Managing Order Batching Issue of Supply Chain Management with Multi-Agent System

[1,]Manoj Kumar , [2,]Dr. S Srinivasan , [3,]Dr. Gundeep Tanwar

[1]*Research Scholar, Suresh GyanVihar University Jaipur, Rajasthan (India),*

[2]*Professor & Head, PDM College of Engineering, Bahadurgarh, Haryana (India),*

[3]*Associate Professor, BRCM College of Engineering & Technology, Bahal, Haryana (India),*

Abstract

Recently, agent-based technology has been taken as a promising approach for developing advanced manufacturing systems. Such an approach provides rapid responsive and dynamic reconfigurable structures to facilitate flexible and efficient use of manufacturing resources in a rapidly changing environment. A multi-agent system (MAS) is a loosely coupled network of software agents that interact to solve problems that are beyond the individual capacities or knowledge of each problem solver. In this paper we will apply the concept of Multi-agent system for handing the order batching issue faced in the supply chain management. The order batching is major cause of generating the bullwhip effect. The bullwhip effect produces the worst impact on the performance of the supply chain system. With this aim, we have used the JADE for developing the supply chain management system in which the intelligent agents maintain the information related the order batching issues and the decisions regarding managing the order batching.

Keywords: *Supply chain management, Bullwhip effect, Multi-agent system, Ordering Batching, Coordination, JADE.*

I. INTRODUCTION

For most application tasks, it is extremely difficult or even impossible to correctly determinate the behavioral repertoire and concrete activities of a multi-agent system a priori, that is, at the time of its design and prior to its use. This would require, for instance, that it is known a priori which environmental requirements will emerge in the future, which agents will be available at the time of emergence, and how the available agents will have to interact in response to these requirements. Such problems resulting from the complexity of multi-agent systems can be avoided or at least reduced by endowing the agents with the ability to learn, that is, with the ability to improve the future performance of the total system, or a part of the system.

Over the past decade, there has been an increased industry interest in the application of multi-agent systems. With the advent of distributed software applications, traditional approaches to programming the interaction between sub-systems has proven to be time consuming and error prone. Typically, distributed systems are embedded in, and must interact with, a changing environment. The interaction with the environment is particularly problematic where there are many external entities that must be controlled, serviced or modeled. In the past, implementing such systems has entailed explicit programming of the interaction with each external entity. If this is more than a simple client/server interaction, then the application can become hard to manage, that is, difficult to implement, maintain or change. Furthermore, the application may not have the required flexibility or responsiveness to its environment.

Agent-based approaches have proven to be well suited where complex interaction with an ever-changing environment is required. Arguably, the most significant attribute of agent-based systems is that each agent is an autonomous computational entity. Autonomy, coupled with an ability to perceive the environment, act upon it and communicate with other agents, provides system builders with a very powerful form of encapsulation. A given agent can be defined in terms of its goals, knowledge and social capability, and then left to perform its function autonomously within the environment it was designed to function in. This is a very effective way of building distributed systems—each agent in the system is responsible for pursuing its own goals, reacting to events and communicating with other agents in the system. There is no need to explicitly program the interactions of the whole system; rather, the interactions *emerge* as a by-product of the individual goals and capabilities of the constituent agents. In the next section, we will further expand our discussion of the motivation for adopting agent-oriented design and programming in industry.

II. MULTI-AGENT SYSTEM

Multi-agent systems (MAS) consist of intelligent agents that are autonomous, dynamic, cooperative or self-interested by nature. Each agent in a multi-agent system is characterized by a degree of autonomy. On basis of its autonomy feature, the agents have ability to decide on their own behaviour. Given the autonomous nature of each agent, agents need to coordinate among themselves, or else, the group quickly changes to a number of individuals with chaotic behaviour. The intelligent agent can update on its knowledge base regarding the environment. They are dynamic by the nature. Cooperative agents coordinate among themselves towards achieving a common goal, where self-interested agents have individual interacting goals. An effective way to achieve coordination is via imposing a specific group organization. An organization comprises roles and their interrelations.

