8-2001

A Multi-Agent Framework for Supporting Intelligent Fourth-Party Logistics

Hoong Chuin LAU
Singapore Management University, hclau@smu.edu.sg

G. LO

A MULTI-AGENT FRAMEWORK FOR SUPPORTING WEB-BASED INTELLIGENT FOURTH PARTY LOGISTICS

LAU, H.C.[1] and **LO YUK TING, G.**[2]
School of Computing
National University of Singapore
3 Science Drive 2
Singapore 11754
Singapore
lauhc@comp.nus.edu.sg
loyuktin@comp.nus.edu.sg

Abstract: We propose a distributed intelligent agent-based framework that supports fourth-party logistics optimization under a web-based e-Commerce environment. In our system, customer job requests come through an e-Procurement service. These requests are consolidated and pushed to the e-Market Place service periodically. The e-Market Place then serves as a broker that allows intelligent agents to bid to serve these requests optimally in real-time by solving multiple instances of underlying logistics optimization problem. The resulting system was implemented based on the Java™ 2 Enterprise Edition (J2EE™) platform using distributed system technology for communication between objects.

Keywords: e-Logistics, Supply chain integration, Decision support systems.
Other Keywords: Distributed systems, Enterprise Java, Intelligent agents, Optimization, Vehicle routing.

1. INTRODUCTION

Merrill Lynch predicted that by 2009, up to 85 percent of all business transactions will be conducted on the Internet [1]. The Yankee Group estimates there will be $3 trillion in B2B transactions by 2004. Of that, nearly $850 billion will be generated by e-market transactions. Gartner estimates that B2B e-commerce will represent 7% of the forecasted $105 trillion global sales transactions. Forrester predicts that worldwide B2B e-commerce will reach $2.7 trillion, and 53% of which will be conducted via online market places [2].

The success of e-Commerce ultimately lies in the ability to fulfill the requests (i.e. movement of goods to the customers' doorsteps) through effective logistics management. Fulfillment problems faced by Toysrus.com and many others during the 1999 US holiday season tells us that fulfillment has a long way to go in order to cope with e-Commerce. The recent technology stock crisis experienced by dot-com companies is a clear reflection of the over-promise of e-Commerce.

With the Internet, literally all details of logistics activities such as customer demands, warehousing, inventory and transportation information can be made available electronically. This opens the possibility for logistics activities to be streamlined and optimized across company boundaries on a macro basis. For this vision to be realized, third-party logistics operators (3PL) will need to form alliance or a consortium, which is to be managed by a fourth-party logistics provider (4PL) whose role is to provide optimized plans and schedules for the member companies. One of these management functions is to efficiently and intelligently fulfills the demands of customers at their specified timings with the available transportation resources.

With the above motivation as the backdrop, we envisage the need to provide an integrated, robust and distributed system that empowers e-Procurement, e-Market Place and logistics agents to work seamlessly together over the Internet.

---

[1] LAU Hoong Chuin
[2] LO YUK TING Gary

We have developed an innovative software framework and a prototype system that combines a hybrid of technologies involving web technology, multi-agents technology and optimization technology. In this paper, we describe our software framework. To simplify discussion, we will direct our discussion on an application of the system in home delivery. In this system, the logistics operators are assumed to own a number of loose-cargo (LTL) trucks. There are a number of depots from which all goods, assumed identical, are picked up. Customers submit jobs by specifying a demand load (number of units of goods to deliver), a number of preferred time windows of delivery and their geographical locations. The system will be responsible for satisfying this job by assigning it to a logistics operator who is capable of transporting the specified units of goods to the customers' location at one of her prescribed time windows.

2. BACKGROUND

Web technology enables enterprise systems and applications to be deployed on the web. The Java™ 2 Platform, Enterprise Edition (J2EE™) [7] defines the standard and a suite of tools for developing web-based multi-tier enterprise systems and applications. J2EE simplifies enterprise applications by basing them on standardized, modular components, by providing a complete set of services to those components, and by handling many details of application behavior automatically, without complex programming.

