$b_1 = \min\left[l(d) + \pi\left(\frac{1}{s^2(d)} - 1\right)\right] \text{ for all } d \in \{RT - T_{i(RU)} \pm (0,1,2 \text{ days})\} \quad \textbf{4.7}$$

If $\pi = 1$, then a one-day late delivery with certain fulfillment has the same penalty as an on-time delivery with an approximately 31% chance of failure.

5. Calculate the round-trip travel time.

This is simply

$$b_2 = T_{i(RU)} + T_{(RU)i} \quad \textbf{4.8}$$

6. Calculate the "changeability" penalty.

The system changeability penalty is incremented by 1 every time the unit assigned to fulfill a requisition changes. This includes a change from a unit fulfilling a requisition to no unit fulfilling a requisition. The following steps are used to calculate b_3:

1. Add the announced requisition to the set of plan elements P^k and sort the set from earliest event to latest.
2. Check if this unit or another unit had been planning to fulfill the proposed requisition.
    a. If this unit had been planning to fulfill, set $b_3 = 0$.
    b. Otherwise, set $b_3 = 1$.
3. Remove tasks from plan that will no longer be able to be fulfilled.
    a. While $min\left(x^k(t,0)\right) < 0, \quad t_d \leq t \leq t_f$
        i. Remove the lowest priority plan event from P^k

ii. Increase b_3 by 1.

4.3.3 Bid Selection

Once all bids have been received, the Operational Unit selects a bid using weights from the system's utility function. Its goal is to pick the minimum total bid, with each bid defined as

$$total\ bid\ value = \alpha b_1 + \beta b_2 + \gamma b_3 \quad \textbf{4.9}$$

Once the low bidder is identified, acceptance and rejection notices are sent to the all units that submitted bids.

4.3.4 Plan Example

To illustrate the ability of units to make and keep plans, an example plan is given here using the point of view of a single unit in the system. Each unit –both operational units and distribution centers—keeps a list of plan events for each commodity type that, given the current state of the unit, can be used to predict the unit's state in the future. A plan event is defined by a time, a quantity (positive for receipts and negative for issuances), and a requisition or event name (such as "exogenous resupply") associated with the event. Each unit begins the simulation run with no events on its plan. It first receives notification of exogenous resupply event times and quantities, and adds them to its plan. Then, as the unit accepts responsibility to fulfill requisitions, events are added to the plan corresponding to those requisitions.

In the scenario used for this example, the planning for one operational unit, known as OU 1-1, is examined. Only one commodity type exists, and the unit must plan for 22 requisitions. It requests and processes bids for each of these requisitions, starting with the highest priority requisition. As the bidding process determines which unit—the operational unit (OU 1-1), its direct superior (DC 1), a lateral unit of DC 1 (DC 2), or no unit at all—the unit adjusts its plan by adding plan events to create its initial plan. The results of this process are shown in Table **4-1** for fulfilled requisitions and Table **4-2** for unfulfilled requisitions.

Table **4-1**: Fulfilled requisitions of example plan.

Requisition Number	Priority	Earliness/Lateness	Recipient Unit	Quantity	Source Unit	Execution Time
40	0.86	0.0	OU 1-1	0.99	OU 1-1	Mon May 04 14:30:36 EDT 2009
38	0.83	0.0	OU 1-1	0.81	OU 1-1	Mon May 04 16:42:42 EDT 2009
13	0.79	0.0	OU 1-1	0.53	OU 1-1	Wed May 06 20:14:00 EDT 2009
37	0.72	0.0	OU 1-1	0.08	OU 1-1	Wed May 06 21:24:42 EDT 2009
34	0.72	0.0	OU 1-1	0.37	DC 1	Sun May 03 05:21:04 EDT 2009
27	0.72	0.0	OU 1-1	0.39	OU 1-1	Wed May 06 05:56:02 EDT 2009
17	0.71	0.0	OU 1-1	1.88	DC 1	Wed May 06 13:36:14 EDT 2009
16	0.71	0.0	OU 1-1	0.13	OU 1-1	Thu May 07 10:02:57 EDT 2009
36	0.66	0.5	OU 1-1	1.45	DC 1	Wed May 06 02:47:59 EDT 2009
24	0.57	0.0	OU 1-1	0.53	OU 1-1	Sat May 09 23:52:19 EDT 2009
10	0.50	0.0	OU 1-1	0.81	DC 2	Wed May 06 05:51:07 EDT 2009
20	0.35	0.0	OU 1-1	0.16	DC 1	Mon May 04 23:02:58 EDT 2009
28	0.11	0.0	OU 1-1	0.44	OU 1-1	Sat May 09 12:05:54 EDT 2009

Table **4-2**: Unfulfilled requisitions of example plan.

Requisition Number	Priority	Recipient Unit	Quantity	Execution Time
8	0.59	OU 1-1	3.26	Mon May 04 15:10:46 EDT 2009
30	0.38	OU 1-1	3.17	Fri May 08 06:47:23 EDT 2009
14	0.23	OU 1-1	0.25	Mon May 04 10:23:17 EDT 2009
5	0.20	OU 1-1	0.55	Wed May 06 08:56:22 EDT 2009
6	0.18	OU 1-1	1.83	Mon May 04 07:02:24 EDT 2009
3	0.14	OU 1-1	0.97	Thu May 07 14:52:16 EDT 2009
2	0.12	OU 1-1	0.68	Tue May 05 10:23:32 EDT 2009
18	0.10	OU 1-1	0.73	Sun May 03 20:47:09 EDT 2009
19	0.07	OU 1-1	2.62	Sun May 03 04:44:00 EDT 2009

From the decisions made about requisitions, the unit populates its plan with plan events. A chronological list of plan events is shown in Table **4-3**. Based on these events, the future inventory level can be predicted for the remaining time in the simulation.

Table **4-3**: Plan event list for unit OU 1-1.

Time	New Inventory Level	Event Description
Fri May 01 00:00:33 EDT 2009	2.00	Starting Inventory
Fri May 01 17:26:04 EDT 2009	2.37	Req. 34 Arrives
Sun May 03 06:00:33 EDT 2009	2.00	Req. 34 Consumed
Sun May 03 11:07:58 EDT 2009	2.16	Req. 20 Arrives
Mon May 04 15:00:33 EDT 2009	1.17	Req. 40 Consumed
Mon May 04 17:00:33 EDT 2009	0.36	Req. 38 Consumed
Mon May 04 17:56:07 EDT 2009	1.18	Req. 10 Arrives
Tue May 05 00:00:33 EDT 2009	1.01	Req. 20 Consumed
Tue May 05 00:00:33 EDT 2009	2.01	Exogenous Resupply
Tue May 05 01:41:14 EDT 2009	3.89	Req. 17 Arrives
Tue May 05 14:52:59 EDT 2009	5.34	Req. 36 Arrives
Wed May 06 03:00:33 EDT 2009	3.89	Req. 36 Consumed
Wed May 06 06:00:33 EDT 2009	3.08	Req. 10 Consumed
Wed May 06 06:00:33 EDT 2009	2.69	Req. 27 Consumed
Wed May 06 14:00:33 EDT 2009	0.81	Req. 17 Consumed
Wed May 06 21:00:33 EDT 2009	0.28	Req. 13 Consumed
Wed May 06 22:00:33 EDT 2009	0.20	Req. 37 Consumed
Thu May 07 11:00:33 EDT 2009	0.07	Req. 16 Consumed
Sat May 09 00:00:33 EDT 2009	1.07	Exogenous Resupply
Sat May 09 13:00:33 EDT 2009	0.63	Req. 28 Consumed
Sun May 10 00:00:33 EDT 2009	0.09	Req. 24 Consumed
Mon May 11 00:00:33 EDT 2009	0.09	End of Simulation

From this table, the graph shown in Figure 4-7 can be drawn showing the predicted future inventory level over time. For a plan to be feasible, the predicted inventory level at any time in the future must never be negative. If no perturbations occur to disrupt the plan from being executed as initially planned, this graph also represents the historical record of inventory consumption.

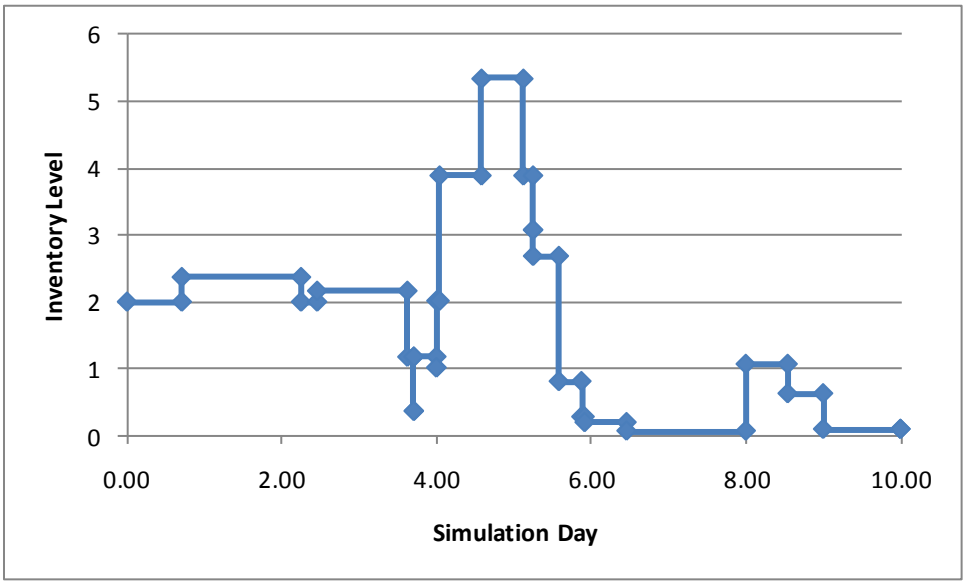

Figure **4-7**: Predicted inventory level over time for unit OU 1-1.

4.3.5 Re-planning

When units receive messages from the Simulation Agent that change the state of the system, they consider adjusting their plans for requisitions to fulfill and when to fulfill them. As an assumption of the logistics system, requisitions that have begun to be fulfilled through shipment or internal allocation of supplies are ineligible to have their fulfillment plans changed. Only requisitions in the bidding process are eligible to be pre-empted or to have other units submit revised pre-bids.

After a unit is notified that it has responsibility to fulfill a requisition during the simulation run, the unit checks to make sure it can still honor all of its supply allocation commitments. If accepting the new requisition causes its plan to be infeasible, it removes

requisitions, starting with the lowest priority requisition, until the plan is feasible. The recipient unit for each of these removed requisitions is notified through a requisition status update, and that unit can subsequently re-announce and re-bid these requisitions.

If supplies are lost unexpectedly during the simulation run, a unit undergoes a similar process. It also removes requisitions from its plan of supply allocations until it can honor all allocation commitments. Again, the recipient unit for each requisition is notified and can find another supplier through another iteration of the bidding process.

When a mission task's priority changes, units adjust their plans to cancel fulfilling requisitions related to that mission task, except those already in the process of fulfillment. The removed requisitions are then treated as new requisitions and re-bid. In light of these changing plans, especially if a high-priority requisition becomes a low-priority requisition, units iteratively re-calculate bids for the requisitions that it was not scheduled to fulfill. Lower bids are re-submitted, and the recipient unit can change which unit is to fulfill the requisition.

4.4 Centralized Control

An alternate method of decision making introduces another agent, the *CentralController* agent. Instead of deciding which units fulfill which requisitions based on bids, all units send updates to the CentralController at specified intervals. The CentralController's goal is to make more informed decisions based on a global view of the system. To do so, it formulates and solves a mixed-integer program and disseminates

the results as orders. Unit agents receive the orders and prepare Action Events as necessary.