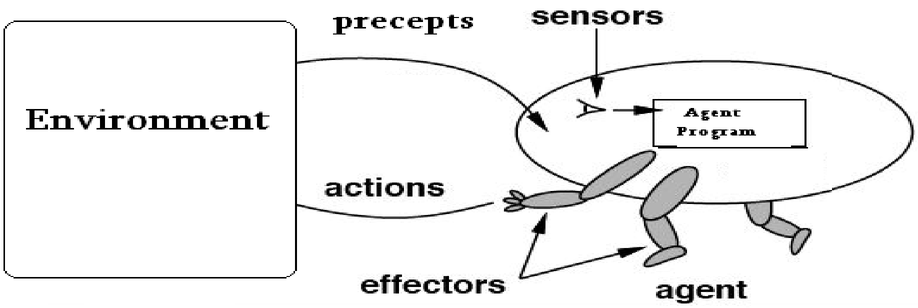

Figure 1: Intelligent Agent

The basic features of MAS are that (1) each agent has partial information or capabilities for solving the problem; (2) the data are decentralized; and (3) working out is not synchronous (4) there does not exit any system global control. The multiagent systems have various capabilities & abilities to implement various types of real time applications. The Multi-agent systems have aptitude to solve problems that are too large for a centralized agent to solve because of resource limitations or the sheer risk of having one centralized system that could be a performance bottleneck or could fail at critical times. The Multi-agent systems can resolve the problems efficiently in which information sources that are spatially distributed. They can also provide the solutions in situations where capability is distributed. Examples of such problems include concurrent engineering. The Multi-agent systems have potential to allow for the interconnection and interoperation of multiple existing inheritance systems. The MAS can provide solutions to problems that can naturally be regarded as a society of autonomous interacting components agents.

There are some major concerns or issues that have to resolve in implementing the real time applications. For example task allocation, communication & decision-making are some of the main issues. Task allocation is the problem of assigning duties and problem-solving resources to an agent. KQML is a protocol for communications among both agents and application programs. Agents need to deliberate effectively by evaluating their options and opportunities towards achieving their shared goals without considering low-level details of their plans. To plan and act robustly in a dynamically and unpredictably changing environment, agents must reason about their intended behavior in an abstract way. An MAS has the following advantages over a single agent or centralized approach:

- The MAS distributes computational resources and capabilities across a network of interconnected agents. Whereas a centralized system may be plagued by resource limitations, performance bottlenecks, or critical failures, an MAS is decentralized and thus does not suffer from the "single point of failure" problem associated with centralized systems.
- The MAS allows for the interconnection and interoperation of multiple existing legacy systems. By building an agent wrapper around such systems, they can be incorporated into an agent society.
- The MAS models problems in terms of autonomous interacting component-agents, which is proving to be a more natural way of representing task allocation, team planning, user preferences, open environments, and so on.
- The MAS efficiently retrieves, filters, and globally coordinates information from sources that are spatially distributed.
- The MAS provides solutions in situations where expertise is spatially and temporally distributed.
- The MAS enhances overall system performance, specifically along the dimensions of computational efficiency, reliability, extensibility, robustness, maintainability, responsiveness, flexibility, and reuse.

III. RELATED WORK

Hsu et al. 2005 developed an order batching approach based on genetic algorithms (GAs) to deal with order batching problems with any kind of batch structure and any kind of warehouse layout. Unlike to previous batching methods, the proposed approach, additionally, did not require the computation of order/batch proximity and the estimation of travel distance. The proposed GAbased order batching method, namely GABM, directly minimized the total travel distance. The potential of applying GABM for solving medium- and large-scale order batching problems was also investigated by using several examples. From the batching results, the proposed GABM approach appeared to obtain quality solutions in terms of travel distance and facility utilization.

Potter et al. 2006 derived a closed form expression for bullwhip when demand is deterministic. This was validated through a simple model of a production control system. An expression for bullwhip in a ‘‘pass on orders’’ scenario with stochastic demand was also derived and validated. Using simulation, they showed the impact of changing batch size on bullwhip in a production control system. Their results showed that a manager may achieve economies through batching while minimizing the impact on bullwhip through the careful selection of the batch size.Zunino et al. 2009 described Chronos, a multi-agent system for helping users in organizing their meetings. The system assigned an intelligent Organizer Agent to each user. These agents were able to schedule events negotiating time, place, day, etc. according to users' habits and preferences. Chronos agents did not reveal users' habits or calendars to other users in order to maintain privacy. In addition, each Organizer Agent learnt from its user's preferences and from negotiations with other agents, to provide better and more accurate assistance. The paper reported encouraging experimental results and comparisons with related approaches.Herrero et al. 2009 focused the development of network-based IDSs from an architectural point of view, in which multiagent systems were applied for the development of IDSs, presenting an up-to-date revision of the state of the art. Intrusion Detection Systems (IDSs) are seen as an important component in comprehensive security solutions. Thus, IDSs are common elements in modern infrastructures to enforce network policies. So far, plenty of techniques have been applied for the detection of intrusions, which has been reported in many surveys.