Multi-agent systems have its root in artificial intelligence. An intelligent agent is a computational entity (such as a software program) that is autonomous in that it operates intelligently and rationally in an environment. In our case, this behavior refers to the ability to perform planning and scheduling given a set of jobs. In a multi-agent system, agents interact with one another to either co-operate or compete in solving problems. An excellent book about multi-agent systems is [8]. In our work, we represent each 3PL by an agent, and the 4PL as a supervisor agent.

Optimization (also known as advanced planning and scheduling) is the underlying technology that enables resources to be effectively assigned to jobs such that certain objectives (such as cost or profit measures) are optimized. In this paper, the underlying optimization problem is the Vehicle Routing with Time Windows, which is a well-studied combinatorial optimization problem. It should be noted that this problem is NP-hard [4], which means that an optimal algorithm is too time-consuming in the worst case. Much progress has been achieved by using heuristic algorithms in determining near-optimal solutions. For a comprehensive recent survey of algorithms to solve this problem, the reader may refer to [5]. In this paper, we encapsulate heuristics algorithms into the agents.

In the remainder of this section, we describe the terminology we will be using throughout the paper.

Table 1: Terminology used throughout the paper.

Name	Description
Administrator	Someone who manages the e-Market Place. In this paper, the Administrator represents the 4PL.
Agent owner	Someone who owns and manages an agent. In this paper, an Agent owner represents a 3PL provider.
Customer	Someone who submits a job request.
Job	Job submitted by a customer, which comprises a customer ID, location, demand load and a list of preferred time windows
Time window	A date, followed by a start time and end time, e.g. 1 Feb 2001, 9am-12noon.
Resources	Consist of a set of vehicles with their vehicle capacities and the availability time windows.
Plan	The set of routes created by the respective agents.
Transaction	Information about a job existing in the e-Market Place. After the job has been assigned to a resource, this information comprises the agent ID, depot ID, vehicle ID, and actual service time windows.

3. SYSTEM REQUIREMENTS

In this section, we discuss our concept of the behavior of the proposed system.

Recall that the proposed system is to be an integrated system that serve both the customers, the 3PLs and the 4PL. Hence, from the user point of view, the following facilities are supported by the system:

- Customers can submit a job request.
- Customers can subsequently check its job request status.
- Administrator can trigger bidding process.
- Administrator can view transactions of registered agents.
- Agent owner can either register its agent into the system or unregister it from the system.
- Agent owner can specify resources for its agent.
- Agent owner can view its agent's plan.

As far as the system is concerned, the following are its specifications.

- System is a fully integrated web-based application.
- System implements services for e-Procurement, e-Market Place and agents.
- Each of these services can independently run on its respective server.
- System is built on the J2EE™ architecture.
- System is able to re-schedule plans dynamically when new job requests arrive.
- Agents should make efficient use of available resources.

The following is a normal scenario that describes the entire workflow of the system:

- Customer submits a job request to the system.
- Agent owner registers its agent with the e-Market Place.
- Agent owner specifies available resources for its agent.
- Administrator connects to the e-Procurement Server to download newly arrived jobs.
- These jobs are then broadcast to all registered agents.
- All agents perform their own planning and scheduling with their own resources based on their own algorithms. (Note that this takes into consideration previous job commitments.) They then make bids for the jobs that they wish to satisfy.
- Bids are returned to the e-Market Place, where the supervisor agent will do a final allocation (based on the bids) and inform the agents about the allocation.
- All agents update their original plans according to the final allocation.
- The final allocation, with detailed information on the assignments, is pushed to the e-Procurement Server.
- Customer can check for status of its job from the e-Procurement Server.
- Administrator can view all transactions from the e-Market Place Server.
- Each agent owner can view its agent's plans from the Agent Server.
- Agent owners can decide whether to have their agent participate in the next round of bidding by leaving the agent registered or unregistering it.

4. SYSTEM OVERVIEW

This section describes how our proposed system is constructed. The reader is assumed to familiar with J2EE™ terminologies, which are explained in [7].

In our proposed system, there are three servers, namely the e-Procurement Server, e-Market Place Server and one (or as many as required) Agent Server. The following tables list all the software components deployed within each of the 3 servers.

Table 2: Components in e-Procurement Server.

Component	e-Procurement Server
Enterprise Bean	DatabaseBean
Web Component	SubmitJob CheckJobStatus

Table 3: Components in e-Market Place Server.