4.4.1 Communication Protocols

The following list of communication events between a CentralController and unit agents support this method for controlling the system. Messages passed from the CentralController to unit agents include:

Name	Request for status update
Message Data	(None)
Processing	• Prepare a status update by examining the state of resources and requisitions. • Send status update to CentralController.

Name	Receipt of orders
Message Data	List of orders, each pertaining to a requisition.
Processing	• Prepare necessary Action Events to follow orders..

The only message passed from unit agents to the CentralController is the following:

Name	Receipt of unit update
Message Data	List of requisitions, status reports for resources.
Processing	• Record information. • If updates have been received from all unit agents, build and solve the mathematical program.

4.4.2 Mathematical Model

The mixed-integer program presented by Lichter (2009) was used with modifications to model the problem. Differences include a more refined representation of transportation assets in shipment using discrete transportation assets and knowledge of requisition destinations for the purpose of combining multiple requisitions into one shipment. The program's main purpose is to allocate available repair parts to requisitions over a fixed time horizon in a deterministic environment.

Sets

I	Set of Requisitions
J	Set of Shipping Methods
K	Set of Units
L	Set of Commodities
T	Set of Time Periods

Parameters

R_{ik}	$\begin{cases} 1 \text{ if requisition } i \in I \text{ has destination unit } k \in K \\ 0 \text{ otherwise} \end{cases}$
D_i	Due-date of requisition $i \in I$
t_0	Current time period, or time of decision
t_f	Final time period
I_{kl}	Initial on-hand inventory of commodity $l \in L$ held by unit $k \in K$
M_{il}	$\begin{cases} 1 \text{ if requisition } i \in I \text{ is for commodity } l \in L \\ 0 \text{ otherwise} \end{cases}$

Q_{il} Quantity of commodity $l \in L$ associated with requisition $i \in I$

C_{ijk} Cost of fulfilling requisition $i \in I$ with unit $k \in K$ using shipping method $j \in J$

T_{ijk} Delivery time for fulfilling requisition $i \in I$ with unit $k \in K$ using shipping method $j \in J$

P_i Earliness/Lateness penalty for requisition $i \in I$

S_{klt} Additional exogenous supplies of commodity $l \in L$ received by unit $k \in K$ at the beginning of time period $t \in T$

V_l Volume ratio of commodity $l \in L$ in truckloads per unit

L_{kt} Maximum number of truckloads allowed to be shipped by unit $k \in K$ at time period $t \in T$

Variables

x_{ijkt} $\begin{cases} 1 \text{ if requisition } i \in I \text{ is shipped by unit } k \in K \text{ in time period } t \in T \\ \qquad\qquad\qquad\qquad \text{using method } j \in J \\ 0 \text{ otherwise} \end{cases}$,

d_i Ship date of requisition $i \in I$ (assumed to be beginning of a time period)

s_i Earliness or lateness of requisition $i \in I$

$l_{kk't}$ Truckloads from the fleet of unit $k \in K$ carrying shipments to unit $k' \in K$ during time period $t \in T$

$l'_{kk't}$ $\lceil l_{kk't} \rceil$, Integer-valued truckloads from the fleet of unit $k \in K$ carrying shipments to unit $k' \in K$ during time period $t \in T$

a_{klt} Inventory of part $l \in L$ held by agent $k \in K$ during time period $t \in T$

r_{klt} Quantity of part $l \in L$ shipped by agent $k \in K$ during time period $t \in T$

Objective

$$\text{Minimize } z = \sum_{i \in I} \sum_{j \in J} \sum_{k \in K} \sum_{t \in T} C_{ijk}\, x_{ijkt} + \sum_{i \in I} {s_i}^2 P_i \quad \textbf{4.10}$$

Constraints

$$\sum_{j \in J} \sum_{k \in K} \sum_{t \in T} x_{ijkt} = 1 \text{ for all } i \in I \quad \textbf{4.11}$$

$$d_i = \sum_{t \in T} t x_{ijkt} \text{ for all } i \in I \quad \textbf{4.12}$$

$$d_i + s_i + \sum_{j \in J} \sum_{k \in K} \sum_{t \in T} T_{ijk}\, x_{ijkt} = D_i \text{ for all } i \in I \quad \textbf{4.13}$$

$$t_0 \le d_i \le t_f - 1 \text{ for all } i \in I \quad \textbf{4.14}$$

$$a_{kl t_0} = I_{kl} \text{ for all } k \in K, l \in L \quad \textbf{4.15}$$

$$a_{klt} = a_{kl(t-1)} + S_{klt} - r_{klt} \text{ for all } k \in K, l \in L, t \in T \text{ such that } t > t_0 \quad \textbf{4.16}$$

$$r_{klt} = \sum_{i \in I} \sum_{j \in J} M_{il} Q_{il} x_{ijkt} \text{ for all } k \in K, l \in L, t \in T \quad \textbf{4.17}$$

$$l_{kk't} = \sum_{i \in I} \sum_{j \in J} \sum_{l \in L} M_{il} Q_{il} V_l x_{ijkt} \text{ for all } k \in K, k' \in K, t \in T \quad \textbf{4.18}$$

$$l_{kk't} \le l'_{kk't} \text{ for all } k \in K, k' \in K, t \in T \quad \textbf{4.19}$$

$$l'_{kk't} \text{ is an integer} \quad \textbf{4.20}$$

$$\sum_{k' \in K} l'_{kk't} \le L_{kt} \text{ for all } k \in K, t \in T \quad \textbf{4.21}$$

$$a_{klt} \ge 0 \text{ for all } k \in K, l \in L, t \in T \quad \textbf{4.22}$$

4.4.3 Model Implementation and Input Parameters

The translation of a snapshot of the simulation to input parameters for the model is fairly straightforward, except for a couple unique situations. Notably, the set of unit agents is a combined set of the distribution centers, operational units, and a "shadow unit" representing a requisition being unfulfilled. For the shadow unit, the cost of fulfilling is defined to be the same as the penalty for that requisition not being fulfilled. The shadow unit has infinite on-hand inventory and transportation capacity.

For transportation constraints, the transportation capacity is the number of truckloads a unit can provide per time period. The cost of fulfilling a requisition, C_{ijk}, is the shipping cost plus a changeability penalty γ if another unit is currently scheduled to fulfill that requisition. A large number is for this fulfillment cost is used to prevent operational units from fulfilling other operational units' requisitions. Similarly, the truck capacity is infinite for operational units to allow for allocation from on-hand inventory.

In DRAPES, the model is implemented with one-day time periods. The earliness/lateness and unfulfilled penalties for requisitions incorporate each requisition's priority and the penalty α from the simulation's utility function. Likewise, the transportation cost is determined by the round-trip transportation time and the penalty β.

It is scheduled to run at approximately midnight of every day of the simulation. Orders the unit agents receive from the most recent run countermand any previous orders. Requisitions that already have supplies removed from on-hand inventory for their shipment are not included in the snapshot of the system.

4.4.4 Re-planning

To account for perturbations to the initial plan during the simulation run, the mixed integer program is re-solved every day of the simulation using a snapshot of the system state assembled from updates sent by all units in the system. As with decentralized control, requisitions that have begun the fulfillment process are not eligible to have their fulfillment plan changed. All other requisitions' plans are eligible to be changed. When a unit receives orders after the mathematical program is run, it clears its old action list and replaces it with a new action list based on the new orders.

An example of how a plan changes over time is shown in Table **4-4**. Only requisitions from unit OU 1-1 are considered, and all requisitions are for the same commodity. In this scenario, requisition priorities change at random, causing changes in the optimal assignment of units to fulfill requisitions. Requisitions are planned to be left unfulfilled (denoted by an 'x') or fulfilled internally by unit OU 1-1, externally by its direct superior DC 1, or externally by a lateral distribution center DC 2. Once requisitions begin the shipping process or internal consumption, they are no longer eligible to have their plans changed.

Table **4-4**: Example of plan changing over time.

Requisition Number	Initial Priority	Last Priority	Quantity	Execution Time	Quantity 0	1	2	3	4	5	6	7	8	9
40	0.86	0.56	0.99	Mon May 04 14:30:36 EDT 2009	OU 1-1	DC 1	**Ship**							
38	0.83	0.83	0.81	Mon May 04 16:42:42 EDT 2009	DC 1	DC 2	**Ship**							
13	0.79	0.79	0.53	Wed May 06 20:14:00 EDT 2009	OU 1-1	OU 1-1	DC 1	OU 1-1	OU 1-1	OU 1-1	**Cons.**			
37	0.72	0.50	0.08	Wed May 06 21:24:42 EDT 2009	OU 1-1	OU 1-1	DC 1	OU 1-1	OU 1-1	OU 1-1	**Cons.**			
34	0.72	0.72	0.37	Sun May 03 05:21:04 EDT 2009	OU 1-1	OU 1-1	OU 1-1	**Cons.**						
27	0.72	0.72	0.39	Wed May 06 05:56:02 EDT 2009	OU 1-1	OU 1-1	OU 1-1	OU 1-1	OU 1-1	OU 1-1	**Cons.**			
17	0.71	0.71	1.88	Wed May 06 13:36:14 EDT 2009	DC 1	DC 1	DC 1	DC 1	**Ship**					
16	0.71	0.71	0.13	Thu May 07 10:02:57 EDT 2009	OU 1-1	OU 1-1	DC 2	OU 1-1	OU 1-1	OU 1-1	OU 1-1	**Cons.**		
36	0.66	0.66	1.45	Wed May 06 02:47:59 EDT 2009	DC 1	DC 1	DC 1	**Ship**						
8	0.59	0.59	3.26	Mon May 04 15:10:46 EDT 2009	x	x	x	x	x	x	x			
24	0.57	0.23	0.53	Sat May 09 23:52:19 EDT 2009	DC 1	OU 1-1	OU 1-1	OU 1-1	OU 1-1	OU 1-1	OU 1-1	OU 1-1	OU 1-1	**Cons.**
10	0.50	0.84	0.81	Wed May 06 05:51:07 EDT 2009	OU 1-1	OU 1-1	DC 1	OU 1-1	OU 1-1	OU 1-1	**Cons.**			
30	0.38	0.38	3.17	Fri May 08 06:47:23 EDT 2009	x	x	x	x	x	x	x	x	x	x
20	0.35	0.35	0.16	Mon May 04 23:02:58 EDT 2009	DC 2	OU 1-1	DC 2	**Ship**						
14	0.23	0.23	0.25	Mon May 04 10:23:17 EDT 2009	OU 1-1	DC 1	**Ship**							
5	0.20	0.20	0.55	Wed May 06 08:56:22 EDT 2009	DC 1	x	OU 1-1	DC 1	**Ship**					
6	0.18	0.18	1.83	Mon May 04 07:02:24 EDT 2009	x	x	x	x	x	x	x			
3	0.14	0.13	0.97	Thu May 07 14:52:16 EDT 2009	x	OU 1-1	OU 1-1	DC 1	DC 1	**Ship**				
2	0.12	0.93	0.68	Tue May 05 10:23:32 EDT 2009	x	DC 1	OU 1-1	OU 1-1	OU 1-1	**Cons.**				
28	0.11	0.44	0.44	Sat May 09 12:05:54 EDT 2009	OU 1-1	x	OU 1-1	OU 1-1	OU 1-1	OU 1-1	OU 1-1	OU 1-1	OU 1-1	**Cons.**
18	0.10	0.10	0.73	Sun May 03 20:47:09 EDT 2009	x	x	x	x	x	x				
19	0.07	0.07	2.62	Sun May 03 04:44:00 EDT 2009	x	x	x	x	x	x				

4.4.5 Discussion of Model Assumptions

The mathematical model satisfies two important requirements to make it valuable for decision making and a generally accurate representation of the system. First, the constraints for supply and transportation availability for each unit are exact when generalized to a daily basis, except when queues form for transportation resources as discussed below. Second, each unit's cost for fulfilling a requisition, as well as the

penalty for requisition's being unfulfilled, conforms exactly to the utility function of the system.