Henn et al. 2010 dealed with the question of how a given set of customer orders should be combined such that the total length of all tours is minimized which are necessary to collect all items. The authors introduced two metaheuristic approaches for the solution of this problem: the first one is based on Iterated Local Search; the second on Ant Colony Optimization. In a series of extensive numerical experiments, the newly developed approaches are benchmarked against classic solution methods. It demonstrated that the proposed methods are not only superior to existing methods but provide solutions which may allow distribution warehouses to be operated significantly more efficiently. Kocsis et al. 2010 gave firstly a classification scheme of scheduling problems and their solving methods. The main aspects under examination were the following: machine and secondary resources, constraints, objective functions, uncertainty, mathematical models and adapted solution methods. In a second part, based on this scheme, they examined a corpus of 60 main articles in scheduling literature from 1977 to 2009. The main purpose was to discover the underlying themes within the literature and to examine how they had evolved. To identify documents likely to be closely related, they were going to use the cocitation-based method. Their aim is to build a base of articles in order to extract the much developed research themes and find the less examined ones as well, and then try to discuss the reasons of the poorly investigation of some areas.

IV. ODER BATCHING PROBLEM IN SUPPLY CHAIN MANAGEMENT

Order batching is one of the methods used in warehouses to minimize the travel distance of pickers. In this paper, we focus on developing order-batching methods for an order-picking warehouse with two cross aisles and an I/O point at one of its corners. Each of these methods is made up of one seed-order selection rule and one accompanying-order selection rule. Eleven seed-order selection rules and 14 accompanying-order selection rules are studied here. These rules include those newly proposed by us and those by others. Rules proposed by others have been shown to perform well in minimizing the travel distance of pickers. They are included here for the comparison purpose. Unlike previous studies that only focus on developing aisle or location-based rules, this study also develops rules that are distance- or area-based.

Order picking, or order selection, is the process of retrieving individual items (from storage locations) for the purpose of fulfilling an order for a customer. Schemes by which to achieve efficient order picking will vary widely. However, in all cases it involves locating the items in storage; creating a plan for retrieving the items; physically picking the items (either automatically or manually); sorting and/or assembling them into discrete orders; and in the end even packaging the orders for delivery.

Although defined as a process, order picking cannot be achieved without the appropriate computer software and mechanical equipment, including the storage medium, such as pallet racks, shelving, AS/RS(including carousels) and flow delivery racks, and a means for transporting items from receiving to storage and from storage to packaging and shipment. A variety of industrial trucks and conveyors will be found in most order picking applications. Order picking may also involve robotic like devices for physically picking discrete items from their storage location When performing his/her tasks, an order picker is guided by a so-called pick list.

This list specifies the sequence according to which the requested articles should be collected, as well as the quantities in which they are to be picked. A pick list may contain the articles of a single customer order (pick-by- order) or of a combination of customer orders (pick-by-batch). In practice, the sequence in which the articles are to be picked and the corresponding route of the order picker (which starts at the de- pot, proceeds to the respective storage locations, and returns to the depot) is usually determined by means of a so-called routing strategy, e.g., by the S-Shape Heuristic or by the Largest-Gap Heuristic. Despite the fact that an optimal polynomial time algorithm for the picker routing problem exists, it is hardly ever used in practice. Order pickers seem not to accept the optimal routes provided by the algorithm, because of their not-always-straightforward or sometimes even confusing routing schemes. The heuristic routing schemes on the other hand, are fast to memorize and quite easy to follow. This helps to reduce the risk of missing an article to be picked -- an aspect that might be more important than a small reduction of the tour length.

Batching determines which orders are released together. With batch picking, multiple orders are picked together in one pick tour and need to be sorted by order later. By sharing a pick tour, the average travel time per order is reduced. Basically, two criteria for batching exist: proximity of pick locations batching and time-window batching. Proximity batching refers to the clustering of a given number of orders based on retrieval locations. Time-window batching studies the order batching problem in a stochastic context. The number of orders per batch can be fixed or variable. Variable time-window batching groups all orders that arrive during the same time interval or window. With fixed number of orders time-window batching, the time window is the variable length until a batch has a predetermined number of orders. Zoning is closely related to batching; it divides the pick area into sub-divisions (or zones), each with one or few pickers dedicated to it.