Component	e-Market Place Server
Enterprise Bean	DatabaseBean
Web Component	AgentRegistration MarketPlaceManager ViewTransactions ViewStatistics

Table 3: Components in Agent Server.

Component	Agent Server
Enterprise Bean	DatabaseBean AgentBean
Web Component	SpecifyResources ViewPlans

Figure 1 shows how the Enterprise Java^{TM} Beans (EJB) and web components are related architecturally within the e-Market Place Server. The same can be inferred for the e-Procurement Server and the Agent Server. In the Agent Server, the Agent Bean encapsulates the logic of the agents that perform the scheduling of the routes. Various optimizing algorithms have been used in the system. They will be further discussed in the section on **SYSTEM DESCRIPTION**.

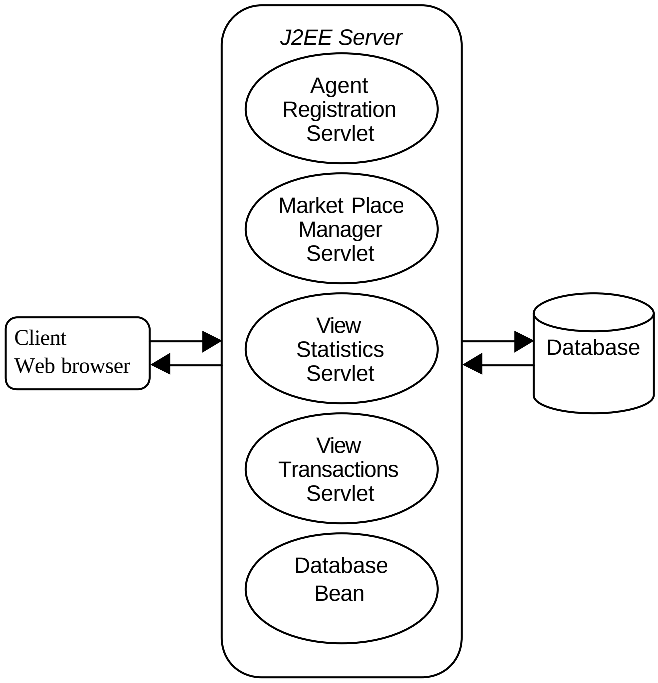

Figure 1. Architectural design of e-Market Place Server.

5. SYSTEM INTEGRATION

This section provides details on how the various components of the system communicate. Major emphasis is placed on the communication between the e-Procurement Server, e-Market Place Server and the Agent Servers. Figure 2 depicts the direction of the communication between these components.

Communication can only be initiated by the e-Market Place Server. Therefore, the e-Procurement Server and Agent Servers are observer classes awaiting the e-Market Place Server to establish connection for the bidding process to occur.

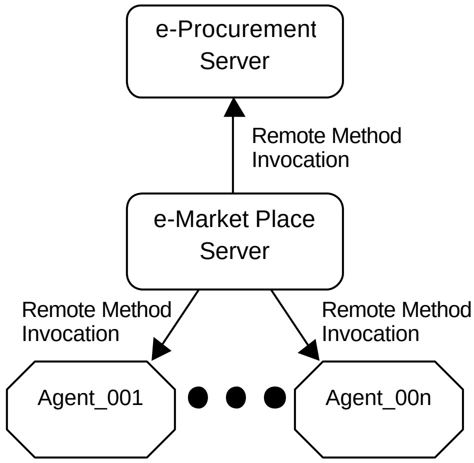

Figure 2. Integrated system with agents deployed.

As shown in Figure 2, Java's Remote Method Invocation (RMI) is the communications mechanism used. In case where the various services are not Java-based, one would require a more heterogeneous communications protocol such as CORBA.

We recognize that, in a real life application where hundreds or even thousands of customers and agents were participating at the e-Market Place, there could be a large number of job requests that would need to be downloaded to the e-Market Place and allocated to the agents. This results in very high volume of transactions and hence communications bandwidth. In order to guarantee certain level of reliability and scalability, our proposed system would need to be deployed on commercial web and application servers and run on broadband networks.