The main loss of precision for the mathematical model comes from the use of time periods instead of continuous time. Although the shipping capacities are accurate for each day, the mathematical program's resulting orders are essentially that all shipments should go out at midnight on their designated days. This can cause minor penalties with earliness/lateness less than 1.0 day, as requisitions to be fulfilled "on-time" must be shipped at midnight on a day before or after their due date. However, because earliness and lateness is squared in the objective function, this should limit the cumulative effect of slightly early or late requisitions. Future work could analyze the results and solve an optimization problem to determine shipping times for each day. It should be noted that problems would arise from making time periods too small, as some shipments could have a volume that is too big to ship in a smaller time period.

Another potential source for error is that if a queue is present for transportation resources at a distribution center when the unit's resource status snapshot is taken. The math program will not know about these requisitions—to whom supplies had already been allocated from on-hand inventory—that are waiting for transportation resources. As a result, any shipments scheduled until the queue has time to diminish could face waiting times. A possible solution for the future would be to index the transportation capacity for each unit by time period. However, it is not very clear how it would affect the computational feasibility of the model under dynamic conditions.

Chapter 5

Experimental Design

The purpose of experimentation is to demonstrate numerically how the Distributed Resource Allocation Planning and Execution Simulation (DRAPES) operates and to test decentralized and centralized methods of decision making. This will illustrate the strengths and weaknesses of each approach, based on the results of different system configurations. This section defines the initial data, perturbation parameters, and the evaluation criteria used for each simulation run. Each run is evaluated on the criteria of demands fulfilled, transportation efficiency, and plan changeability. Stresses to the system during execution—categorized as perturbations—include the arrival of new tasks during execution, failure of resources, and change of task priorities.

5.1 Configuration Parameters

Two setups of two-echelon systems will be used and are differentiated by degrees of complexity. The first setup, Configuration 1, represents a smaller system while Configuration 2 represents a more complicated system. Each configuration will be run for four different scenarios, defined by the type of perturbations to which the system is subjected.

5.1.1 Unit Hierarchies and System Parameters

The primary differences between the two configurations are the number of units, number of tasks, and number of commodities. The basic differences between the two configurations are identified in Table **5-1**. The configurations are designed so that scarcity of transportation resources and supplies will prevent a minority of task demands—approximately one-quarter—from being successfully fulfilled. The "initial batch" of tasks is the set of tasks that are known to units at the beginning of a simulation run. Also, an operational unit's primary distribution center (DC) is its immediate superior and preferred supplier.

Table **5-1**: Comparison of system configurations.

Parameter	Configuration 1	Configuration 2
Number of Distribution Centers (DC)	2	2
Operational Units (OU) Per DC	1	2
Number of Types of Commodities	1	2
Tasks in Initial Batch	40	40
Simulation Duration (days)	10	20
Transportation Assets Per DC	4	4
Travel Time from Primary DC to OU (days)	0.25	0.25
Travel Time from Other DC to OU (days)	0.5	0.5

The values used for weights in determining the system's utility reflect the relative priority of each aspect of the system performance, and are the same for all configurations,

scenarios, and runs. They are presented in Table **5-2**. The unfulfilled requisition penalty equivalent is the earliness or lateness of a fulfilled requisition with the same priority that would result in the same penalty. For example, an unfulfilled requisition penalty equivalent of seven days would mean that an unfulfilled requisition would cause the same penalty as if it were fulfilled one week early or late.

Table **5-2**: Utility function parameters.

Parameter	Value
Weight for Earliness/Lateness Penalty	1.0
Unfulfilled Requisition Penalty Equivalent (days)	7.0
Weight for Travel Time Penalty	0.5
Weight for Plan Changeability Penalty	0.1

5.1.2 Initialization Information

Each unit begins each simulation run with an empty plan and all transportation resources available. The initial batch of tasks are generated and made known to their respective units at the simulation start time. The initial quantity of supplies possessed by a unit is a uniformly distributed random variable with parameters specified in the next section.

5.1.3 Task and Commodity Parameters

At simulation initialization and during the simulation run, tasks are generated and given values according to a set of parameters. Once a task is generated, each operational unit has an equal chance of being assigned a task. It is also assumed that all commodities have the same parameters, and that each task has a demand for all commodities. Parameters for all commodities in the system are shown in Table **5-3**. Demand quantities are randomly generated according to distribution parameters. The maximum quantity for a demand is the number of transportation assets in each scenario possessed by a distribution center.

The exogenous resupply rate defines a constant length of a time interval between exogenous resupply events. The first exogenous resupply event occurs one full time interval after the simulation start.

Initial inventories refer to the amount of a specific type of commodity that a unit has on-hand at the beginning of the simulation. This value is randomly generated from a uniform distribution, with the minimum and maximum values specified in Table **5-3**.

Table **5-3**: Commodity parameters.

Parameter	Config. 1	Config. 2
Mean Demand Quantity	1.0	1.0
Demand Quantity Distribution	exp.	exp.
Space Ratio (truckloads per unit)	1.0	1.0
Exogenous Resupply Quantity for DCs	2.0	2.0
Exogenous Resupply Quantity for OUs	1.0	1.0

Exogenous Resupply Rate for DCs (resupplies/day)	0.25	0.2
Exogenous Resupply Rate for OUs (resupplies/day)	0.25	0.1
Minimum Initial Inventory for DCs	3	3
Maximum Initial Inventory for DCs	6	5
Minimum Initial Inventory for OUs	1	1
Maximum Initial Inventory for OUs	3	2

Besides commodity information, other important information for task generation concerns how time information related to tasks—particularly due dates—is calculated. Demands have a one-day time window for delivery that expires one day before that task is to be executed and the demand consumed. The time window begins and ends with a required delivery date (RDD). The task execution time is randomly generated based on a uniformly distributed time interval. That time interval begins two days from the current time, so soonest possible earliest RDD would be the current time. The task execution time interval has a length equal to the simulation duration minus three days, which accounts for the two-day RDD buffer mentioned above and a one-day buffer at the end of the simulation. If a task execution time is randomly generated to be outside of the simulation start and end times, then that task is not distributed to the system. Task generation parameters are specified in Table **5-4**.

Table **5-4**: Task time-related parameters.

Parameter	Value
Execution Time – Latest RDD =	1.0 day

Latest RDD – Earliest RDD =	1.0 day
Execution Time: Earliest Possibility	Current Simulation Time + 2.0 days
Execution Time: Latest Possibility	Simulation Duration – 3.0 days

Finally, it is important to note that each task is assigned a priority that is randomly generated from a uniform distribution between 0 and 1.

5.1.4 Control Parameters

For centralized decision making, the only two important parameters for running the mathematical program are the run intervals and run time limits. In this system, runs will be made daily at midnight, and the run time limit will be determined based on a trade-off between run time and solution quality. Runs will be made on a single PC with a 2.13 GHz processor and 2.0 GB of RAM.

For decentralized decision making, the only parameter that needs to be specified relates to the trade-off between supportability risk and earliness/lateness. The value of this trade-off parameter used in bid evaluation is 1.0.

5.1.5 Scenario Parameters

Four different scenarios were designed to test the performance of each control method for each configuration. The scenarios represent different types of stresses being placed on resource allocation.

In Scenario 1, the system only needs to satisfy the initial batch of tasks given to units in the system. For the centralized and decentralized approaches, all plans are executed as planned—subject to slight imprecision due to approximations of continuous time—without any perturbations. Therefore, only an initial plan is generated and tested for each control method in this scenario.

In Scenario 2, new tasks are generated by the system as Poisson arrivals. For Configuration 1, the arrival rate is 5 per day per unit. For Configuration 2, the arrival rate is 1 per day per unit. The execution time of the new tasks is randomly generated using the lower bound as the current time and the upper bound as the total simulation duration added to the current time. Therefore, it is possible to have an execution time generated that would occur after the simulation end time. If that is the case, the task is ignored and not disseminated to the system.

In Scenario 3, the system experiences the stress of resource failures, in which a certain quantity of a certain type of supply is made unavailable to a unit. Resource failures of commodities require the immediate removal from the inventory of a unit a specified number of that commodity, or the removal of all supplies of that type from the unit's inventory if there are insufficient supplies. The unit subjected to the failure is selected at random from the set of units, each having equal likelihood. After the unit is selected, the resource is selected at random from the set of all commodities, each item in the set having equal chance. Scenario 3 parameters are listed in Table **5-3**.

Table **5-5**: Scenario 3 parameters.

Parameter	Value

Rate of Failures (per unit per day)	0.2
Lost Resource Quantity	0.5
Lost Resource Quantity Distribution	exponential

In Scenario 4, tasks priorities change at random as a stochastic process. During the simulation, the task priority change events occur at the rate of 2 per day. When such an event occurs, a mission task is selected at random from the subset of mission tasks that have not yet been executed, with each mission task having equal likelihood. Units in the system are notified immediately of this priority change.

5.2 Run Parameters

Simulation runs will be made for 2 configurations x 4 scenarios x 2 control methods x *n* replications. Each unique combination of a configuration, scenario, and control method is known as a "condition" and is replicated 20 times for Configuration 1 and 6 times for Configuration 2 of Scenario 1. For all other scenarios, 10 replications were made for Configuration 1 scenarios, and 3 replications were made for Configuration 2 scenarios. The different conditions are enumerated in Table **5-6**. Each replication is given a different seed for generating random numbers so that the tasks generated, the units' initial inventories, and the perturbations are the same across conditions.

Table **5-6**: Run conditions.

	Configuration 1: Small System		Configuration 2: Large System	
	Centralized	Decentralized	Centralized	Decentralized
Scenario 1: No Perturbations	Condition 1C1	Condition 1D1	Condition 2C1	Condition 2D1
Scenario 2: New Tasks	Condition 1C2	Condition 1D2	Condition 2C2	Condition 2D2
Scenario 3: Resource Loss	Condition 1C3	Condition 1D3	Condition 2C3	Condition 2D3
Scenario 4: Priority Changes	Condition 1C4	Condition 1D4	Condition 2C4	Condition 2D4

Chapter 6

Results and Discussion

6.1 Mathematical Program Runtime

A replication for each Scenario 1 condition was run to estimate an appropriate time limit for running the CPLEX solver. A trade-off naturally exists between run time and solution quality, with diminishing improvements in solution quality as run time lengthens. Figure **6-1** shows the output from the CPLEX solver for replication 1 of Condition 1C1, which has the simplest configuration with 40 requisitions, 2 distribution centers with 1 operational unit each, and one commodity. Based on this solution, a run time limit of 120 seconds was chosen.

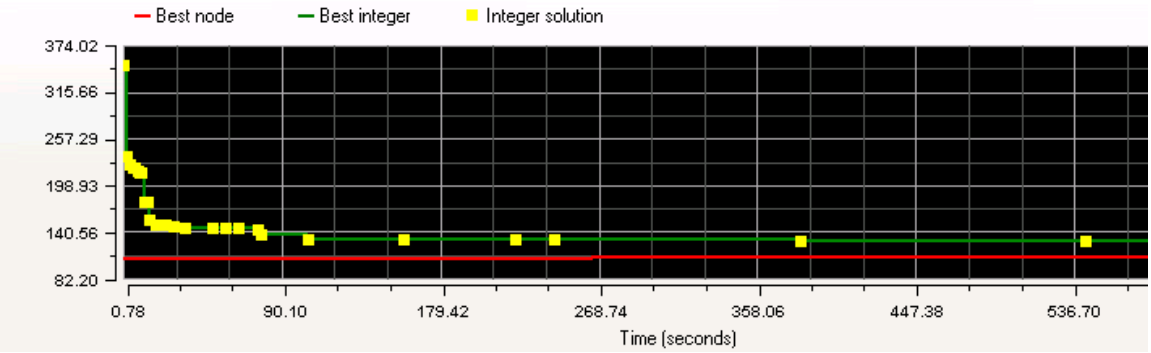

Figure **6-1**: Solution quality vs. run time for condition 1C1.