The major advantages of zoning are: reduction of the travel time (because of the smaller traversed area and also the familiarity of the picker with the zone) and of the traffic congestion. Depending on the pick process sequence, zoning can be further classified as progressive zoning or synchronized zoning. With progressive zoning, orders are sequentially picked zone by zone (this system is also called ‘pick-and-pass’); a batch is finished when all (order) lines of the orders in the batch are picked. In contrast, in synchronized (or parallel) zoning, pickers in all zones can work on the same batch at the same time. In synchronized zoning, the picking process must be followed by a sorting (and often also a packing) process, to group the items of the same order picked by the multiple pickers. Zoning has received little attention in the literature despite its important impact on the performance of the order picking system. Choe et al. (1993) study the effects of three order picking strategies in an aisle-based system: single order pick, sort-while-pick, and pick-and-sort. They propose analytical tools for the planner to quickly evaluate various alternatives without using simulation.

V. INFORMATION MAINTAINED BY MAS

In this proposed work , the system concentrate on the order processing that is a key element of Order fulfillment. Order processing operations or facilities are commonly called "distribution centers". Order processing is the term generally used to describe the process or the work flow associated with the picking, packing and delivery of the packed item(s) to a shipping carrier. The specific "order fulfillment process" or the operational procedures of distribution centers are determined by many factors. Each distribution center has its own unique requirements or priorities. There is no one size fits all process that universally provides the most efficient operation. Some of the factors that determine the specific process flow of a distribution center are:

- The nature of the shipped product - shipping eggs and shipping shirts can require differing fulfillment processes
- The nature of the orders - the number of differing items and quantities of each item in orders
- The nature of the shipping packaging - cases, totes, envelopes, pallets can create process variations
- Shipping costs - consolidation of orders, shipping pre-sort can change processing operations
- Availability and cost and productivity of workforce - can create trade-off decisions in automation and manual processing operations

- Timeliness of shipment windows - when shipments need to be completed based on carriers can create processing variations
- Availability of capital expenditure dollars - influence on manual verses automated process decisions and longer term benefits
- Value of product shipped - the ratio of the value of the shipped product and the order fulfillment cost
- Seasonality variations in outbound volume - amount and duration of seasonal peaks and valleys of outbound volume
- Predictability of future volume, product and order profiles -
- Predictability of distribution network - whether or not the network itself is going to change

This list is only a small sample of factors that influence the choice of a distribution centers operational procedures are being managed by multi-agent system. Because each factor has varying importance in each organization the net effect is that each organization has unique processing requirements.

VI. MULTI-AGENT SYSTEM ARCHITECTURE FOR HANDING ORDER BATCHING ISSUE

In this system, there are multiple intelligent agents having the information about the order scheduling. These intelligent agents communicate with each other to shared the information regarding scheduling. Lets us discuss how multi-agent system can be applied to solve timetable-scheduling problem very efficiently. In supply chain management system at big organization, whole operational responsibilities are divided on various coordinators. Each coordinator has responsibility of own departments. Each coordinator has maintained own knowledge base. It also follows the soft & hard constraints. They share the information regarding the product scheduling. Consider there are multiple departments in the organization. In proposed multi-agent system, the various intelligent agents are communicating with each other.

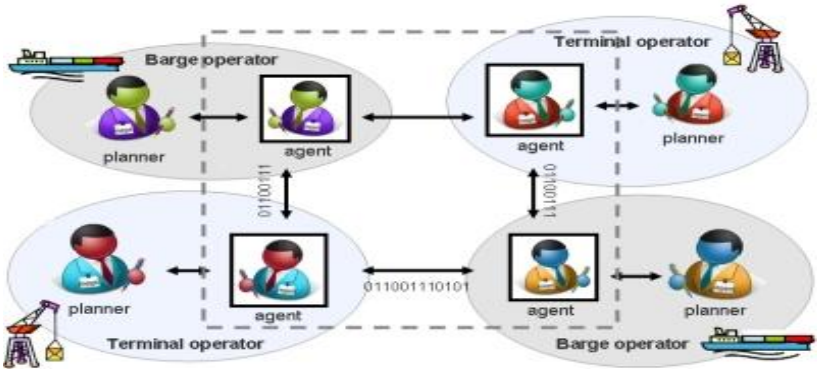

Figure 2: Multi-agent Based SCM

VII. IMPLEMENTATION

For this work we use the JADE soft for designed the multi-agent based supply chain for resolving the order batching issues. The Java Agent Development Environment is a software framework fully implemented in Java language. It simplifies the implementation of multi-agent systems through a middleware that complies with the Foundation for Intelligent Physical Agents specifications and through a set of graphical tools that supports the debugging and deployment phases. The agent platform can be distributed across machines that not even need to share the same OS and the configuration can be controlled via a remote graphical user interface (GUI). The configuration can be even changed at run-time by moving agents from one machine to another one, as and when required. First we create the JADE agent is as simple as defining a class extending the jade.core.Agent class and implementing the setup() method as shown in the code below.

```
import jade.core.Agent;
public class SCMAgent extends Agent {
protected void setup() {
// The logic of the agent will be written here
```

