Keywords: Warehouse, Supply Chain Management, Methodology

Abstract
Warehousing is one of the most important elements of a supply chain. It is common to use analytical and simulation-based operations research techniques in the studies held on this topic. Warehousing systems are complex in their nature and analytical techniques lack modeling high level of system detail. On the other hand, simulation is more convenient to model complex warehouse systems with high level of detail. In this paper we present a new approach for modeling and simulation of unit-load warehouse systems. Our approach includes the use of a discrete event simulation (DES) library and a warehouse class library. We also showed a real application by utilizing our approach.

1. INTRODUCTION
In today’s highly competitive markets, the use of Operations Research (OR) methods which makes improvements in Supply Chain Systems (SCSs) has much importance than it has in the past. Improvements in any stage of SCSs have individual and whole system effects. In this context, warehouses are accepted as one of the key elements which have crucial impacts on SCSs. Plenty of researches in the literature accordingly have sought ways to improve warehousing systems.

We present a new methodology for creating simulation models of unit-load warehouse systems. By using our methodology, the user can build models for unit-load warehouses operating with single putaway/picking policies to study some warehousing problems related with resource allocation, layout design and storage policies.

We developed our methodology in three phases. First, we developed a general purpose simulation library; SharpSim (Ceylan and Gunal, 2011) which is a Discrete Event Simulation (DES) software library written in CSharp (C#) programming language. SharpSim implements Event Scheduling (ES) world-view and uses Event Graphs (EG) as the modeling tool. Although SharpSim is developed within our methodology, it is a generic tool which can be used for modeling any system.

Secondly, we developed a warehouse library (WareLib) which maintains the basic warehouse objects and data structures required to simulate a warehouse. This library provides us a generic virtual warehouse including 3D visualization and animation features.

Lastly, we have developed a general conceptual model for a unit-load warehouse operating with single putaway/picking policy, using EGs. This conceptual model has been implemented by using of SharpSim and WareLib. The conceptual model is a template who intends to develop conceptual warehouse simulation models adopting ES approach. Modelers can simply generate adaptive conceptual models using this template. The model includes (1) receiving, (2) putaway, (3) crossdocking, (4) picking and (5) shipping functions.

After a review of the related literature, we give some technical details of the WareLib and the conceptual model. The readers can find more about SharpSim, which is the first phase of our methodology, at Ceylan and Gunal, 2011. A section for the implementation is also included before the conclusion.

2. LITERATURE REVIEW
We examine some of the studies held on warehousing systems in this limited literature review. First, we present a general picture of warehousing problems by examining two literature reviews; then, evaluate some warehouse models.

One of the reviews was done by [2]. They classified operational problems based on main warehouse functions; receiving, shipping, storage and order picking. Regarding to receiving and shipping; there are key parameters such as incoming shipments’ arrival time, demand, orders, dock layout, material handling resources, and problems are related to layout, relative location of docks, management policies and throughput requirements etc. On the storage side; fundamental questions such as, what should be the inventory level for an SKU and where should an SKU be kept in the warehouse, come forward. In terms of storage location assignment problem, there are three main storage policies, namely random policy, dedicated policy and class-based policy. This review revealed that with dedicated storage policy, material handling systems can be used in a more effective way, since fast-moving items can be placed close to docks. On the other hand, this policy suffers from
higher storage space requirement, because, for each product a number of storage locations which equals to its maximum inventory level should be reserved. In class-based location assignment, there are three frequently used criteria: popularity, max inventory and cube-per-order index (COI). In popularity criterion, items or product classes are ordered according with their number of retrievals per unit time. Items having highest popularity are assigned to the most preferential locations. In maximum inventory criterion, items or product classes are ordered according with maximum storage space requirements. Items having lowest space requirements are assigned to the most preferential locations. COI criterion combines former two. It is the ratio of maximum inventory to popularity. Items having lowest COI values are assigned to the most preferential locations.

This review, lastly, categorizes order picking into batching, sequencing/routing and sorting. Batching is a planning problem and the decision variable for this problem is to partition of orders for assignment to pickers. In sequencing/routing, the aim is to determine the best sequence and route of locations. Sorting is required for the warehouses having accumulation conveyor. It is used to separate picked items according with shipping needs.