Similarly, Figure **6-2** has the CPLEX output for replication 1 of Condition 2C1, which has a more complicated configuration with 100 requisitions, 2 distribution centers with 2 operational units each, and two commodities.

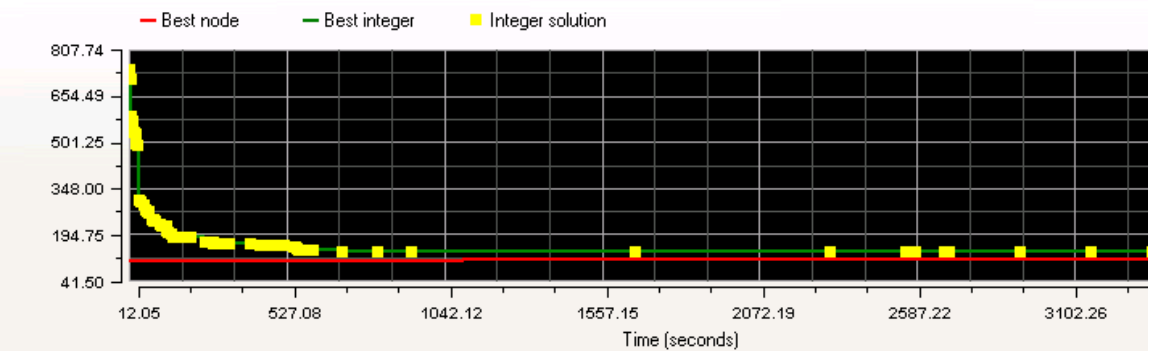

Figure **6-2**: Solution quality vs. run time for condition 2C1.

Based on the CPLEX output, a run time limit of 10 minutes was chosen for all Configuration 2 simulation runs.

6.2 Demand Fulfillment Comparison

To illustrate differences in the decentralized iterative priority approach and the centralized mixed-integer program approach, the solutions are compared in more detail. To do so, graphs are presented to illustrate differences in which demands are fulfilled at what earliness or lateness for replication 2 of Conditions 2D1 and 2C1.

Figure **6-3** shows the resulting decision for each requisition using the decentralized approach. Each mark on the chart represents a requisition for one of 80 demands generated at the beginning of the simulation run. These results are taken from simulation statistics at run end, so it also helps verify the iterative planning approach for decentralized decision-making. Not surprisingly, all high-priority requisitions—which were scheduled with a relatively empty plan—were delivered on-time. The plan began to fill as lower priority requisitions were bid, and some could only be fulfilled late or not at

all. The eleven unfulfilled requisitions are noted as marks on the priority axis, although in some cases marks represent two requisitions for the same mission task. As part of the same mission task, these requisitions had the same priority.

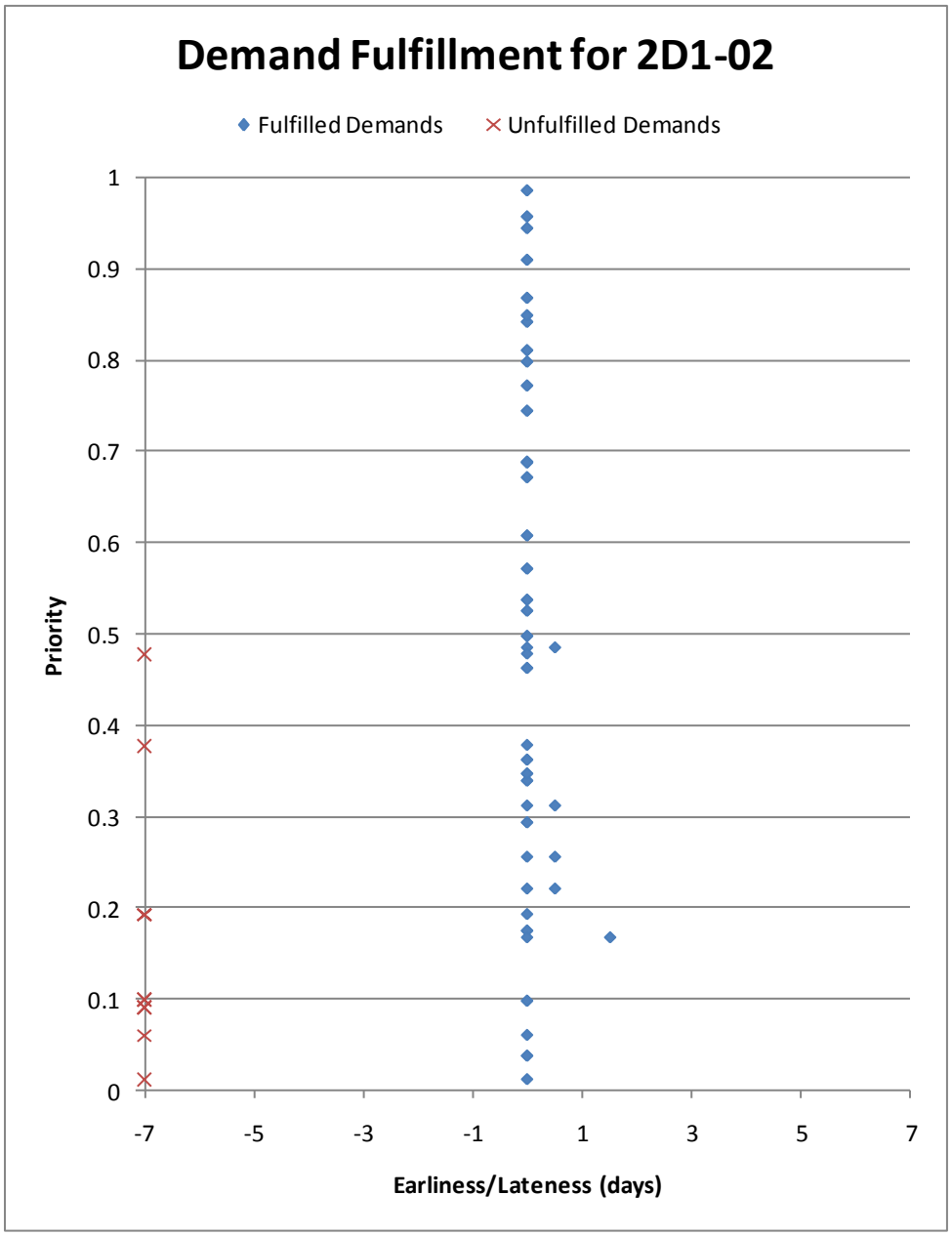

Figure **6-3**: Example of demand fulfillment results for decentralized approach.

Because of the bidding process, the only possible fulfillment times were 0.5 days early or late and 1.5 days early or late. This is due to the requested delivery time being the mean of early and late required delivery dates. Because each demand has a one-day delivery window, a bid filled one day after the requested delivery time would be 0.5 days outside the delivery window.

Figure **6-4** shows the centralized solution based on the mixed-integer program to the same set of demands with the same initial conditions for the simulation. Overall this solution saw only 6 demands left unfulfilled. In this particular replication, no requisitions were fulfilled early, although some were early for other replications. Compared to the decentralized solution, some requisitions with high priorities were allowed to be fulfilled late. This was presumably done to increase the objective function by allowing shipments to be made to lower priority requisitions that otherwise would not have been made.

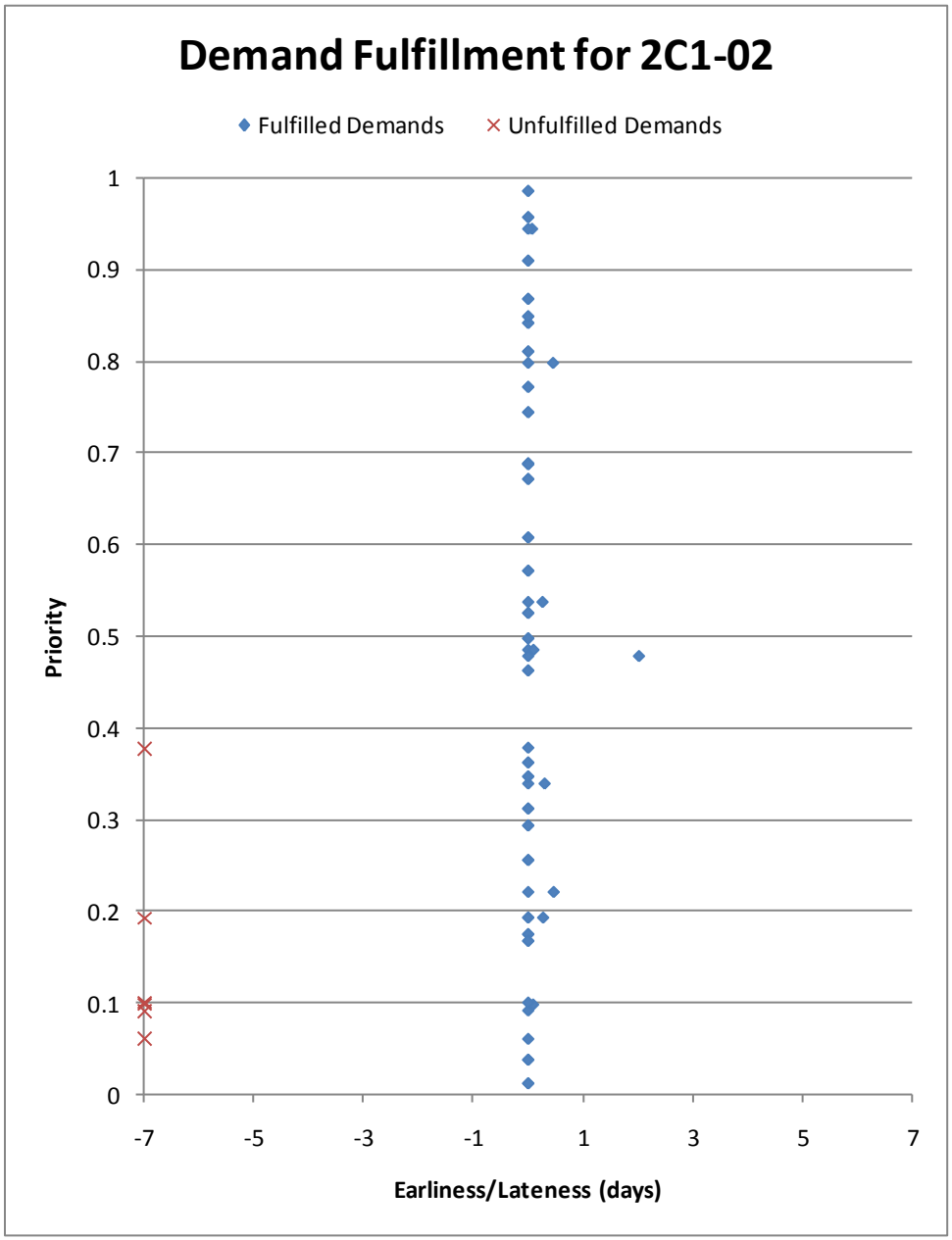

Figure **6-4**: Example of demand fulfillment results for centralized approach.