6. SYSTEM DESCRIPTION

6.1 Intelligent Agents

The fully integrated system comprises the e-Procurement Server, the e-Market Place Server and multiple agents. These agents have their own optimizing strategies and own resources. For the agents to bid for jobs on the e-Market Place Server, they must register themselves and have sufficient resources available. With at least one agent registered into the system, a bidding process can be carried out. The Administrator can trigger the whole process remotely via a web browser. Consequently, newly arrived jobs are downloaded from the e-Procurement Server and allocated by the e-Market Place Server to all the agents.

Registered agents will perform their own planning and scheduling with their available resources. As a result, each registered agent will have its plans. Plans from all agents are sent to the e-Market Place Server where a final decision will be made about which agent will get a particular job (in case the same job is bidded by more than one agents). Currently the final allocation is done by randomized First-Come-First-Served, which means that if for each job J that is bidded by agents A_1 to A_n, each agent A_I is assigned a probability proportional to the time its bid arrives, and one agent is selected randomly according to this probability distribution. This allocation scheme is by far simplistic. However, the system allows the user to define any arbitrary bidding and allocation mechanism that the user deems fit.

Upon the final allocation, agents will need to update their plans accordingly. At the same time, the assigned jobs are pushed to the e-Procurement Server where customers can login to check for their job status.

6.2 Optimization Algorithms

Each agent will plan its set of routes for their vehicles according to job requests from customers. This involves solving the Vehicle Routing Problem with Time Windows, minimizing the number of vehicles as well as distance travelled by the vehicles. In our system, each agent runs its own optimization algorithm. They compete against each other to acquire a maximum number of jobs to service. They have been made to compete because participating agents in the system could each be provided by separate organizations, therefore gives rise to competition to acquire job requests by these organizations.

In our systems, the agents use different optimization algorithms adopted from the literature. Typically, each algorithm consists of two phases: (a) the construction phase to initialize the routes based on Greedy heuristics, and (b) the second stage where the routes would be further optimized.

The second stage is based on a sophisticated search strategy called Tabu Search [6]. Tabu Search (TS) is a form of local search augmented with adaptive memory. In TS, a move operator defines the neighborhood of the current solution. Starting with an initial solution, TS proceeds iteratively by replacing current solution with a best neighbor among all possible moves. One crucial feature of TS is the notion of a tabu list, which is a short-term memory that helps the search avoid cycling as well as escape from local optimality. In this system, we devise a Tabu Search scheme based on the following neighborhoods:

- Inter-route Two Opt
- Inter-route Relocate
- Intra-route Two Opt.

A combination of the above algorithms have been used on different agents, therefore giving them different behaviors so that their performance can be compared. The comparison of the different algorithms under this multi-agent environment is beyond the scope of this paper.

7. CONCLUSION

We designed and implemented a J2EE^{TM} based distributed system for an intelligent 4PL coordinating the activities of 3PLs. This work has the potential to become a commercial tool for enabling real-time one-stop integrated planning and optimization that intelligently matches demands of shippers with transportation service providers. It is a system that starts with a strategy and ends with technology, the technology to do smart e-business.

8. ACKNOWLEDGEMENT

This work is supported by the National University of Singapore. We thank Kent Ridge Digital Labs for providing the motivation for this work, and a development environment for the first version of this system.

9. REFERENCES

[1] E-Commerce Times, *E-Commerce Rewards the Fearless*, August 3, 2000.

[2] Broadbase Software Inc., *Broadbase E-Market Place*, 2000. http://www.broadbase.com/news/emktplace_faq.asp

[3] Trintech Group, *Statistics for Electronic Transactions*, 2000. http://www.epaynews.com

[4] M. M. Solomon, Algorithms for the vehicle routing and scheduling problem with time window constraints, Operations Research, 35, 254-265, 1987.

[5] Desrosiers J. *et al*., *Time Constrained Routing and Scheduling,* In *Handbooks in Operations Research and Management Science, Vol 8: Network Routing,* North-Holland, Amsterdam, pp. 35-139, 1985.

[6] Glover F., and Laguna M., *Tabu Search*, Reading, *Kluwer Academic. Publ*., 1997.

[7] J2EE Documentation. http://java.sun.com/j2ee/

[8] Gerhard Weiss (ed.), *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence*, The MIT Press, Massachusetts, 1999.