Another type of problems in warehouse systems is the design problems. These are categorized into overall structure, sizing and dimensioning, department layout, and equipment selection and operation strategy [3]. The determination of the quantity of departments and flows among departments is in overall structure problems. Warehouse sizing and dimensioning problems deal with the storage capacity and operating costs in terms of floor space respectively. Department layout problems consider door location, aisle orientation, length and width of aisles, and number of aisles. The determination of storage and material handling systems considering performance aspects is dealt with in equipment selection category. Lastly, operation strategies are related with the decisions about storage and order picking operations.

Although there are many models developed for analyzing warehouse systems in the literature, we give two examples here. A model to obtain best trailer-door assignment to minimize total cost which is a function of travel time between doors and waiting time up to congestion is developed by [4]. In this study, they found that, in brief, activity tends to concentrate in the center. Real case implementation of the model provided a 11% improvement in productivity. An extension of this study is [5] which reports that the most common crossdock warehouse shapes are I, L and T. I-shaped is preferred in general for smaller crossdock warehouses. The door with smallest average distance is indicated as the best door and center doors of an I-shaped warehouse are given as the most convenient doors in the study. Two important definitions are diameter and centrality. The diameter refers to the distance between doors and the centrality is the number of doors required to increase the diameter by one door offset. It is indicated that I-shape lose more centrality as the number of doors increase comparing to other designs. On the other hand, those designs having some corners incur some other costs because of labor efficiency up to safety regulations and congestion. As the result of computational experiments, it is reported that alternative shapes other than I-shape can be preferable when the number of doors exceeds 150.

Another model is developed by [6] to analyze how the number of cross aisles affects picking travel time in a rectangular warehouse operating with multi-picking tours. For this reason, two types of cross aisles, literally equally spaced and unequally spaced, have been considered. The computational experiments resulted that equally spaced cross aisles aid to reduce picking travel time more, comparing to unequally spaced ones. They provide a table for the number of equally spaced cross aisles that should be applied when length of the warehouse and the number of main aisles are given. Furthermore, they indicated that the number of cross aisle should be set to three when length of warehouse and number of main aisles are not determined.

As a result of this limited review, we concluded that, although, detailed warehouse simulation models are very useful in measuring the effects of planned strategic and tactical changes on the system, the number of simulations which examine warehouses in more detail is limited in the literature. The models in the literature are generally developed for specific purposes to study specific problems in the domain.

3. WAREHOUSE LIBRARY (WARELIB)

WareLib is a class library which maintains the basic warehouse objects and data structures required to simulate a warehouse. WareLib hosts a Warehouse object along with Resource and Entity objects. There are two types of Resource objects; locations and material handling systems. These resources interact with Entity objects, such as a pallet and a truck on warehouse platform.

For modeling purposes, a warehouse can be thought as a grid which is composed of Cell objects. Each cell can be used for storage or passage purposes. This approach eases life since a spreadsheet can represent a warehouse layout. A spreadsheet is divided into cells having the size of a pallet for storage cells and appropriate size for passage cells.

Figure 1 shows an object hierarchy for grid based warehouse. Cell class is the base class and others down in the hierarchy inherit from Cell class.

A detailed explanation of these classes can be found at [7] which can be accessed upon request from the authors. Here, we only give some detail on how these classes are used. For example, CriticalCrossAisleCell objects are linked to each other to create a graph of the warehouse. Since, relations among each StorageLocationCell object and
CriticalCrossAisleCell objects are already defined in the warehouse data structure; the graph is used in the computation of all the shortest routes among any storage locations with Floyd-Warshall algorithm. We assume that the layout of a warehouse would not change during the simulation execution, the shortest route calculations can be done at the start of the simulation. The design can be done on an Excel sheet.

Figure 1. Grid Structure Hierarchy.

In addition to Cell objects, there are four main entity type in the warehouse library. These are truck, pallet, demand and order classes which inherit from abstract Entity Class.