6.3 Plan Results

Run results for replications of the 1C1 and 1D1 conditions are presented in Table **6-1** and summarized in Table **6-2**. With paired replications, the centralized system on average fulfilled 11% more, or 4.45 requisitions, than the decentralized system. This contributed the majority of the mean difference in the DRAPES objective of 65.20. In 19 of 20 runs, the centralized approach fulfilled more requisitions than the decentralized approach. In that other run, they fulfilled the same number of requisitions, but the centralized approach still had a lower objective due to fulfilling higher priority requisition and using transportation more efficiently.

Confidence intervals constructed from paired comparisons using the t-distribution show that the centralized approach provides a better solution than the decentralized approach. For the percentage of unfulfilled requisitions, the difference between the centralized and decentralized approach is with 99% confidence. The confidence limits on the true mean difference in percentage of unfulfilled requisitions are -0.076 and 0.147. Likewise, the DRAPES objective is also significantly better for the centralized approach with 99% confidence and limits of -98.4 and -32.0.

The metrics for CPLEX objective and CPLEX unfulfilled refer to the solution directly received from the mathematical program. The DRAPES objective is the score for that solution after the plan was simulated. That the number of unfulfilled requisitions is the same for the simulation as the CPLEX solution verifies the accuracy of modeling the system as a mixed-integer program. The slight variations in the objectives are primarily due to different delivery window penalties and transportation modeling. The

mixed-integer program uses a point due date instead of a delivery window but uses the same penalty function for earliness or lateness. For the mixed-integer program, the penalty for shipping a requisition is not discounted when batched, influencing the DRAPES objective to be potentially less than the CPLEX objective. However, the simulation does not allow two partial demands to be shipped on the same truck, influencing the DRAPES objective to be potentially greater than the CPLEX objective.

The results also reveal an important characteristic of the decentralized and centralized approaches—that the centralized approach uses its transportation assets more efficiently. The metric, sum of travel times, refers to the total asset-days summed over all of its transportation resources. On average by replication, the centralized approach fulfilled 0.12 more requisitions per asset-day. This is due to constraints in the mathematical model that account for combining requisitions going to the same destination. No intentional batching occurs in the decentralized strategy.

Table **6-1**: Results for 1C1 and 1D1 runs.

Replication		Fulfilled	Unfulfilled	% Fulfilled	Internally Fulfilled	Externally Fulfilled	% Internally Fulfilled	Sum of Travel Time (days)	Total Earliness (days)	Total Lateness (days)	DRAPES Objective	CPLEX Objective	CPLEX Unfulfilled
	Centralized	30	10	75.0%	19	11	63.3%	9	0.0	5.2	121.9	131.6	10
	Decentralized	27	13	67.5%	17	10	63.0%	11	0.0	0.5	182.2		
1	Difference	3	-3	7.5%	2	1	0.4%	-2	0.0	4.7	-60.3		
	Centralized	36	4	90.0%	21	15	58.3%	17	0.4	9.7	66.2	72.4	4
	Decentralized	30	10	75.0%	21	9	70.0%	11.5	0.0	0.4	143.3		
2	Difference	6	-6	15.0%	0	6	-11.7%	5.5	0.4	9.2	-77.2		
	Centralized	29	11	72.5%	15	14	51.7%	16.5	0.7	6.1	166.9	155.3	11
	Decentralized	25	15	62.5%	13	12	52.0%	16	1.5	3.5	198.4		
3	Difference	4	-4	10.0%	2	2	-0.3%	0.5	-0.8	2.6	-31.5		
	Centralized	30	10	75.0%	15	15	50.0%	12	0.4	4.9	127.4	134.8	10
	Decentralized	24	16	60.0%	9	15	37.5%	13	0.0	4.4	215.5		
4	Difference	6	-6	15.0%	6	0	12.5%	-1	0.4	0.5	-88.1		
	Centralized	29	11	72.5%	12	17	41.4%	16.5	0.3	12.6	234.2	241.0	11
	Decentralized	19	21	47.5%	10	9	52.6%	12.5	0.0	3.0	487.7		
5	Difference	10	-10	25.0%	2	8	-11.3%	4	0.3	9.6	-253.5		
	Centralized	33	7	82.5%	19	14	57.6%	16	0.0	5.8	45.1	54.0	7
	Decentralized	28	12	70.0%	15	13	53.6%	16.5	0.0	2.5	80.0		
6	Difference	5	-5	12.5%	4	1	4.0%	-0.5	0.0	3.2	-34.9		
	Centralized	29	11	72.5%	15	14	51.7%	17	0.0	3.1	136.8	142.4	11
	Decentralized	26	14	65.0%	19	7	73.1%	13	0.0	3.0	190.6		
7	Difference	3	-3	7.5%	-4	7	-21.4%	4	0.0	0.1	-53.8		
	Centralized	27	13	67.5%	13	14	48.1%	13	0.0	7.4	158.5	167.8	13
	Decentralized	24	16	60.0%	11	13	45.8%	12.5	0.0	1.5	232.7		
8	Difference	3	-3	7.5%	2	1	2.3%	0.5	0.0	5.9	-74.2		
	Centralized	30	10	75.0%	11	19	36.7%	15	0.0	7.7	173.3	183.0	10
	Decentralized	27	13	67.5%	10	17	37.0%	16.5	0.0	5.0	234.8		
9	Difference	3	-3	7.5%	1	2	-0.4%	-1.5	0.0	2.7	-61.5		
	Centralized	31	9	77.5%	15	16	48.4%	12.5	0.0	11.9	61.0	71.6	9
	Decentralized	27	13	67.5%	19	8	70.4%	9.5	0.0	3.5	99.6		
10	Difference	4	-4	10.0%	-4	8	-22.0%	3	0.0	8.4	-38.7		
	Centralized	31	9	77.5%	16	15	51.6%	17.5	0.0	8.3	39.6	43.8	9
	Decentralized	31	9	77.5%	13	18	41.9%	17	0.5	4.0	45.4		
11	Difference	0	0	0.0%	3	-3	9.7%	0.5	-0.5	4.3	-5.8		
	Centralized	32	8	80.0%	16	16	50.0%	17	0.0	24.9	175.3	157.4	8
	Decentralized	25	15	62.5%	13	12	52.0%	13.5	0.0	2.0	217.5		
12	Difference	7	-7	17.5%	3	4	-2.0%	3.5	0.0	22.9	-42.2		
	Centralized	31	9	77.5%	15	16	48.4%	14.5	0.1	7.8	217.7	228.7	9
	Decentralized	24	16	60.0%	12	12	50.0%	14.5	0.5	3.5	336.4		
13	Difference	7	-7	17.5%	3	4	-1.6%	0	-0.5	4.4	-118.7		
	Centralized	32	8	80.0%	18	14	56.3%	14	0.0	5.8	62.7	72.7	8
	Decentralized	28	12	70.0%	15	13	53.6%	13	0.0	2.0	132.4		
14	Difference	4	-4	10.0%	3	1	2.7%	1	0.0	3.8	-69.7		
	Centralized	33	7	82.5%	19	14	57.6%	16.5	0.0	10.7	138.5	146.1	7
	Decentralized	27	13	67.5%	14	13	51.9%	12.5	0.0	1.0	231.9		
15	Difference	6	-6	15.0%	5	1	5.7%	4	0.0	9.7	-93.4		
	Centralized	31	9	77.5%	17	14	54.8%	12	0.0	6.6	84.4	94.4	9
	Decentralized	30	10	75.0%	10	20	33.3%	18	0.0	4.4	125.0		
16	Difference	1	-1	2.5%	7	-6	21.5%	-6	0.0	2.2	-40.6		
	Centralized	35	5	87.5%	19	16	54.3%	15.5	0.4	10.0	55.3	61.4	5
	Decentralized	30	10	75.0%	11	19	36.7%	16.5	0.5	3.0	113.5		
17	Difference	5	-5	12.5%	8	-3	17.6%	-1	-0.1	7.0	-58.2		
	Centralized	32	8	80.0%	16	16	50.0%	14.5	0.0	12.0	92.0	95.9	8
	Decentralized	27	13	67.5%	17	10	63.0%	11.5	0.0	2.0	140.5		
18	Difference	5	-5	12.5%	-1	6	-13.0%	3	0.0	10.0	-48.5		
	Centralized	36	4	90.0%	19	17	52.8%	14	0.0	10.7	52.4	54.2	4
	Decentralized	33	7	82.5%	20	13	60.6%	11.5	0.0	3.5	64.4		
19	Difference	3	-3	7.5%	-1	4	-7.8%	2.5	0.0	7.2	-12.1		
	Centralized	30	10	75.0%	17	13	56.7%	15.5	0.0	9.5	106.0	115.4	10
	Decentralized	26	14	65.0%	15	11	57.7%	16	0.0	4.5	147.1		
20	Difference	4	-4	10.0%	2	2	-1.0%	-0.5	0.0	5.0	-41.1		

Table **6-2**: Summary of results for 1C1 and 1D1 runs.

	Mean			Minimum			Maximum			Standard Deviation		
	1C1	1D1	Difference by Run	1C1	1D1	Difference by Run	1C1	1D1	Difference by Run	1C1	1D1	Difference by Run
Fulfilled	31.35	26.90	4.45	27.00	19.00	0.00	36.00	33.00	10.00	2.37	3.09	2.24
Unfulfilled	8.65	13.10	-4.45	4.00	7.00	-10.00	13.00	21.00	0.00	2.37	3.09	2.24
% Fulfilled	0.78	0.67	0.11	0.68	0.48	0.00	0.90	0.83	0.25	0.06	0.08	0.06
Internally Fulfilled	16.35	14.20	2.15	11.00	9.00	-4.00	21.00	21.00	8.00	2.60	3.65	3.15
Externally Fulfilled	15.00	12.70	2.30	11.00	7.00	-6.00	19.00	20.00	8.00	1.72	3.60	3.71
% Internally Fulfilled	0.52	0.53	-0.01	0.37	0.33	-0.22	0.63	0.73	0.22	0.06	0.12	0.11
Sum of Travel Time (days)	14.78	13.80	0.98	9.00	9.50	-6.00	17.50	18.00	5.50	2.20	2.40	2.74
Total Earliness	0.12	0.15	-0.03	0.00	0.00	-0.76	0.74	1.50	0.43	0.21	0.37	0.27
Total Lateness	9.01	2.85	6.16	3.05	0.41	0.10	24.86	4.98	22.87	4.57	1.34	4.96
DRAPES Objective	115.75	180.95	-65.20	39.61	45.37	-253.52	234.15	487.67	-5.76	58.60	100.62	51.92
CPLEX Objective	121.19			43.81			240.96			57.53		
CPLEX Unfulfilled	8.65			4.00			13.00			2.37		

Six replications were also made for the 2C1 and 2D1 conditions with similar results, shown in Table **6-3**. These results for paired comparisons of the solution objective are also significantly better for the centralized approach than the decentralized approach with 99% confidence. Like the 1C1 and 2C1 runs, the centralized approach always fulfills at least as many requisitions as the decentralized approach.

Table **6-3**: Results for 2C1 and 2D1 runs.