The setup() method is intended to include agent initializations. The actual job an agent has to do is typically carried out within “behaviours”. JADE is relatively suitable for simple agent applications and developers, that require FIPA compliance. JADE’s debugging and monitoring tools and its support for FIPA ACL message format are a sound foundation for systems interactions that require cross-platform interoperability. JADE can be distributed over several hosts, resulting in a distributed system that seems like a single platform from the outside. For running the jade, the following command is being executed *java jade.Boot –gui* then we get following screen

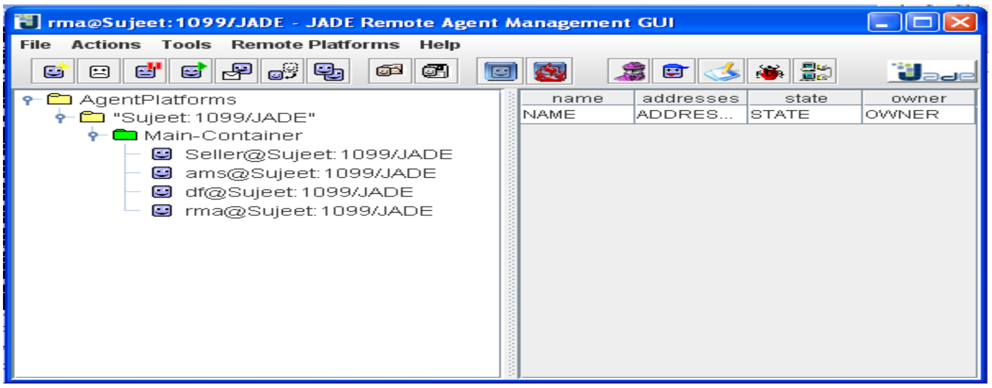

Figure 3: JADE Screen

Now we can create various intelligent agent in the main containers deployed in the tool.

VIII. CONCLUSION

In this paper, we have applied the Multi-agent system in the supply chain management sytem. This approach is more efficient in comparison of other techniques such like genetic algorithm, machine learning & searching algorithms. Due to autonomous & distributed nature of Multi-agent system, it is more efficient in term of time space. Other techniques are time consuming. It provides more flexibility. It means there is scope of flexibility. Whole system is not dependent on set of predefined rules.. Further research will mainly focus on the improvement of the Multi-agent case base reasoning system. For the RETRIEVE phase, it should be done in distributed mode. The case should be retrieved from all other individual intelligent agent case base.

REFERENCES

[1] Weiss, G. "Multiagent systems", The MIT Press, Cambridge,Massachusetts, 1999.

[2] Wooldridge, M., and Jennings, N.R. "Intelligent agents: theory and practice". The Knowledge Engineering Review, 10, 2, 115-152. 2002.

[3] FIPA. "The foundation for intelligent physical agents". URL: http://www.fipa.org, 2003.

[4] Sarne, D., and Grosz, B. "Estimating Information Value in Collaborative Multi-Agent Planning Systems" In *AAMAS*. Forthcoming., 2007

[5] M. M. Veloso, "Planning and Learning by Analogical Reasoning", Lecture Notes in Artificial Intelligence 886, Springer, Berlin, 1994.

[6] E. Crawford and M. Veloso "Opportunities for learning in multi-agent meeting scheduling" In Proceedings of the Fall AAAI Symposium on Artificial Multiagent Learning, Wash- ington, DC, October 2004.

[7] A. B. Hassine and T. B. Ho. "An agent-based approach to solve dynamic meeting scheduling problems with preferences" Eng. Appl. Artif. Intell., 20(6):857–873, 2007.

[8] Sahay, B. S. & Gupta, A. K, Supply Chain Modeling and Solutions, (1sted.). New Delhi: Macmillilan India Ltd.,2007

[9] Schnetzler, M., "Supply Chain Strategies for Business Success", Proceedings of the International IMS Forum 2004 Global Challenges in Manufacturing, (pp. 691-698.) Cernobbio, Italy., 2004

[10] Sebestyénová, "Case-Based Reasoning in Agent-Based Decision Support System", ActaPolytechnicaHungarica, 4(1), pp. 127 – 138, 2007.

[11] Z. Udin, "Collaborative Supply Chain Management:The Hybrid Knowledge-Based Development Approach of Suppliers-Customers Perspective", Operations and Supply Chain, Vol. 3 Issue 2, pp. 130-147, 2008