4. CONCEPTUAL MODEL OF A WAREHOUSE

Warehouses vary in design and procedures. However, some patterns and functions are shared by most warehouses. Starting from this point, we developed a general warehouse conceptual model. We specified common functions of a warehouse as receiving, putaway, picking, shipping, and crossdocking.

We used Event Graphs (EGs) to create our conceptual model. An EG has two elements; events and edges. Events occur in time which transforms entities and leads to state changes in the system. Edges link events and represent transitions between events. EGs are useful for representing DES models which adopts event scheduling world-view. There are two major advantages of EGs. First, they are easy since a modeler only focuses on the events in the system. Second, they are flexible since any component can be removed from the model easily without requiring any other modifications. EGs also support component based modeling.

Our conceptual model is designed for unit-load warehouses operating with single-picking policy. The model tracks each pallet and it offers a high level of detail without suffering from prolong runtime.

The model separates warehouse operations into inbound and outbound processes as encountered in almost all warehouses. Each process decomposes into some sub-processes and components. According to this, inbound process involves truck receiving, pallet receiving and putaway sub-processes; whereas outbound process comprises demand receiving component, demand satisfaction and shipping sub-processes. The total decomposition of warehouse operations used in the model is presented in Figure 2.

Figure 2. Decomposition of Warehouse Operations.

Although we built an EG for each component in Figure 2, we only explained “Demand Receiving” component in this paper. This component is shown in Figure 3 and includes demand arrival, truck to shipping dock and entity transition events. In the model, double demand arrival events are utilized to demonstrate that multi-arrival can be defined in the model to manipulate different type of demand arrival patterns, e.g. two types of products.

Figure 3. Demand Receiving Component of the Conceptual Model.

Demand_Arrival is represented by the node number 18a and 18b in the EG model. Demand_Arrival event represents...
the arrival of a demand. It triggers a Truck_to_Shipping_Dock event and itself with a self loop. This self loop mechanism is useful to schedule events recursively in DES. There is a condition up to dock availability between arrival event and docking event. Truck entities associated waiting for dock are kept in dock queue. The first demand entity in the dock queue is passed to Truck_to_Shipping_Dock event. A new demand entity is created in this event and passed to loop for triggering a new arrival. List of order entities which constitute demand entities are created together with demand entities. This loop has a delay which represents inter arrival time. Demand_Arrival events can be multiplied according to arrival needs. Demand_Arrival event also triggers Demand_to_Order_Transition event up to dock availability condition. Besides any demand arrival events can be thought as sub-components. They can be removed from the model or new demand arrival events can be added to the model easily. Truck_to_Shipping_Dock is represented by the node number 19 in the EG model. Each demand is shipped with a separate shipping truck.

Demand_to_Order_Transition is represented by the node number 20 in the EG model. Demands are a set of orders and each order in a demand order list should be treated one by one. Multi-passing between Demand_Arrival event and Demand_to_Order_Transition event provide the triggering of a number of orders in return of an arriving demand.

Demand satisfaction sub-process is grouped into crossdocking, picking and no picking components. Here, the picking type is determined and if possible it is satisfied. Offloading area is primarily checked for an available item to satisfy demand by crossdocking. If crossdocking is not possible, then the storage area is checked for picking. In case, order cannot be satisfied by both functions, no picking occurs. For this model, selecting a picking type and no picking are assumed to be a mid-component under demand satisfaction.

Pin numbers 1, 33, and 21 in Figure 3 are the connectors for this component. For example, number 1 connects “Run” event and number 33 connects “Release Truck” event. In other words, when the Run event is executed, the DES algorithm schedules, or say inserts, two “Demand Arrival” events. Likewise, when “Entity Transition” event occurs “Select Picking” event is scheduled to the Future Event List in the DES model.

5. IMPLEMENTATION
To use our approach in action, we implemented the conceptual model as a DES model. A C# project which uses SharpSim and WareLib libraries were created. SharpSim provided the required simulation environment for the implementation of the conceptual model, while WareLib was used to simulate a generic warehouse. We also created a user-friendly interface for easy handling of the analysis tool.