Replication		Fulfilled	Unfulfilled	% Fulfilled	Internally Fulfilled	Externally Fulfilled	% Internally Fulfilled	Sum of Travel Time (days)	Total Earliness (days)	Total Lateness (days)	DRAPES Objective	CPLEX Objective	CPLEX Unfulfilled
	Centralized	65	15	81.3%	31	34	47.7%	31.5	0.0	1.6	134.0	138.3	15
	Decentralized	62	18	77.5%	30	32	48.4%	31.5	1.0	4.0	189.2		
1	Difference	3	-3	3.8%	1	2	-0.7%	0	-1.0	-2.4	-55.2		
	Centralized	74	6	92.5%	38	36	51.4%	36	0.0	4.0	65.5	70.0	6
	Decentralized	69	11	86.3%	44	25	63.8%	32.5	0.0	3.5	109.8		
2	Difference	5	-5	6.3%	-6	11	-12.4%	3.5	0.0	0.5	-44.3		
	Centralized	68	12	85.0%	35	33	51.5%	32	2.5	18.2	263.4	283.3	12
	Decentralized	62	18	77.5%	35	27	56.5%	26.5	1.5	7.0	366.1		
3	Difference	6	-6	7.5%	0	6	-5.0%	5.5	1.0	11.2	-102.7		
	Centralized	68	12	85.0%	34	34	50.0%	33	1.8	6.6	126.2	144.2	12
	Decentralized	61	19	76.3%	34	27	55.7%	30	0.0	2.5	212.9		
4	Difference	7	-7	8.8%	0	7	-5.7%	3	1.8	4.1	-86.7		
	Centralized	74	6	92.5%	41	33	55.4%	36	1.2	10.6	50.1	64.9	6
	Decentralized	69	11	86.3%	40	29	58.0%	31	0.5	2.0	82.0		
5	Difference	5	-5	6.3%	1	4	-2.6%	5	0.7	8.6	-31.9		
	Centralized	67	13	83.8%	38	29	56.7%	31.5	0.0	7.3	133.2	144.6	13
	Decentralized	63	17	78.8%	38	25	60.3%	27.5	0.0	2.5	166.3		
6	Difference	4	-4	5.0%	0	4	-3.6%	4	0.0	4.8	-33.2		

For Scenario 2, in which new tasks were introduced during the simulation, units had to adjust their plans in response to the addition of between twenty-seven and thirty-nine requisitions. The mean percentage of requisitions fulfilled in Configuration 1 was 72.9% for the centralized control method and 65.3% for decentralized control. With this mean difference of 7.5% requisitions fulfilled, a t test on paired comparisons shows the difference to be significant with confidence of 99%. The full results for Configuration 1-Scenario 2 tests are shown in Table **6-4**. On average, requisitions' planned source unit changed sixteen more times with the centralized method than the decentralized method.

Furthermore, the centralized planning method uses its transportation resources more efficiently than the decentralized iterative planning approach. The difference in the number of requisitions fulfilled internally is significantly higher for the decentralized approach than the centralized approach. This could indicate limitations of the iterative priority approach for scheduling transportation. A two-step approach to scheduling transportation could provide a useful alternative, in which reservations are made on daily transportation capacities and subsequently assigned ship times rather than immediately scheduling ship times. This approach could improve utilization of transportation assets.

Table **6-4**: Results for 1C2 and 1D2 runs.

Replication		Fulfilled	Unfulfilled	% Fulfilled	Internally Fulfilled	Externally Fulfilled	% Internally Fulfilled	Sum of Travel Time (days)	Plan Change Penalty	Total Earliness (days)	Total Lateness (days)	DRAPES Objective
	Centralized	60	19	75.9%	23	37	38.3%	14.5	42	0.8	5.9	381.7
	Decentralized	52	27	65.8%	26	26	50.0%	14.5	28	0.1	3.2	504.4
1	Difference	8	-8	10.1%	-3	11	-11.7%	0.0	14	0.8	2.7	-122.7
	Centralized	58	9	86.6%	21	37	36.2%	18.0	64	0.0	15.0	159.8
	Decentralized	53	14	79.1%	23	30	43.4%	19.0	21	0.0	8.1	212.1
2	Difference	5	-5	7.5%	-2	7	-7.2%	-1.0	43	0.0	6.9	-52.3
	Centralized	53	25	67.9%	18	35	34.0%	11.5	48	2.1	3.1	413.1
	Decentralized	47	31	60.3%	29	18	61.7%	14.5	39	0.2	2.5	555.7
3	Difference	6	-6	7.7%	-11	17	-27.7%	-3.0	9	1.9	0.6	-142.6
	Centralized	53	18	74.6%	21	32	39.6%	14.0	49	0.1	4.5	278.4
	Decentralized	46	25	64.8%	24	22	52.2%	16.0	40	0.0	3.5	377.2
4	Difference	7	-7	9.9%	-3	10	-12.6%	-2.0	9	0.1	1.0	-98.8
	Centralized	51	25	67.1%	16	35	31.4%	15.5	65	0.0	9.9	509.6
	Decentralized	44	32	57.9%	16	28	36.4%	12.0	35	0.0	7.3	676.1
5	Difference	7	-7	9.2%	0	7	-5.0%	3.5	30	0.0	2.6	-166.5
	Centralized	54	23	70.1%	23	31	42.6%	17.0	53	0.0	0.7	238.5
	Decentralized	50	27	64.9%	20	30	40.0%	20.0	57	0.5	5.5	325.3
6	Difference	4	-4	5.2%	3	1	2.6%	-3.0	-4	-0.5	-4.9	-86.8
	Centralized	54	20	73.0%	18	36	33.3%	20.0	68	0.0	2.8	316.2
	Decentralized	51	23	68.9%	27	24	52.9%	17.5	41	0.0	5.0	372.8
7	Difference	3	-3	4.1%	-9	12	-19.6%	2.5	27	0.0	-2.2	-56.6
	Centralized	51	19	72.9%	24	27	47.1%	14.0	52	0.0	10.8	288.2
	Decentralized	46	24	65.7%	21	25	45.7%	16.0	36	0.5	4.0	350.4
8	Difference	5	-5	7.1%	3	2	1.4%	-2.0	16	-0.5	6.8	-62.2
	Centralized	48	27	64.0%	16	32	33.3%	16.5	42	1.5	6.6	521.8
	Decentralized	39	36	52.0%	15	24	38.5%	18.0	39	0.0	6.8	689.8
9	Difference	9	-9	12.0%	1	8	-5.1%	-1.5	3	1.5	-0.1	-167.9
	Centralized	56	17	76.7%	25	31	44.6%	14.0	45	1.1	2.2	204.8
	Decentralized	54	19	74.0%	25	29	46.3%	16.5	32	0.5	6.0	250.0
10	Difference	2	-2	2.7%	0	2	-1.7%	-2.5	13	0.6	-3.7	-45.2
	Centralized	53.8	20.2	72.9%	20.5	33.3	38.0%	15.5	52.8	0.6	6.2	331.2
	Decentralized	48.2	25.8	65.3%	22.6	25.6	46.7%	16.4	36.8	0.2	5.2	431.4
Mean	Difference	5.6	-5.6	7.5%	-2.1	7.7	-8.7%	-0.9	16	0.4	1.0	-100.1

The results for the three replications of Configuration 2-Scenario 2 are not significant with 90% confidence, and are shown in Table **6-5**. However the mean differences in fraction of requisitions fulfilled, travel time, and plan changes exhibit the same trends as Configuration 1.

Table **6-5**: Results for 2C2 and 2D2 runs.

Replication		Fulfilled	Unfulfilled	% Fulfilled	Internally Fulfilled	Externally Fulfilled	% Internally Fulfilled	Sum of Travel Time (days)	Plan Change Penalty	Total Earliness (days)	Total Lateness (days)	DRAPES Objective
	Centralized	113	23	83.1%	41	72	36.3%	30.0	232	1.4	2.1	452.0
	Decentralized	99	37	72.8%	54	45	54.5%	37.0	81	0.5	3.0	595.3
1	Difference	14	-14	10.3%	-13	27	-18.3%	-7.0	151	0.9	-1.0	-143.3
	Centralized	140	24	85.4%	61	79	43.6%	38.0	286	1.1	10.1	286.7
	Decentralized	137	27	83.5%	80	57	58.4%	46.5	89	0.5	15.4	296.0
2	Difference	3	-3	1.8%	-19	22	-14.8%	-8.5	197	0.6	-5.3	-9.3
	Centralized	105	41	71.9%	36	69	34.3%	36.0	209	1.3	8.8	700.8
	Decentralized	99	47	67.8%	50	49	50.5%	39.0	70	0.5	6.9	867.1
3	Difference	6	-6	4.1%	-14	20	-16.2%	-3.0	139	0.8	1.9	-166.3
	Centralized	119.3	29.3	80.1%	46	73.3	38.0%	34.7	242	1.3	7.0	479.8
	Decentralized	111.7	37.0	74.7%	61.3	50.3	54.5%	40.8	80	0.5	8.5	586.1
Mean	Difference	7.7	-7.7	5.4%	-15	23	-16.4%	-6.2	162	0.8	-1.5	-106.3

Scenario 3—the resource loss scenario—shows similar comparative solution performance to Scenario 1 for both Configurations 1 and 2. As expected, the loss of supplies negatively impacted the average number of requisitions fulfilled by over 2 requisitions for both centralized and decentralized approach of Configuration 1. The differences between centralized and decentralized approaches for requisitions fulfilled are

significant with 99% confidence for Configuration 1 and 90% confidence for Configuration 2. These results are shown in Table **6-6** and Table **6-7**.

Table **6-6**: Results for 1C3 and 1D3 runs.

Replication		Fulfilled	Unfulfilled	% Fulfilled	Internally Fulfilled	Externally Fulfilled	% Internally Fulfilled	Sum of Travel Time (days)	Plan Change Penalty	Total Earliness (days)	Total Lateness (days)	DRAPES Objective
	Centralized	30	10	75.0%	10	20	33.3%	14.0	49	1.3	9.1	169.8
	Decentralized	26	14	65.0%	13	13	50.0%	13.5	7	0.0	5.8	221.9
1	Difference	4	-4	10.0%	-3	7	-16.7%	0.5	42	1.3	3.3	-52.1
	Centralized	36	4	90.0%	13	23	36.1%	15.5	60	0.0	7.6	103.4
	Decentralized	30	10	75.0%	19	11	63.3%	13.0	8	0.0	4.9	137.1
2	Difference	6	-6	15.0%	-6	12	-27.2%	2.5	52	0.0	2.7	-33.7
	Centralized	28	12	70.0%	10	18	35.7%	15.5	32	0.1	2.1	167.4
	Decentralized	22	18	55.0%	11	11	50.0%	13.5	5	0.5	2.5	266.7
3	Difference	6	-6	15.0%	-1	7	-14.3%	2.0	27	-0.4	-0.4	-99.3
	Centralized	29	11	72.5%	11	18	37.9%	14.0	35	0.1	5.4	183.5
	Decentralized	23	17	57.5%	10	13	43.5%	11.0	4	0.0	7.3	257.2
4	Difference	6	-6	15.0%	1	5	-5.5%	3.0	31	0.1	-1.9	-73.7
	Centralized	28	12	70.0%	10	18	35.7%	17.0	39	0.0	27.0	309.8
	Decentralized	20	20	50.0%	9	11	45.0%	12.0	12	0.0	3.5	478.3
5	Difference	8	-8	20.0%	1	7	-9.3%	5.0	27	0.0	23.5	-168.5
	Centralized	31	9	77.5%	16	15	51.6%	17.0	37	0.0	5.3	138.4
	Decentralized	24	16	60.0%	12	12	50.0%	12.5	13	0.0	5.4	203.7
6	Difference	7	-7	17.5%	4	3	1.6%	4.5	24	0.0	0.0	-65.3
	Centralized	26	14	65.0%	10	16	38.5%	20.0	60	0.0	1.7	230.4
	Decentralized	25	15	62.5%	16	9	64.0%	15.0	7	0.0	2.2	233.6
7	Difference	1	-1	2.5%	-6	7	-25.5%	5.0	53	0.0	-0.4	-3.2
	Centralized	27	13	67.5%	11	16	40.7%	13.5	44	0.0	13.1	189.0
	Decentralized	22	18	55.0%	11	11	50.0%	10.0	6	0.0	0.7	295.5
8	Difference	5	-5	12.5%	0	5	-9.3%	3.5	38	0.0	12.3	-106.5
	Centralized	29	11	72.5%	9	20	31.0%	17.5	49	0.2	5.1	212.8
	Decentralized	23	17	57.5%	11	12	47.8%	12.5	11	0.0	7.8	313.3
9	Difference	6	-6	15.0%	-2	8	-16.8%	5.0	38	0.2	-2.7	-100.5
	Centralized	28	12	70.0%	12	16	42.9%	17.0	63	0.0	14.1	115.3
	Decentralized	25	15	62.5%	18	7	72.0%	11.5	6	0.0	2.3	123.6
10	Difference	3	-3	7.5%	-6	9	-29.1%	5.5	57	0.0	11.9	-8.3
	Centralized	29.2	10.8	73.0%	11.2	18	38.4%	16.1	46.8	0.2	9.0	182.0
	Decentralized	24	16	60.0%	13	11	53.6%	12.5	7.9	0.1	4.2	253.1
Mean	Difference	5.2	-5.2	13.0%	-1.8	7	-15.2%	3.7	38.9	0.1	4.8	-71.1

Table **6-7**: Results for 2C3 and 2D3 runs.

Replication		Fulfilled	Unfulfilled	% Fulfilled	Internally Fulfilled	Externally Fulfilled	% Internally Fulfilled	Sum of Travel Time (days)	Plan Change Penalty	Total Earliness (days)	Total Lateness (days)	DRAPES Objective
	Centralized	63	17	78.8%	25	38	39.7%	36.0	169	5.3	8.6	269.3
	Decentralized	56	24	70.0%	27	29	48.2%	35.0	33	0.5	6.1	362.6
1	Difference	7	-7	8.8%	-2	9	-8.5%	1.0	136	4.8	2.5	-93.3
	Centralized	68	12	85.0%	29	39	42.6%	35.0	192	0.0	14.8	236.7
	Decentralized	64	16	80.0%	37	27	57.8%	32.0	22	0.5	5.7	274.0
2	Difference	4	-4	5.0%	-8	12	-15.2%	3.0	170	-0.5	9.1	-37.2
	Centralized	66	14	82.5%	26	40	39.4%	38.5	180	3.4	40.5	332.1
	Decentralized	56	24	70.0%	30	26	53.6%	26.5	18	0.4	4.7	479.2
3	Difference	10	-10	12.5%	-4	14	-14.2%	12.0	162	3.0	35.8	-147.1
	Centralized	65.7	14.3	82.1%	26.7	39	40.6%	36.5	180	2.9	21.3	279.4
	Decentralized	58.7	21.3	73.3%	31.3	27.3	53.2%	31.2	24.3	0.5	5.5	371.9
Mean	Difference	7	-7	8.8%	-4.7	11.7	-12.6%	5.3	156	2.4	15.8	-92.5

For Scenario 4, task priority changes caused a slight decrease in the mean number of requisitions fulfilled and an increase in the mean DRAPES objective. This shows that, on average, the changing priorities are a stress to the system, as would be expected. Results from these replications are displayed in Table **6-8** and Table **6-9**.

One simulation run—replication 2 of conditions 2C4 and 2D4—merits attention, as it was the only replication in which the decentralized approach produced a better solution than the centralized approach. Their DRAPES objective values were 147.7 and 165.9, respectively. For those runs, the difference primarily came from two sources. The centralized approach had 176 more plan changes than the decentralized approach, which

accounts for a score difference of 17.6. Also, one requisition with changed from low priority to high priority only nine minutes before its execution time toward the end of the simulation run, threatening a penalty of 42.6. Insufficient supplies remained at that point in the system for the centralized approach to re-plan to fulfill that requisition. However, the decentralized system had resources available and was able to re-plan to find a source and successfully fulfill that requisition.

Table **6-8**: Results for 1C4 and 1D4 runs.

Replication		Fulfilled	Unfulfilled	% Fulfilled	Internally Fulfilled	Externally Fulfilled	% Internally Fulfilled	Sum of Travel Time (days)	Plan Change Penalty	Total Earliness (days)	Total Lateness (days)	DRAPES Objective
	Centralized	31	9	77.5%	17	14	54.8%	11.5	47	0.0	5.0	120.0
	Decentralized	25	15	62.5%	16	9	64.0%	10.5	18	0.5	1.0	227.1
1	Difference	6	-6	15.0%	1	5	-9.2%	1.0	29	-0.5	4.0	-107.1
	Centralized	36	4	90.0%	20	16	55.6%	14.5	30	0.0	5.0	88.8
	Decentralized	27	13	67.5%	18	9	66.7%	11.5	23	0.0	4.8	209.1
2	Difference	9	-9	22.5%	2	7	-11.1%	3.0	7	0.0	0.2	-120.3
	Centralized	30	10	75.0%	11	19	36.7%	18.0	44	1.0	14.0	160.8
	Decentralized	22	18	55.0%	11	11	50.0%	13.0	11	0.0	3.1	239.5
3	Difference	8	-8	20.0%	0	8	-13.3%	5.0	33	1.0	10.9	-78.7
	Centralized	31	9	77.5%	13	18	41.9%	15.0	52	2.8	9.2	163.5
	Decentralized	21	19	52.5%	7	14	33.3%	15.0	12	0.0	2.0	215.3
4	Difference	10	-10	25.0%	6	4	8.6%	0.0	40	2.8	7.2	-51.8
	Centralized	29	11	72.5%	13	16	44.8%	17.0	42	0.4	12.2	260.0
	Decentralized	18	22	45.0%	10	8	55.6%	11.0	7	0.0	3.0	566.3
5	Difference	11	-11	27.5%	3	8	-10.7%	6.0	35	0.4	9.2	-306.2
	Centralized	33	7	82.5%	14	19	42.4%	15.5	42	0.0	10.5	84.6
	Decentralized	29	11	72.5%	15	14	51.7%	18.0	3	0.0	5.4	95.1
6	Difference	4	-4	10.0%	-1	5	-9.3%	-2.5	39	0.0	5.2	-10.5
	Centralized	29	11	72.5%	15	14	51.7%	18.0	40	0.0	4.2	105.9
	Decentralized	27	13	67.5%	19	8	70.4%	13.0	12	0.0	3.0	132.4
7	Difference	2	-2	5.0%	-4	6	-18.6%	5.0	28	0.0	1.2	-26.5
	Centralized	28	12	70.0%	10	18	35.7%	15.0	35	0.0	9.0	221.4
	Decentralized	26	14	65.0%	9	17	34.6%	13.5	7	0.0	10.0	294.4
8	Difference	2	-2	5.0%	1	1	1.1%	1.5	28	0.0	-1.1	-73.0
	Centralized	31	9	77.5%	9	22	29.0%	16.5	50	0.7	4.8	196.5
	Decentralized	23	17	57.5%	13.5	18	42.9%	13.5	18	0.0	4.1	322.3
9	Difference	8	-8	20.0%	-4.5	4	-13.8%	3.0	32	0.7	0.8	-125.8
	Centralized	32	8	80.0%	16	16	50.0%	16.5	31	0.1	10.6	117.5
	Decentralized	27	13	67.5%	19	8	70.4%	10.0	13	0.0	2.5	140.1
10	Difference	5	-5	12.5%	-3	8	-20.4%	6.5	18	0.1	8.1	-22.6
	Centralized	31	9	77.5%	13.8	17.2	44.3%	15.8	41.3	0.5	8.4	151.9
	Decentralized	24.5	15.5	61.3%	13.8	11.6	53.9%	12.9	12.4	0.1	3.9	244.2
Mean	Difference	6.5	-6.5	16.3%	0.05	5.6	-9.7%	2.9	28.9	0.4	4.6	-92.2

Table **6-9**: Results for 2C4 and 2D4 runs.

Replication		Fulfilled	Unfulfilled	% Fulfilled	Internally Fulfilled	Externally Fulfilled	% Internally Fulfilled	Sum of Travel Time (days)	Plan Change Penalty	Total Earliness (days)	Total Lateness (days)	DRAPES Objective
	Centralized	67	13	83.8%	28	39	41.8%	36.5	215	0.3	5.1	178.3
	Decentralized	62	18	77.5%	29	33	46.8%	34.0	68	2.0	8.4	280.3
1	Difference	5	-5	6.3%	-1	6	-5.0%	2.5	147	-1.7	-3.3	-102.0
	Centralized	72	8	90.0%	32	40	44.4%	37.5	222	0.0	13.1	165.9
	Decentralized	70	10	87.5%	40	30	57.1%	35.0	46	0.0	8.4	147.7
2	Difference	2	-2	2.5%	-8	10	-12.7%	2.5	176	0.0	4.7	18.2
	Centralized	69	11	86.3%	26	43	37.7%	44.5	191	1.0	19.3	350.5
	Decentralized	64	16	80.0%	36	28	56.3%	37.0	53	0.0	11.2	394.5
3	Difference	5	-5	6.3%	-10	15	-18.6%	7.5	138	1.0	8.1	-44.0
	Centralized	69.3	10.7	86.7%	28.7	40.7	41.3%	39.5	209	0.4	12.5	231.6
	Decentralized	65.3	14.7	81.7%	35	30.3	53.4%	35.3	55.7	0.7	9.4	274.2
Mean	Difference	4	-4	5.0%	-6.3	10.3	-12.1%	4.2	154	-0.2	3.1	-42.6

Figure **6-5** shows a plot of all differences of paired DRAPES objectives for all four scenarios. 20 replications were made for the initial planning scenario, Scenario 1, and 10 were made for all others. The plot of the differences of the solution quality shows little evidence of significant differences in the relative performance of the decentralized and centralized approaches for the different scenarios. This is demonstrated by comparing 95% confidence intervals, as shown in Figure **6-6**.

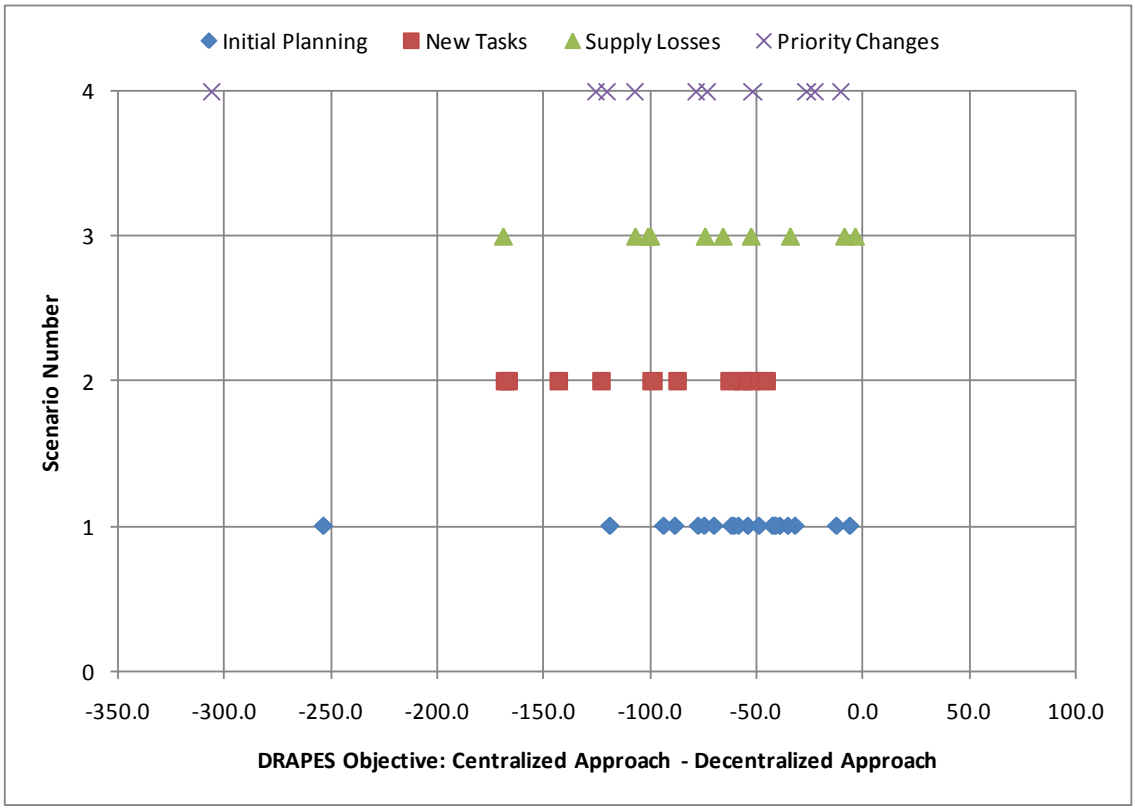

Figure **6-5**: Solution quality for all scenarios of Configuration 1.