Main properties of the model are a simulation object provided by SharpSim and a warehouse object provided by WareLib. Mainly these two objects interact with each other on a windows form. Simulation model is constituted by instantiating Simulation, Event and Edge objects. Warehouse environment is defined by Excel inputs. As explained earlier, a warehouse can be thought as a grid composed of cells such as storage location, main aisle, cross aisle, critical cells, offloading area location, shipping area location, inbound dock and outbound dock. The user designs the input Excel file by specifying the cell types on the spreadsheet.

There are some other spreadsheets required for dedicating locations for item types, setting traffic direction along the aisles and setting initial configuration of pallets inside the warehouse. Truck load information is set through implementation form.

The interface of the project has been designed to handle the model easily. The interface is mainly composed of three forms. Main form (GWS: Generic Warehouse Simulation), warehouse and simulation forms. The interface is shown in Figure 4.

Figure 4. User interface of the implementation.

Animation is one of the methods used for validating simulation models. Within our approach, we also added animation features to WareLib. We used XNA Game Studio [8] to animate entities and visualize the objects. In WareLib, each physical entity is represented by a 3D graphics object. The animation is done by the “Ping” events in which graphic objects are updated on the screen in regular time steps. When inter event time between any two Ping events decreases, a smoother move of Pallet objects is obtained. However, this causes the simulation to run slower since the
number of Ping events to be executed increases in Future Event List (FEL).

6. CONCLUSION

We presented a new approach for simulating warehouses. We developed our approach in three phases. In the first phase we developed a general purpose DES library (SharpSim) to gain total control of the simulation. We chose Event Scheduling (ES) world-view for this simulation library. In the next phase, we developed a warehouse library (WareLib) to maintain the basic warehouse objects and data structures required to simulate a warehouse. By using Excel input, WareLib provides flexibility in designing warehouses. Finally, we developed a general conceptual model for a unit-load warehouse operating with single putaway/picking policy. The conceptual model includes main warehouse functions, literally receiving, putaway, crossdocking, picking and shipping. By the implementation of the conceptual model with SharpSim and WareLib, we reached our aim in developing an analysis tool. In addition to this, we believe that the conceptual model can be used as a template by the modelers who intend to develop conceptual warehouse simulation models adopting ES approach.

As a show-case for our approach, we developed a general analysis tool for unit-load warehouses operating with single putaway/picking policies to study some warehousing problems such as resource allocation, layout design and storage policies. Using this tool we conducted some experiments to evaluate the effects of dock positioning on forklift utilization. We gained some insights based on relative comparisons among warehouse layouts and relevant dock configurations, however our findings haven’t been included in this paper.

References


Acknowledgments and Disclaimer

The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of any affiliated organization or government. Arda Ceylan spent five months of his masters study at DIPTEM, University of Genoa, under supervision of Prof. Agostino Bruzzone.

Biography

Arda Ceylan received his MSc Naval Operations Research degree at the Institute of Naval Science and Engineering in Turkish Naval Academy. He is interested in simulation methodologies.

Murat Gunal is currently working as an assistant professor at the Industrial Engineering Department of the Turkish Naval Academy. He completed his MSc in Operational Research and PhD in Management Science at Lancaster University in U.K. in 2000 and 2008 respectively. His main research interest is simulation modelling for better decision making. He developed many simulation models in many domains including healthcare, military, and logistics. He is also interested in simulation methodologies and their use for optimisation. He is in the editorial board of the Journal of Simulation and is an area editor of newly arriving Health Systems.

Agostino Bruzzone is currently Full Professor at DIPTEM, University of Genoa. He has utilized extensively simulation techniques in harbor terminals, maritime trading and sailboat racing sectors. He worked on research projects involving innovative modeling, AI techniques and DOE (design of Experiments); particular attention was focused on the application of Neural Networks and Fuzzy Logic to industrial plant problems using Simulation and Chaos Theory. He has been actively involved in the scientific
community from several years and served as Director of the McLeod Institute of Simulation Science (MISS), Associate Vice-President and Member of the Board of the SCS (Society for Modelling & Simulation international), President of the Liophant Simulation, VicePresident of MIMOS (Movimento Italiano di Simulazione) and Italian Point of Contact for the ISAG (International Simulation Advisory Group) and Sim-Serv. Currently he is Director of the M&S NET.