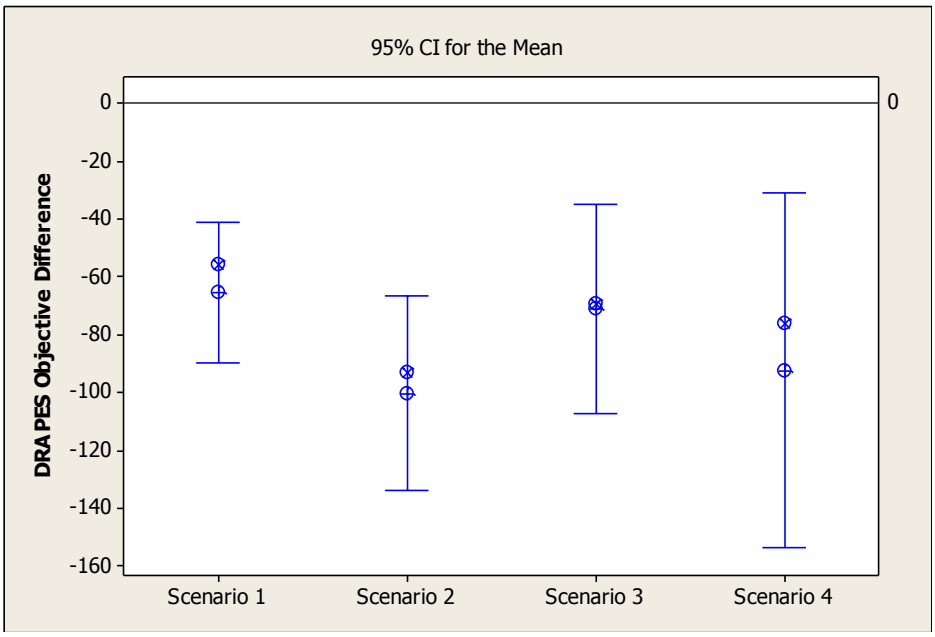

Figure **6-6**: Solution quality confidence intervals for all scenarios of Configuration 1.

Chapter 7

Conclusions and Recommendations

7.1 Simulation

The Distributed Resource Allocation Planning and Execution Simulation (DRAPES) provides both a framework and a decision test bed for simulating distributed logistics operations. It presents a world in which resources are allocated and consumed in a distributed environment according to specified sets of rules and decision logic. The different types of decision logic can be tested under varying conditions in repeatable simulation runs. Modeling the target system as a multi-agent system presented advantages on a domain, design and operational level.

On the domain level, the agent-based approach naturally represents the interactions between autonomous entities that cooperate to fulfill demands. Entities of a real distributed logistics system identifiably sense and receive information, communicate with other agents for specific purposes, and have the ability to take certain actions. They make decisions to achieve a goal based on their limited knowledge of the system, and can react when they sense a change in the state of the system.

On the design level, an agent can then be functionally decomposed into concurrently-executed planning, execution, and simulation activities. Through events triggered by timers and messages, the logic and rules that comprise an agent can be clearly defined and identified. Additionally, the design feature of the simulation agent allows for repeatable experimentation, control over the stresses to which the system is subjected, and reports and statistics at the end of the simulation.

Finally, at an operational level, the distributed nature of the simulation enforces autonomy and real-world constraints on information availability across a network. Also, the use of object-oriented programming makes programming rules and logic more efficient and standardized. For example, when a plan is represented as an object, it becomes easier to use more complicated representations of time, manipulate the plan, and extract information from the plan. Built in Java, DRAPES is also platform independent and allows agents to interface with other software and XML files.

Other future work for the simulation could include better abstraction of business rules for agents in the system. The toughest challenges of developing DRAPES were programming agents' business rules, verifying that they worked, and determining how to fix them if they failed. Visualization of resource allocations and decision processes could have expedited the verification process, supplementing logs, end reports, and debugging. For example, a visualization of an agent's plan changing over time could have helped understand bid values for the decentralized process.

Another important direction for future development of the simulation to make it more valuable to logisticians would be to emulate software components of the logistics systems. For example, an agent representing a unit could be located on a machine with a copy of the real database system used for inventory tracking. Or, if control or consultation is performed through a web service, agents could interact with that web service and make decisions based on its input. Building the agent-based simulation in a more formal and advanced agent platform, such as the Cougaar agent architecture, would make the simulated system more closely resemble a real distributed system and provide a clear pathway to implementation in a logistics system. At the very least, agent

communications could emulate the format of real communications between distributed units—possibly through XML files—for a specific target system. A loftier goal for the future would be to involve logisticians or commanders in the simulation process to get a better sense for how human decisions and interaction with the system affect performance.

7.2 Control Methods

For the scenarios tested in this simulation with simple control methods, the centralized controller based on a mixed-integer program produces significantly better results than a decentralized iterative bidding process. It is able to more efficiently handle trade-offs in allocating supply and transportation resources, resulting in a solution that fulfills a higher percentage of requisitions. For the smaller scenarios, the mathematical program returned a good solution in a reasonable amount of time. The simple rules of the decentralized solution—iterative assignment of requisitions according to priority by bidding—provide a consistently inferior solution but without the computational burden of the mathematical program. However, the more important results are observations about what will be needed in future generations of centralized and decentralized decision rules, and how they can be crossed.

The results of the perturbation scenarios demonstrate the ability of DRAPES to simulate re-planning during execution in response to different types of perturbations. Whether new demands are assigned, supply resources are lost, or mission task priorities change, both the decentralized and centralized approaches can respond and adjust plans accordingly. Re-solving the mixed integer program for every day of the simulation

generally provides a better solution than the iterative planning approach, but the different states of the system at the time perturbations occur can result in rare runs in which the decentralized approach is better than the centralized approach.

The mixed-integer program, as formulated in this thesis, exhibits problems when used to try to solve larger problems, and the solution time could become prohibitive for real-time or quasi real-time planning. One suggested avenue to investigate is to combine requisitions into low, medium, and high-priority demands by commodity by unit by time period. This could eliminate the requisition index set and reduce the need for other variables. The previously cited disaster relief examples in literature take an approach similar to this. Methods from literature for decomposing the problem by commodity could be also be investigated and tested for how well they allocate transportation resources. Furthermore, pre-processing and post-processing of the solution could add to its quality. Perhaps, the planning horizon could be reduced without losing significant quality. Or, decision rules or another mathematical program could help formulate more exact transportation schedules to account for delivery times that do not fit time-period approximations.

Changes in the decentralized decision methods could also help improve its results. It could use a more intelligent and flexible plan that could evaluate trade-offs that would, for example, allow it to slightly shift the delivery date of a high-priority requisition so that a medium-priority requisition would not go unfulfilled. Alternatively, it could solve a mathematical program based on local knowledge to calculate its bid and eliminate some of the inefficiencies of the iterative priority approach. Also, if communication between agents is limited across a network, the decentralized system would need to be redesigned

to account for limitations on the number of requisitions that could go through the bidding process within a short time period.

Both styles of control—centralized and decentralized—need to be further developed and investigated for how they enable re-planning in response to perturbations, such as new task arrivals, dynamic requisition priorities, resource unavailability, and loss of communication. It is suggested that level of centralized or decentralized control could adapt the current state of a network. Units operating in distributed environment with undependable network connections would make more decisions on their own about how to source requisitions, rather than wait for orders from superior units in the unit hierarchy. Also, when perturbations to a plan occur, units could implement their own short-term remedies, notify a central controller, and wait for order from the central controller for how to change a long-term plan.

Bibliography

Adelman, D. 2007. Price-directed control of a closed logistics queueing network. *Operations Research* **55** 1022-1038.

Aknine, S., S. Pinson, M. Shakun. 2004. An extended multi-agent negotiation protocol. *Autonomous Agents and Multi-Agent Systems* **8** 5-45.

Balcik, B., B. Beamon, K. Smilowitz. 2008. Last mile distribution in humanitarian relief. *Journal of Intelligent Transportation Systems* **12** 51-63.

Barbarosoglu, G., Y. Arda. 2004. A two-stage stochastic programming framework for transportation planning in disaster response. *Journal of the Operations Research Society* **55** 43-53.

Davidson, P., L. Henesey, L. Ramstedt, J. Tornquist, F. Wernstedt. 2005. An analysis of agent-based approaches to transport logistics. *Transportation Research Part C* **13** 255-271.

Drogoul, A., D. Vanbergue, T. Meurisse. 2003. Multi-agent based simulation: where are the agents? *Lecture Notes in Computer Science* **2581** 43-49.

Fischer, K., J. Muller, M. Pischel. 1996. Cooperative transportation scheduling: An application domain for DAI. *Applied Artificial Intelligence* **10** 1-34.

Gnanasambandam, N., S. Lee, N. Guatam, S. R. T. Kumara, W. Peng, V. Manikonda, M. Brinn, M. Greaves. 2004. Reliable MAS performance prediction using queueing models. *IEEE First Symposium on Multi-Agent Security and Survivability* 55-64.

Graves, S. 1996. A multiechelon inventory model with fixed replenishment intervals. *Management Science* **42** 1-17.

Haghani, A., S. Oh. 1996. Formulation and solution of a multi-commodity, multi-modal network flow model for disaster relief operations. *Transportation Research Part A* **30** 231-250

Jennings, N. 2000. On agent-based software engineering. *Artificial Intelligence* **117** 277-296.

Lee, S., S. Kumara, N. Gautam. 2008. Market-based model predictive control for large-scale information networks: completion time and value of solution. *IEEE Transactions on Automation Science and Engineering* **5** 630-640.

Lichter, S. 2009. Quasi-static inventory assignment: a centralized and decentralized approach. Working paper.

OpenCybele™ Agent Infrastructure Users Guide. 2002. Distributed Intelligent Systems, Intelligent Automation Inc. http://www.OpenCybele.org.

Ozdamar, L., E. Ekinci, B. Kucukyazici. 2004. Emergency logistics planning in natural disasters. *Annals of Operations Research* **129** 217-245.

Perugini, D. 2006. Agents for logistics: A provisional agreement approach. Ph.D. thesis. University of Melbourne, Australia.

Powell, W., T. Carvalho. 1998. Dynamic control of logistics queueing networks for large-scale fleet management. *Transportation Research* **32** 90-109.

Shen, W., Q. Hao., H. Yoon, D. Norrie. 2006. Applications of agent-based systems in intelligent manufacturing: An updated review. *Advanced Engineering Information* **20** 415-431.